Environmental Effects From SRB Exhaust Effluents - Technique Development and Preliminary Assessment


CONTRACT NAS8-31806
NOVEMBER 1977
Environmental Effects From SRB Exhaust Effluents - Technique Development and Preliminary Assessment

Science Applications, Inc.
Huntsville, Alabama

Prepared for
George C. Marshall Space Flight Center
under Contract NAS8-31806

NASA
National Aeronautics
and Space Administration
Scientific and Technical Information Office
1977
FOREWORD

This final report is submitted to Aerospace Environment Division, Space Sciences Laboratory, Science and Engineering Directorate, NASA-Marshall Space Flight Center, Alabama, in partial fulfillment of requirements under Contract No. NAS8-31806.

Science Applications, Inc., is indebted to Dr. Briscoe Stephens, Atmospheric Diffusion/Environmental Assessment Task Team, Aerospace Environment Division, for the technical guidance and useful suggestions in planning the revision to the NASA/MSFC Multilayer Diffusion Models and in the development of the objective meteorological parameter selection concepts. The suggestions and the assistance of the following are gratefully acknowledged: the H. E. Cramer Company - diffusion modeling; the IIT Research Institute - scavenging and washout data; Dr. Ronald Dawbarn of ARO, Inc. - Al₂O₃ experiments; NASA Langley Research Center - measurements and laboratory chemistry; McDonnell Douglas Corporation - Delta Thor parameters; Chemical Systems Division of United Technologies Corporation - Titan III booster parameters; U.S. Air Force Weather Service and their range contractor, Pan American World Airways, Inc. - meteorological data collected at Kennedy Space Center. The work under this contract is under the direction of Arnold I. Goldford, Science Applications, Inc.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>2</td>
<td>CLIMATOLOGICAL ASSESSMENT</td>
</tr>
<tr>
<td>2.1</td>
<td>Selection of Average Year</td>
</tr>
<tr>
<td>2.2</td>
<td>Meteorological Regimes</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Air Quality Impact and Associated Meteorological Patterns</td>
</tr>
<tr>
<td>2.3</td>
<td>Preliminary Results from 1969 Meteorological Data</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Air Quality Guidelines</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Transport Direction of the Stabilized SRB Cloud</td>
</tr>
<tr>
<td>2.4</td>
<td>New Objective Concepts</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Transport Layer Height Determination</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Bivariate Normal Wind Distribution</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Development of Objective Methods for Estimation of Meteorological Input Variables for the Multi-Layer Diffusion Model</td>
</tr>
<tr>
<td>2.5</td>
<td>Modification to the Multi-Layer Diffusion Model and Pre-Processor Programs</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Screening Program Modifications</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Addition of New Vehicle and Updating of Constants</td>
</tr>
<tr>
<td>2.6</td>
<td>Surface Deposition Model</td>
</tr>
<tr>
<td>2.7</td>
<td>Absorption and Scavenging</td>
</tr>
<tr>
<td>2.8</td>
<td>Conclusions</td>
</tr>
<tr>
<td>3</td>
<td>EXHAUST CHEMISTRY</td>
</tr>
<tr>
<td>3.1</td>
<td>Two-Phase Flow Phenomena</td>
</tr>
<tr>
<td>3.3</td>
<td>Afterburning and Mixing Analysis</td>
</tr>
<tr>
<td>3.3</td>
<td>Other Losses</td>
</tr>
<tr>
<td>3.4</td>
<td>Conclusions</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (CONTINUED)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>CONVERSION PROGRAMS</td>
</tr>
<tr>
<td>4.1 Conversion Programs (Generic)</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1.1 IBM 370/360 BCD → ASCII Conversion Programs</td>
<td>4-2</td>
</tr>
<tr>
<td>4.1.2 IBM 370/360 EBCDIC → ASCII Conversion Programs</td>
<td>4-3</td>
</tr>
<tr>
<td>4.1.3 UNIVAC 1108 BCD → ASCII Conversion Program</td>
<td>4-3</td>
</tr>
<tr>
<td>4.1.4 UNIVAC 1108 EBCDIC → ASCII Conversion Program</td>
<td>4-4</td>
</tr>
<tr>
<td>4.2 Conversion Programs (Specific)</td>
<td>4-4</td>
</tr>
<tr>
<td>4.2.1 HP EBCDIC → ASCII Conversion Program</td>
<td>4-5</td>
</tr>
<tr>
<td>4.2.2 1965 KSC Rawinsonde Data Conversion Program</td>
<td>4-5</td>
</tr>
<tr>
<td>4.2.3 1974 Vandenberg Rawinsonde Data Conversion Program</td>
<td>4-9</td>
</tr>
<tr>
<td>4.2.4 1964-1970 Jimsphere Data Conversion Program</td>
<td>4-13</td>
</tr>
<tr>
<td>4.3 Conclusions and Recommendations</td>
<td>4-15</td>
</tr>
<tr>
<td>5</td>
<td>INTERACTIVE REEDA PROGRAMS</td>
</tr>
<tr>
<td>5.1 MOD3A Program</td>
<td>5-1</td>
</tr>
<tr>
<td>5.2 MOD3B Program</td>
<td>5-2</td>
</tr>
<tr>
<td>5.3 JIMPL Program</td>
<td>5-2</td>
</tr>
<tr>
<td>5.4 METPL Program</td>
<td>5-7</td>
</tr>
<tr>
<td>5.5 STAN5 Program</td>
<td>5-7</td>
</tr>
<tr>
<td>5.6 MIXH Program</td>
<td>5-9</td>
</tr>
<tr>
<td>5.7 SKEW T Program</td>
<td>5-10</td>
</tr>
<tr>
<td>5.8 Conclusions and Recommendations</td>
<td>5-10</td>
</tr>
<tr>
<td>6</td>
<td>NUMERICAL CLOUD RISE MODEL</td>
</tr>
<tr>
<td>6.1 Modifications to PUFF</td>
<td>6-5</td>
</tr>
<tr>
<td>6.1.1 General Changes</td>
<td>6-5</td>
</tr>
<tr>
<td>6.1.2 Ground Cloud Changes</td>
<td>6-7</td>
</tr>
<tr>
<td>6.2 Current Status of PUFF Program</td>
<td>6-8</td>
</tr>
<tr>
<td>6.3 Conclusions</td>
<td>6-10</td>
</tr>
</tbody>
</table>

iv
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>overall conclusions and recommendations</td>
<td>7-1</td>
</tr>
<tr>
<td>7.1 recommended study</td>
<td>7-1</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>references cited</td>
<td>8-1</td>
</tr>
</tbody>
</table>

**Appendix**

<p>| A                             |      |
| software source listings      | A-1  |
| a-1 conversion programs (generic) | A-2  |
| a-2 conversion programs (specific) | A-16 |
| a-3 interactive reeda programs | A-27 |</p>
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Maximum Predicted (Model 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) For a Normal Space Shuttle Launch (Rawinsonde Input Data For 11/16/69, 12Z)</td>
</tr>
<tr>
<td>2-2</td>
<td>Maximum Predicted (Model 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) For a Normal Space Shuttle Launch (Rawinsonde Input Data For 11/16/69, 12Z, Modified For Model Sensitivity Test; Surface (16 Ft) Temperature Reduced By 7°C)</td>
</tr>
<tr>
<td>2-3</td>
<td>Cumulative Distribution of Maximum Predicted Ten-Minute Dosage Expressed in Percent of NAS Short-Term Public Limit (STPL)</td>
</tr>
<tr>
<td>2-4</td>
<td>Transport Direction at Cloud Stabilization Height Expressed in Percent Occurrence</td>
</tr>
<tr>
<td>2-5</td>
<td>Typical Temperature Soundings for Various Categories of Transport Layer Height</td>
</tr>
<tr>
<td>2-6</td>
<td>Hypothetical Schematic Representation of Relation Between Air Quality Impact of SRB Cloud and the Difference Between Cloud Stabilization Height and Transport Layer Height ( (H_s - H_m) )</td>
</tr>
<tr>
<td>2-7</td>
<td>Titan III C Trajectory</td>
</tr>
<tr>
<td>2-8</td>
<td>Space Shuttle Trajectory</td>
</tr>
<tr>
<td>2-9</td>
<td>Minuteman II Trajectory</td>
</tr>
<tr>
<td>2-10</td>
<td>Delta Thor 2914 Trajectory</td>
</tr>
<tr>
<td>2-11</td>
<td>Delta Thor 3914 Trajectory</td>
</tr>
<tr>
<td>3-1</td>
<td>Solid Rocket Motor, Nozzle, and Plume</td>
</tr>
<tr>
<td>3-2</td>
<td>Shuttle Altitude, Mach Number, and SRB Chamber Pressure History</td>
</tr>
<tr>
<td>3-3</td>
<td>Ratio of Stagnation to Chamber Pressure vs Radial Distance Along Last Left-Running Characteristic</td>
</tr>
<tr>
<td>3-4</td>
<td>Dimensionless Axial Coordinate ( (X/R_t) ) SRB Nozzle Diagram</td>
</tr>
<tr>
<td>3-5</td>
<td>Plume Afterburning Schematic</td>
</tr>
<tr>
<td>3-6</td>
<td>Space Shuttle Solid Rocket Motor Exhaust Effluents (Concentrations)</td>
</tr>
<tr>
<td>3-7</td>
<td>Space Shuttle Solid Rocket Motor Exhaust Effluents (Weights)</td>
</tr>
<tr>
<td>4-1</td>
<td>IBM 370/360 BCD→ASCII Conversion Process</td>
</tr>
<tr>
<td>4-2</td>
<td>UNIVAC 1108 BCD→ASCII Conversion Process</td>
</tr>
<tr>
<td>4-3</td>
<td>ReFormatting to 80 Character Records then Converting BCD/EBCDIC→ASCII</td>
</tr>
<tr>
<td>4-4</td>
<td>1965 Rawinsonde Data Base Generation and Processing</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4-5</td>
<td>1974 Vandenberg AFB Rawinsonde Data Converting and Processing</td>
</tr>
<tr>
<td>4-6</td>
<td>1974 Vandenberg AFB Centerline Concentration and Dosage Plot</td>
</tr>
<tr>
<td>4-7</td>
<td>1974 Vandenberg AFB Isopleth Contour Plot</td>
</tr>
<tr>
<td>4-8</td>
<td>Jimsphere Data Conversion and Processing</td>
</tr>
<tr>
<td>5-1</td>
<td>Example Jimsphere Scalar Wind Speed Plot</td>
</tr>
<tr>
<td>5-2</td>
<td>Example Jimsphere Wind Direction Plot</td>
</tr>
<tr>
<td>5-3</td>
<td>Example of Operations JWSPL Plasmascope</td>
</tr>
<tr>
<td>5-4</td>
<td>Example of METPL Plasmascope Program</td>
</tr>
<tr>
<td>5-5</td>
<td>Example of SKEW T Plot</td>
</tr>
<tr>
<td>6-1</td>
<td>General Representation of the PUFF Program</td>
</tr>
<tr>
<td>6-2</td>
<td>Cloud Shape</td>
</tr>
<tr>
<td>6-3</td>
<td>Exhaust Cloud Configuration</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2-1</td>
<td>Percent Occurrence of Synoptic and Air Mass Types at Kennedy Space Center During 1965</td>
</tr>
<tr>
<td>2-2</td>
<td>Percent Occurrence of Synoptic and Air Mass Types for the Percent Sample (726 cases) and the Sub-sample (102 cases) for 1965 at KSC</td>
</tr>
<tr>
<td>2-3</td>
<td>Summary of Two Worst Cases from 1969 Sample of 101 Cases</td>
</tr>
<tr>
<td>2-4</td>
<td>Transport Layer Height Percent Occurrence in Six Categories Derived from Weekly Rawinsonde Soundings at KSC During 1969</td>
</tr>
<tr>
<td>2-5</td>
<td>Transport Layer Height Percent Occurrence in Five Categories Derived from Weekly Rawinsonde Soundings at KSC During 1969</td>
</tr>
<tr>
<td>2-6</td>
<td>Calculations of SIGAR using Rawinsonde Data Between the Surface and the First Standard Pressure Level (1,000 mb) with Z = .25 m</td>
</tr>
<tr>
<td>2-7</td>
<td>Comparison of calculated SIGAR with Values Given by Dumbauld</td>
</tr>
<tr>
<td>2-8</td>
<td>Distribution of SIGAR Computed from Rawinsonde Data</td>
</tr>
<tr>
<td>2-9</td>
<td>Mean SIGAR as Function of Potential Temperature Gradient for Two Wind Speed Intervals</td>
</tr>
<tr>
<td>2-10</td>
<td>Stability Criteria</td>
</tr>
<tr>
<td>2-11</td>
<td>Preprocessor Program Constants</td>
</tr>
<tr>
<td>2-12</td>
<td>SRB Particle Size Distribution</td>
</tr>
<tr>
<td>3-1</td>
<td>Space Shuttle SRB Exhaust Effluents at Chamber, Throat and Exit Plane</td>
</tr>
<tr>
<td>3-2</td>
<td>Afterburning Analysis</td>
</tr>
<tr>
<td>3-3</td>
<td>Space Shuttle SRB Exhaust Effluents Concentrations in Weight Percent</td>
</tr>
<tr>
<td>3-4</td>
<td>SSME Exhaust Effluents</td>
</tr>
<tr>
<td>4-1</td>
<td>Representative Rawinsonde Data</td>
</tr>
<tr>
<td>6-1</td>
<td>Phases of PUFF Program</td>
</tr>
<tr>
<td>7-1</td>
<td>Percent Frequency of Occurrence, to Ground-based Inversions by Season at KSC During 1965-1969 at 0700 and 1900 EST</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This final report discusses efforts made on a study over the last nine months which had as its primary objective the development of techniques to determine the environmental effects from the Space Shuttle SRB exhaust effluents. This determination required the blending together of a team which had diverse skills and capabilities. The study required personnel with experience and knowledge in propulsion chemistry, meteorology, computer technology, and fluid dynamics. The study utilized four different computing systems; the NASA REEDA, the NASA UNIVAC 1108, the NASA IBM 360, and the SAI IBM 370. Knowledge of their operating systems and details of similarities and differences between the machines' data storage, instructions, and peripheral equipment operations was required during the study. Intimate knowledge of meteorological data reduction methods was a necessity. The start of the development of the new cloud rise model required expertise in fluid dynamics. To obtain the source terms for the cloud rise calculations required a state-of-the-art analysis of the SRB and its exhaust effluents.

This study has developed many new and needed tools for the determination of the environmental effects from the Space Shuttle SRB exhaust effluents. A preliminary climatological assessment has been performed which will be used to guide the future full scale climatological assessment. The exhaust effluent chemistry study has been performed and the exhaust species have been determined neglecting several possibly important effects. A reasonable exhaust particle size distribution has been constructed which can be used for the deposition model. The effects of scavenging and absorption have not been included in the preliminary climatological assessment. The basic conclusion that can be drawn regarding the entire study is that the team has now done their homework, understands the complete problem more
fully, has developed the required algorithms, learned the required technology, and is now able to perform a meaningful climatological assessment with the operational REED Description which can yield the required answers about the environmental effects from the Space Shuttle SRB exhaust effluents. These algorithms have not been interfaced into the REED Description.

Section 3 on the exhaust chemistry and Section 6 on the numerical cloud rise model are efforts funded under NASA Contract NAS8-31851. The partial results have been included in this report so that a reader can get a clear picture of the overall effort. It should be noted that the basic studies have been conducted with a Titan type vehicle having all solid propellant motors and not the Space Shuttle type vehicle which has both solid and liquid propulsors. The technology for the problem has been learned but the models must be tuned for the Space Shuttle and its unique characteristics.

This study performed and used the results of a preliminary climatological diffusion assessment to define the problems involved in performing a full scale assessment; therefore, these preliminary air quality results should be used with extreme caution in drawing conclusions regarding the environmental effects of the Space Shuttle exhaust effluents.
2. CLIMATOLOGICAL ASSESSMENT

Environmental impact evaluation will be based on calculations of the ground level concentrations using the NASA/MSFC Rocket Exhaust Effluent Diffusion (REED) Description (1,2) input data for each selected meteorological regime. The use of the REED Description for environmental assessment requires a detailed knowledge of the surface mixing layer. The thermodynamic and kinematic properties of this layer can be measured with radiosondes, tetroonsonde, and other instrumental platforms. Large samples of these data are required for a climatological assessment of environmental impact. The only data set available which is sufficiently large to satisfy this requirement was obtained from radiosondes. These data were obtained daily (at 0000Z and 1200Z) at KSC for more than fifteen years by the U.S. Air Force Air Weather Service. In addition, four soundings per day were taken during a five year period (1962 through 1966).

The tapes containing the radiosonde data will be scanned and subsets of profiles will be established which correspond to the various meteorological regimes that were developed for air quality assessments by Stephens and Sloan (3). These data subsets will ultimately be used as input to the REED Description for calculation of air quality impact.

The data to be used will be the KSC soundings from the period 1962 - 1966(*). The sample cumulative probability distribution of maximum ground-level concentrations attributed to each meteorological regime will be calculated; these probability distributions will be useful for estimation of the probability of exceeding a specified maximum concentration for a particular regime.

(*) The data tapes were obtained from the U.S. Air Force Range contractor, Pan American World Airways.
2.1 SELECTION OF AVERAGE YEAR

Monthly average surface data during the subject period (1962 through 1966) were used for determination of the year which was most representative of normal conditions at KSC. Because of the convenience of obtaining the required summaries from regular NOAA weather stations, climatological data from a similar coastal location (Daytona Beach) 50 miles from KSC were used to represent KSC.

The criterion for selection of a particular year was that it have the smallest value of the parameter \( D \) given by

\[
D = \frac{1}{12} \left[ \sum_{i=1}^{12} |T'_{M_i}| + \sum_{i=1}^{12} |T'_{m_i}| + \sum_{i=1}^{12} |v'_{m_i}| \right]
\]

where \( i=1 \) corresponds to January, \( i=2 \), February, etc. and \( |T'_{M_i}| \), \( |T'_{m_i}| \) and \( |v'_{m_i}| \) are the absolute deviation of the monthly mean daily maximum and minimum temperature and monthly mean wind speed from their respective normal monthly means; the quantity \( D \) represents the average monthly total absolute deviation for the three parameters.

The calculated values of \( D \) are given in the table below.

<table>
<thead>
<tr>
<th>Year</th>
<th>62</th>
<th>63</th>
<th>64</th>
<th>65</th>
<th>66</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>4.85</td>
<td>5.30</td>
<td>4.83</td>
<td>3.41</td>
<td>3.65</td>
</tr>
</tbody>
</table>

Thus, the year 1965 was selected as the year most representative of normal conditions at KSC.

In connection with our selection of a climatological data set, the following background data for KSC were acquired from the National Weather Records Center:

- Monthly and annual inversion statistics for the period 1965 through 1969 based on KSC Rawinsonde data
- Monthly and annual STAR summary of atmospheric stability for the period 1965 through 1969 based on Cocoa Beach surface data
Monthly and annual mixing height statistics for the year 1965 based on KSC Rawinsonde data

With regard to the 1965 KSC Rawinsonde data tape, we intend to calculate the following statistics as a function of time of day:

- Distribution of atmospheric stability calculated between 1.2 to 1.5 km (4,000 to 5,000 ft) by taking the gradient of virtual potential temperature
- Distribution of the height of ground based inversions
- Distribution of wind speed at 1.2 km
- Distribution of wind direction at 1.2 km

These statistics are correlated with the diffusion potential of the ambient air at typical SRB cloud stabilization altitudes. The distribution of the height of ground based inversions is useful in the study of how often SRB clouds are expected to penetrate such inversions and thus become effectively isolated from the ground; ground based inversions are also responsible for the largest concentrations observed at ground level whenever there is a release from a non-buoyant ('cold') source. The distribution of wind direction at the typical height of SRB cloud stabilization chosen (1.2 km) is correlated with the expected track of the SRB cloud at the calculated stabilization height.

2.2 METEOROLOGICAL REGIMES

In support of air quality assessments for aerospace vehicle exhaust effluents at Kennedy Space Center, meteorological regimes were defined which correspond to synoptic patterns (3). These regimes are designed to narrow the air quality statistics into categories that reflect temporal development of atmospheric conditions at launch.
In the past the meteorological inputs to the NASA/MSFC REED Description have been based mostly on climatological statistics until about 12 hours prior to launch, at which time a deterministic forecast was made. An obvious drawback to this approach is that the statistical air quality assessment during the pending launch period, two or four days prior to launch, does not reflect atmospheric dynamics identifiable from current synoptic conditions.

Thus, the purpose of defining meteorological regimes in terms of synoptic conditions is to provide a realistic means of classifying subsets of the overall climatological data set for statistical air quality assessments. Since these subsets are more representative of developing atmospheric conditions during the pending launch period, the use of these subsets assures a smoother interface of the statistical air quality assessment with the deterministic assessment. Employing this classification system, the statistical assessment affords error bounds for the deterministic predictions.

It is necessary to consider the types of atmospheric data sources and the applications for which the results of the diffusion predictions will be utilized in order to define appropriate meteorological regimes. The amount of detail required in the atmospheric kinematics is dictated by the planned application of the diffusion prediction. Two extremes in applications are air quality and deployment predictions. If the diffusion predictions are to be utilized in support of air quality predictions to insure public safety, the detail in the atmospheric input parameter can be relaxed in favor of slightly conservative values which incorporates a safety factor. Since the desire is to identify any potential for an air quality problem, the exact location and concentrations are of secondary importance as long as the error bounds for these estimates have been determined and are reasonably conservative. For this application, routine radiosonde data
are satisfactory since small spatial and temporal changes in the atmospheric kinematics can be neglected without a serious impact on the creditability of the results.

On the other hand, if the application for the diffusion prediction is to support the deployment of the cost-effective rocket effluent monitoring network, the resolution requirements of the atmospheric input parameters for the REED Description are very stringent. This increase of rigor is introduced by the need for exactness in the predicted transit path. In this case, local spatial and temporal changes in the atmospheric kinematics must be considered. This means that terrain effects and the land-sea interface effects must be known. Since the radiosonde provides predominately vertical information, other sources of data must be used to obtain horizontal-temporal information. In general, wind tower data are not adequate to totally support this requirement since the available information is limited to the surface boundary layer. Currently, the best source of local spatial-temporal information is a tetroonsonde (a constant level balloon with radiosonde) flown nominally at 600 meters (4). Other potential means to obtain or improve the local spatial-temporal information would be from simultaneous multiple radiosonde releases or a remote sensing system. Hence exactness in predictions of the exhaust cloud transit path is limited by the state-of-the-art of the available small scale atmospheric measurement system. Extensive meteorological support of the NASA rocket exhaust effluent prediction and monitoring program have been documented for a series of seven Titan launches (5-11); the hydrogen chloride measurements for the same series are described by Gregory, et al. (12)

A common requirement for a diffusion prediction is the statistical air quality assessment for planning activities prior to a launch. The objective is to use these statistical assessments for mission planning activities to optimize launch windows.
Meteorological regimes needed for air quality assessment prior to launch were defined. The regimes were not intended for detailed launch effluent monitoring support.

Before defining the meteorological regimes, consideration of the selection and sequential nature of the approach will be described. Typically, there are about nine different patterns that could be associated with the weather conditions at Kennedy Space Center. Within each pattern, there are a wide variation in the small scale kinematic and thermodynamic structure depending on the type and intensity of the mesoscale activity present.

It is appropriate to use existing knowledge of seasonal variation at KSC in the selection of seasonal time regimes for statistical analysis. It is apparent that the length of the seasons at KSC are non-uniform with relatively long summers and winters (mid-May through mid-October and December through March, respectively) which are separated by short (approximately 6 weeks) transition periods. It is known that the summer and winter diffusion meteorology will contribute to the largest variation between calculated seasonal environmental impacts; since the realistic seasonal breakdown of data sets increases the size of the winter and summer sample it follows that the comparison of winter and summer will have better statistical reliability.

The approach is to start with the statistical air quality assessment that is normally used in the mission planning activities; initially the seasonal-temporal regimes are defined; that is, the season of the year—winter, spring, summer, or fall—and the time of day—night, morning, afternoon, or evening. Further narrowing of the regime categories will be achieved by sub-division into the following synoptic patterns:

- The Bermuda anticyclone and associated easterly winds.
• Easterly waves and associated strong vertical mixing.
• Westerly waves and associated frontal activity.
• Continental anticyclones and associated northerly winds.

The next step in the process is the qualification of the intensity of the synoptic regime according to nominal "weak" and "strong" categories. Objective criteria will be established for such a qualification.

In summary the regimes established will consist of the following categories:
• Season
• Synoptic regime
• Intensity of synoptic regime
• Time of day

Other regime categories such as thermodynamic or kinematic parameters may be better suited for climatological air quality assessments.

2.2.1 Air Quality Impact and Associated Meteorological Patterns

Air quality impact can be classified according to concurrent synoptic meteorology patterns and air mass types. The relative frequency of occurrence of these patterns during 1965 at KSC has been calculated. NOAA synoptic charts drawn twice daily (1 a.m. and 1 p.m. EST) were used for the analysis. The following synoptic and air mass classification was used:

<table>
<thead>
<tr>
<th>Synoptic Type</th>
<th>Synoptic Class</th>
<th>Air Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Maritime Anticyclone</td>
<td>Maritime Tropical (MT)</td>
</tr>
<tr>
<td>B</td>
<td>Easterly Wave</td>
<td>Maritime Tropical (MT)</td>
</tr>
<tr>
<td>C</td>
<td>Westerly Wave</td>
<td>Maritime Tropical (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Continental Polar (P)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transition (MT-CP)</td>
</tr>
<tr>
<td>D</td>
<td>Continental Anticyclone</td>
<td>Continental Polar (CP)</td>
</tr>
</tbody>
</table>

(1) Specification of the air mass type for Type C is dependent on the type and strength of the front (cold, warm, stationary) and its location relative to KSC.
This classification is essentially the same as that given in Reference 1 with slight modification of synoptic Type A to represent the general category of maritime anticyclones which is composed of two seasonal sub-types. In summer the maritime anticyclone is synonymous with the Bermuda anticyclone which persistently dominates the weather in the Eastern United States. The only break in this persistent pattern occurs in late summer when inverted low pressure troughs embedded in the tropical easterlies move to the vicinity of KSC (Type B, Easterly Wave). These troughs are in rare instances associated with a hurricane. In winter, anticyclones containing cold dry air move southeastward toward KSC; as these circulations pass over the relatively warm water east of the Florida peninsula, they are rapidly modified. Thus, in winter, there is a typical alternating pattern of Type A and Type D anticyclones. The transition between the two types is characterized by Type C (Westerly Wave) conditions which include clouds and precipitation associated with fronts and eastward propagating waves in the westerlies (Type C, Westerly Wave).

The monthly and annual percent occurrence of the various synoptic regimes and air mass types during 1965 is given in Table 2-1. It is clearly indicated that the predominant synoptic regime is the maritime anticyclone (Type A) with an annual occurrence of 57.6 percent; on a monthly basis Type A predominated during the period March through November. During the winter months (December through February) continental anticyclones are often strong enough to penetrate far enough southward to become the predominant synoptic Type D at KSC. The occurrence of air mass types is correlated with the occurrence of the synoptic types.

It is obvious from the analysis that the summer season is the most critical in the assessment of environmental
Table 2-1. Percent Occurrence of Synoptic and Air Mass Types at Kennedy Space Center During 1965

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>27.4</td>
<td>35.7</td>
<td>43.5</td>
<td>66.7</td>
<td>88.7</td>
<td>90.0</td>
<td>90.3</td>
<td>86.8</td>
<td>60.3</td>
<td>37.1</td>
<td>43.3</td>
<td>19.7</td>
<td>57.6</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.5</td>
<td>6.6</td>
<td>20.7</td>
<td>8.1</td>
<td>0</td>
<td>0</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>14.5</td>
<td>19.6</td>
<td>30.6</td>
<td>6.7</td>
<td>1.6</td>
<td>6.7</td>
<td>3.2</td>
<td>6.6</td>
<td>19.0</td>
<td>29.0</td>
<td>21.7</td>
<td>26.2</td>
<td>15.4</td>
</tr>
<tr>
<td>D</td>
<td>58.1</td>
<td>44.6</td>
<td>25.9</td>
<td>26.7</td>
<td>9.7</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>25.8</td>
<td>35.0</td>
<td>54.1</td>
<td>23.6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>58.1</td>
<td>48.2</td>
<td>29.0</td>
<td>20.0</td>
<td>4.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22.6</td>
<td>20.0</td>
<td>50.8</td>
<td>21.1</td>
</tr>
<tr>
<td>MT</td>
<td>33.8</td>
<td>41.1</td>
<td>51.6</td>
<td>70.0</td>
<td>88.7</td>
<td>95.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>53.2</td>
<td>55.0</td>
<td>26.2</td>
<td>67.9</td>
</tr>
<tr>
<td>CP-MT(1)</td>
<td>8.1</td>
<td>10.7</td>
<td>19.4</td>
<td>10.0</td>
<td>6.5</td>
<td>5.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24.2</td>
<td>25.0</td>
<td>23.0</td>
<td>11.0</td>
</tr>
</tbody>
</table>

Synoptic Types
A = Maritime Anticyclone
B = Easterly Wave
C = Westerly Wave
D = Continental Anticyclone

Air Mass Types
CP = Continental Polar
MT = Maritime Tropical
MT-CP = Transitional Air Mass
impact in populated areas west of KSC. During this period Easterly flow associated with the maritime (Bermuda) anticyclone will occur during a large percentage of the time. During the daytime, the synoptic flow is enhanced in the surface layer by the local sea-breeze circulation. As the air associated with the sea-breeze circulation moves onshore, a ground-based mixed layer develops. The thickness of the mixed layer is a function of the intensity of turbulence generated by mechanical interaction of the air with the land surface roughness elements and land-to-air heat transfer. It is hypothesized that concentrations of SRB effluents may occur at ground level locations in areas west of KSC when portions of the stabilized SRB cloud are within the sea-breeze mixed layer. This hypothesis will be tested in a planned study based on the available sample of Rawinsondes obtained during the period 1100 to 1500 EST during the summer months (June through September) of 1965. The sub-sample of soundings which exhibit a well-developed sea-breeze and a mixed layer extending above the stabilized SRB cloud will be used as input data to the UNIVAC 1108 REED Description. The calculated maximum concentrations and dosages will be compared with those calculated at times of the year during different meteorological regimes and times of the day. If the hypothesis is verified for the 1965 data, a more detailed analysis will be initiated based on the additional summer soundings that can be drawn from the existing data tapes for the year 1962 through 1964 and 1966. The results of this study will comprise the maximum estimated impact, assuming that there are no launch constraints based on air quality impact considerations.

During summer nights, the land breeze will tend to be minimized, since it is opposed by the large scale synoptic flow; it is during this period when the flow is poorly organized that the forecasting of SRB cloud trajectory will be the most difficult. However, calculated downwind concentrations during
the night are not expected to be as large as those during the day because of the decreased rate of vertical diffusion associated with the tendency of the atmosphere near the ground to become neutral or stably stratified during this period.

The idea that a representative sub-sample of meteorological data can be drawn from a larger sample was tested by comparing percent occurrence of synoptic and air mass types during 1965 for 102 cases (based on two NOAA synoptic charts per day at 1 a.m. and 1 p.m. EST for one day per week for 51 weeks) to that obtained for 726 cases based on twice daily data for 363 days. The results of this comparison are given in Table 2-2. It is indicated that the statistics of the sub-sample in most categories correspond closely to those of the parent sample. The only significant deviation is for the occurrence of synoptic Type C which is underestimated in the sub-sample. This can be attributed to the fact that Type C is a transient phenomena that is not accurately seen by weekly sampling.
Table 2-2. Percent Occurrence of Synoptic and Air Mass Types for the Parent Sample (726 Cases) and the Sub-sample (102 Cases) for 1965 at KSC

<table>
<thead>
<tr>
<th>Synoptic Type</th>
<th>Air Mass</th>
<th>Parent Sample (726 Cases)</th>
<th>Sub-Sample (102 Cases)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MT</td>
<td>56.2</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MT-CP</td>
<td>1.4</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>57.6</td>
<td>60.8</td>
</tr>
<tr>
<td>B</td>
<td>MT</td>
<td>3.4</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MT-CP</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>3.4</td>
<td>3.9</td>
</tr>
<tr>
<td>C</td>
<td>MT</td>
<td>8.3</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>1.9</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>MT-CP</td>
<td>5.2</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>15.4</td>
<td>7.8</td>
</tr>
<tr>
<td>D</td>
<td>MT</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>19.1</td>
<td>24.5</td>
</tr>
<tr>
<td></td>
<td>MT-CP</td>
<td>4.1</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>23.6</td>
<td>27.5</td>
</tr>
<tr>
<td>TOTAL</td>
<td>MT</td>
<td>67.9</td>
<td>68.6</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>21.1</td>
<td>26.5</td>
</tr>
<tr>
<td></td>
<td>MT-CP</td>
<td>11.0</td>
<td>4.9</td>
</tr>
</tbody>
</table>
A preliminary climatological assessment study was begun using 1969 meteorological data and effluent parameters given in the Agency Environmental Impact Statement for the Space Shuttle (13). A sample of 101 soundings (one day per week and 2 soundings per day) were generated from a 1969 met data tape using the met screening program. With the aid of the AEC and TVA stability criteria output by the program for each sounding, the height of the surface transport layer was chosen and input cards for the pre-processor were assembled. The 101 cases were then run through the multi-layer/pre-processor system and the results tabulated. Table 2-3 shows the two worst cases of the 101 processed. The November 16th case is further illustrated in Figure 2-1. Note that for January 8, the maximum dosage approaches the critical NAS level (2400 PPM-sec) as does the maximum peak concentration for November 16 (critical NAS level = 8 PPM)(14).

Table 2-3. Summary of Worst Cases from 1969 Sample of 101 Cases

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Date</th>
<th>Time</th>
<th>Model</th>
<th>Pollutant</th>
<th>Adjusted Cloud Rise Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Shuttle</td>
<td>01/08/69</td>
<td>12Z</td>
<td>4</td>
<td>HCl</td>
<td>979.18</td>
</tr>
<tr>
<td>Space Shuttle</td>
<td>11/16/69</td>
<td>12Z</td>
<td>4</td>
<td>HCl</td>
<td>1135.57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>Azimuth</th>
<th>Max Peak Conc.</th>
<th>Max Dosage</th>
<th>Max Peak 10 Min Time Mean Conc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>261.03</td>
<td>80.28</td>
<td>1.522</td>
<td>2176.083</td>
<td>1.450</td>
</tr>
<tr>
<td>1062.87</td>
<td>194.78</td>
<td>5.034</td>
<td>719.015</td>
<td>1.198</td>
</tr>
</tbody>
</table>
Figure 2-1. Maximum Predicted (Model 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) for a Normal Space Shuttle Launch (Rawinsonde Input Data for 11/16/69, 12Z).
Examination of data obtained from towers at various locations at KSC has indicated that surface* temperatures can be highly variable. This variability leads to a degree of uncertainty in the diffusion calculations, which are usually based on data obtained at one location. To illustrate this uncertainty the diffusion calculation for the worst case maximum centerline HCl concentration (11/16/69, 12 Z) was repeated using a revised surface temperature of 0°C which was 7°C colder than the original temperature. This temperature difference is within the expected range of variability of surface temperatures at KSC. The results are illustrated in Figure 2-1 (original calculation, surface temperature = 7.0°C.) and Figure 2-2 (revised surface temperature = 0.0°C). It is shown that the revised maximum concentration increased to 6.92 PPM from the original value of 5.03 PPM; the maximum dosage increased from 719 to 831 PPM-sec for the revised data. It is concluded that surface temperature uncertainties in the input meteorological data lead to uncertainties in the calculated air quality impact. Other workers have indicated an uncertainty of as large as a factor of two in the diffusion model results, largely attributable to meteorological uncertainties. However, field measurements taken after TITAN launches (1) suggest a significantly smaller uncertainty (10 to 25 percent).

In view of the uncertainties in the calculations and the limited sample of KSC meteorological data uses, the results for peak concentration described below are considered very preliminary.

By comparing the peak concentration data to NAS standards the following categorization scheme was devised by Dr. Stephens for mapping of the results. Future results based on a large data sample will use the color categories given below:

* Actual height about 2m above surface.
Figure 2-2. Maximum Predicted (Model 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) for a Normal Space Shuttle Launch (Rawinsonde Input Data for 11/16/69, 12Z Modified for Model Sensitivity Test; Surface (16 Ft) Temperature Reduced by 7°C).
<table>
<thead>
<tr>
<th>COLOR</th>
<th>MAXIMUM PEAK CONCENTRATION (PPM)</th>
<th>% of CASES*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>&lt; 4.00</td>
<td>96%</td>
</tr>
<tr>
<td>Yellow</td>
<td>4.01 to 5.00</td>
<td>3%</td>
</tr>
<tr>
<td>Orange</td>
<td>5.01 to 8.00</td>
<td>1%</td>
</tr>
<tr>
<td>Red</td>
<td>&gt; 8</td>
<td>0</td>
</tr>
</tbody>
</table>

* Based on the 101 cases during 1969 at KSC.
2.3.1 Air Quality Guidelines

The climatological air quality assessment of the impact of the Shuttle SRB exhaust cloud requires the comparison of ground-level concentration and dosage predictions to air quality guidelines given by the National Academy of Sciences (NAS), with the exception of industrial standards applicable to KSC which assume chronic exposure. There are no national standards for the short-term exposures associated with aerospace exhaust effluents (Ref 14). A graphical illustration of how a statistical summary of dosage predictions can be compared with an NAS guideline for aerospace applications is given in Figure 2-3. The particular NAS guideline used in the illustration is the short-term public limit for a 10-minute average exposure (STPL 10) which is 4 parts per million (ppm) for HCl with an 8 ppm ceiling. This is equivalent to a dosage of 2400 ppm-sec. The cumulative distribution of maximum 10-minute dosages (expressed in percent of 2400 ppm-sec) predicted by the NASA/MSFC REED Description for 101 cases during 1969 is plotted in Figure 2-3. It is shown that 98 percent of the predicted dosages were less than 34 percent of the NAS standard. The largest predicted dosage was 2176 ppm-sec (January 8, 1969), which was 91 percent of the NAS standard. These results are preliminary. Additional calculations, based on an updated diffusion model, the objective methods for specification of the standard deviation of wind azimuth angle (SIGAR) and transport layer height, and the large sample of data available for 1965 (>1400 cases) will be made as the study continues.

Initial indications suggest that the Space Shuttle does not have an air quality problem under normal atmospheric conditions. However, marginal air quality conditions could exist within KSC which could result in a requirement for crowd control.
Figure 2-3. Cumulative distribution of maximum predicted 10-minute dosage expressed in percent of NAS short-term public limit (STPL). (Preliminary result based on 101 cases calculated from KSC Rawinsonde data at 00Z and 12Z)
2.3.2 Transport Direction of the Stabilized SRB Cloud

The statistics of expected transport direction of the stabilized SRB exhaust cloud give an indication of the direction where significant impact is most probable (assuming no meteorological launch constraints with regard to expected air quality impact). A preliminary evaluation of the distribution of SRB exhaust cloud transport, based on 101 cases in 1969, is illustrated in Figure 2-4. The transport direction was estimated from Rawinsonde data by taking the wind direction at the altitude nearest the cloud stabilization height. In more than 70 percent of the cases this altitude was within 100 meters of the cloud stabilization altitude. For the other cases, examination of the wind direction profile did not justify interpolation to obtain a better estimate of wind direction. Transport direction is taken as 180 degrees plus the wind direction. Thus an east wind (90°) results in a westward transport direction (270°). It is shown that the transport directions with the largest calculated frequency of occurrence (12 percent) were east-southeast and northwest. Further comments on transport direction statistics are reserved for forthcoming calculations based on larger data samples. The distribution of transport direction at KSC will be derived as a function of time of day (0100, 0700, 1300, 1900 EST) for the 1965 transport directions for each time of day that can be obtained using the 1965 data.
Figure 2-4. Transport direction at cloud stabilization height expressed in percent occurrence. Based on weekly Rawinsonde data obtained twice daily (00Z, 12Z) during 1969 (101) cases.
2.4 NEW OBJECTIVE CONCEPTS

A portion of this study effort was spent in exploring new concepts for the objective analyses of the meteorological data. These objective analyses, if proven by theory and test, would allow the automatic selection of REED Description parameters. That would lead not only to a large savings in manpower but to a better, more uniform treatment of the data.

2.4.1 Transport Layer Height Determination

An attempt to develop objective criteria for selection of the transport layer height used in the NASA/MSFC Multilayer Diffusion Model has been made. Acceptable criteria will permit the development of a computer code for the automation of transport layer height selection. Although an acceptable set of objective criteria have not yet been found, preliminary criteria have been established and are being tested.

Two sets of criteria listed in Tables 2-4 and 2-5 were studied. The relative frequency of occurrence of the various transport layer categories are also given in the tables. In the first set of criteria, outlined schematically in Figure 2-5, strong emphasis is placed on the existence of stable layers below the cloud. This results in a large number of transport layer heights below cloud stabilization height, which in effect reduces calculated ground level concentrations by reducing the amount of cloud mass which can be diffused downward. If these stable layers are proven to have a smaller frequency of occurrence because of inaccuracies of the Rawinsonde data or are not related to actual transport layer heights, the calculated air quality impact will not be conservative.

In the present stage of development of our capability to predict air quality impact, it is not desirable to use techniques that may later be proven unconservative.

The criteria listed in Table 2-5 will give conservative results because emphasis is given to the occurrence of wind
Figure 2-5. Typical Temperature Soundings for Various Mixing Depth Categories of Transport Layer Height Given by Circled Number.
Table 2-4. Transport layer height percent occurrence in six categories derived from weekly Rawinsondes at KSC during 1969.

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>PERCENT OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1200Z (0700 EST)</td>
</tr>
<tr>
<td></td>
<td>50 cases</td>
</tr>
<tr>
<td>1. Base of stable layer above cloud; no stable layers or inversions at or below the cloud altitude</td>
<td>10.0</td>
</tr>
<tr>
<td>2. Top of ground based stable or inversion layer in which the cloud is immersed</td>
<td>0.0</td>
</tr>
<tr>
<td>3. Base of stable or inversion layer in which the cloud is immersed; no stable or inversion layers below</td>
<td>10.0</td>
</tr>
<tr>
<td>4. Top of ground based inversion beneath the cloud</td>
<td>48.0</td>
</tr>
<tr>
<td>5. Top of ground based stable layer (A) beneath the cloud</td>
<td>24.0</td>
</tr>
<tr>
<td>6. Base of lowest stable layer beneath cloud</td>
<td>8.0</td>
</tr>
</tbody>
</table>

(A) Category 5 is synonymous with category 4 when the stable layer extending upward from the ground consists solely of a temperature inversion.
Table 2-5. Transport layer height percent occurrence in five categories derived from weekly Rawinsondes at KSC during 1969.

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>PERCENT OCCURRENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00Z</td>
</tr>
<tr>
<td></td>
<td>1900 EST 0700 EST</td>
</tr>
<tr>
<td>1. Base of stable layer above exhaust cloud</td>
<td>64.7</td>
</tr>
<tr>
<td>2. Top of wind shear layer</td>
<td>19.6</td>
</tr>
<tr>
<td>3. Base of wind shear layer</td>
<td>0</td>
</tr>
<tr>
<td>4. Top of surface based stable layer with potential temperature gradient</td>
<td>13.7</td>
</tr>
<tr>
<td>extending to altitudes &gt;250 meters</td>
<td></td>
</tr>
<tr>
<td>5. Top of stable layer in which cloud is immersed</td>
<td>2.0</td>
</tr>
</tbody>
</table>
shears and stable layers above cloud stabilization height.

A code has been written which executes the logic developed by the H. E. Cramer Co. for selection of the height of the surface transport layer; selection is based on criteria for the vertical gradient of virtual temperature $\Delta T_v/\Delta z$ summarized below:

- A ground based inversion is defined if

$$\Delta z \geq 100\text{m and } \frac{\Delta T_v}{\Delta z} \geq -0.0005 \text{ °C/m}$$

- The base of a stable layer at $z_1$ if

$$\Delta z = z_2 - z_1 \geq 100\text{m and} \frac{\Delta T_v}{\Delta z} \leq -0.005 \text{ °C/m}$$

where $z_2 > z_1$

If a ground based inversion exists, the height of surface transport layer is specified as the top of the ground based inversion; otherwise it is the height of the base of the first stable layer above the ground. If the base of the first stable layer is above 3000m the depth is set equal to 3000m.

The code will be used for specification of the surface transport layer for the 1965 Radiosonde data (>1400 cases). The calculation of transport layer height, $H_m$, will be made concurrent with the calculation of the stabilization height, $H_s$, of the SRB cloud. A criteria will be established to identify cases when calculated downwind concentrations and dosages are essentially zero. These cases are associated with a very low transport layer height relative to cloud
stabilization heights. The criteria would be of the form

\[ H_s - H_m > X \]

where X would be selected on the basis of a sub-set of diffusion calculations for various values of \( H_s - H_m \). The results of these calculations can be illustrated hypothetically as shown in Figure 2-6; based on this hypothetical example X would be chosen to be 400 meters.

The criteria would be used to eliminate trivial cases from the large parent sample.

2.4.2 Bivariate Normal Wind Distribution

In addition to the other analyses of the 1965 data, a study of the theory of the bivariate normal distribution and its use in summarizing the wind statistics at KSC was conducted. The theoretical equations and derivation supplied by O. E. Smith of NASA/MSFC were checked out, and the essential equations have been coded. Given the bivariate normal statistics, the program outputs the following:

- The distribution of wind direction.
- The distribution of wind speed given a specific direction (15).

Since the monthly bivariate normal statistics have already been calculated for KSC, the programs developed would be used as part of our operational forecasting scheme.

2.4.3 Development of Objective Methods for Estimation of Meteorological Input Variables for the Multilayer Diffusion Model

The development of objective methods for the estimation of meteorological input variables for the Multilayer Diffusion Model requires the following:
Figure 2-6. Hypothetical schematic representation of relation between air quality impact of SRB cloud and the difference between cloud stabilization height and transport layer height ($H_s - H_m$).
Establishment of a theoretical basis for the method,

Development of computer codes for calculation of the standard deviation of wind azimuth angle over a ten-minute period (SIGAR) and for specification of height of the transport layer ($H_m$),

Testing the computer codes for a climatological data sample,

Modification of the preprocessor to include the new codes.

The theoretical basis (16) for the method selected for estimating SIGAR is based on solution of equation

\[ \text{SIGAR} = \frac{\sigma_v}{U} = \frac{k f(B)}{\ln \frac{z}{z_o} - \psi(\text{RI})} \]  

(1)

where $\sigma_v$ = standard deviation of the lateral component of turbulence (m/sec)

$U$ = mean wind speed (m/sec)

$z_o$ = roughness length (m)

$k$ = Von Karman's constant = 0.4 (dimensionless)

The function of Richardson number, $\psi(\text{RI})$, for unstable conditions is

\[ \psi(\text{RI}) = 2 \ln \left[ \frac{(1+x)/2}{(1+x^2)/2} \right] + \ln \left[ \frac{(1+x^2)/2}{(1+x^2)/2} \right] \]

\[ - 2 \tan^{-1} x + \pi/2 \]  

(2)

where $x = (1-16\text{RI})^{1/4}$  

(2a)
For stable conditions (17)

\[ \Psi(\text{RI}) = \frac{7 \text{RI}}{1-7\text{RI}} \]  

(3)

The right side of Equation 1 was derived (16) by substitution of expressions for \( \sigma_v \) and \( U \) given below for the ratio \( \sigma_v/U \).

\[ \sigma_v = \mu*f(B) \]  

(4)

\[ U = \frac{\mu*}{k} [\ln \frac{z}{z_o} - \Psi(\text{RI})] \]  

(5)

where \( \mu* \) is the friction velocity, and the function \( f(B) \) is accurately approximated by fitting line segments to experimental measurements of the ratio \( \sigma_v/\mu* \) according to,

\[ f(B) = \begin{cases} 2.7 & B < -0.008 \\ 2.7 + 112(.008+B) & -0.008 \leq B < -0.00175 \\ 3.4 - 725.5(.00175+B) & -0.00175 \leq B < 0.0008 \\ 1.55 + 38.04(B-.0008) & 0.0008 \leq B < 0.029 \\ 2.35 + 5.43(B-.029) & 0.029 \leq B \end{cases} \]

(6)

The Richardson number, RI, is defined by

\[ \text{RI} = \frac{g \frac{\partial \theta}{\partial z}}{\left( \frac{\partial v}{\partial z} \right)^2} \]  

(7)

where \( g = \) acceleration of gravity (m/sec\(^2\))
\( t = \) absolute temperature (\(^\circ\)K)
\( \frac{\partial \theta}{\partial z} = \) vertical gradient of potential temperature (\(^\circ\)K/m)
\[ \frac{\partial V}{\partial z} = \text{vertical gradient of wind speed (1/sec)} \]

The quantity \( \frac{\partial \Theta}{\partial z} \) can be expressed as a function of pressure and vertical gradient of temperature according to

\[ \frac{\partial \Theta}{\partial z} = \frac{\Delta \Theta}{\Delta z} = \left( \frac{1000}{P} \right)^{2.88} \left( \frac{\Delta t}{\Delta z} + 0.098 \right) \tag{8} \]

where \( P \) is the pressure in millibars.

Since available wind measurements are not sufficiently accurate for estimation of the denominator in Equation (7), it is necessary to estimate \( R_I \) from measurements of the non-dimensional stability ratio, \( B \).

\[ B = \frac{g \bar{z}^2}{
 \left( \frac{\Delta \Theta}{\Delta z} \right) U^2 \tag{9} \]

where, \( \bar{z} \) = the geometric mean height (m) between the top and bottom of the layer considered (17)

\[ U = \text{mean wind speed in this layer (m/sec)} \]

The relation (17) between \( B \) and \( R_I \) is

\[ R_I = B \left[ \frac{1 \ln z / z_0 - \psi(R_I)}{\phi(R_I)} \right]^2 \tag{10} \]

where, \( \psi(R_I) = (1-16RI)^{-\frac{1}{2}} \) for unstable conditions \( (11) \)

\[ \phi(R_I) = \frac{1}{1 - 7RI} \text{ for stable conditions} \tag{12} \]
For stable conditions it can be shown that Equation (10) is a quadratic function of the parameter \( y \).

\[
y^2 + y/7kB^{\frac{3}{2}} - (k+1)/7k = 0
\]

where, \( y = (RI)^{\frac{3}{2}} \)

\[
k = \ln(z/z_o) - 1
\]

For unequal real roots, the root given by the following equation will result in physically realistic calculated values of \( RI \) over the expected range of measured values of \( B \).

\[
y = -\frac{1}{14k\sqrt{B}} + \frac{1}{2}\sqrt{\frac{1}{49k^2B} + \frac{4(k+1)}{7k}}
\]  

\( RI = y^2 \)

An additional constraint is required to assure that physically realistic values of \( RI \) are calculated for stable conditions; examination of Equation 12 reveals that a singularity exists for \( RI = 1/7 \). The singularity is eliminated by assuming \( \phi(RI) = \phi(0.137) \) for \( RI > 0.137 \). This constraint is implemented only in rare instances during extremely stable conditions. This problem is also evident in Golder's nomogram (17) for estimating \( RI \) from \( B \); the \( RI \) scale on the nomogram has a maximum value of \( 0.13 \).

For unstable conditions Equation 10 can be written as

\[
\frac{1-x^4}{16x^2 \left[ \ln z/zo + 0.50864 - 2[\ln(1+x)] - \ln(1+x^2) + 2 \tan^{-1}x \right]^2} - B = 0
\]

(14)

where \( x \) is given by Equation 2a. Equation 14 is solved by Newton's method.
The methodology for calculation of SIGAR can be summarized as follows:

- Calculate B from available tower or Rawinsonde data (Equation 9)
- Evaluate F(B) (Equation 7)
- Specify $z_0$. A reasonable first approximation is $z_0 = 0.25$ m (18)
- Solve Equation 13 or 14 (unstable or stable conditions) to obtain $R_I$ from Equation 11 or 13a respectively.
- Calculate SIGAR from Equation 1.

Preliminary calculations using Rawinsonde data and data constructed for the purpose of comparison with Cramer Co. SIGAR values (19) are given in Tables 2-8 and 2-9 respectively.

Table 2-6. Calculations of SIGAR using Rawinsonde data between the surface and the first standard pressure level (1,000 mb) with $z_0 = 0.25$ m.

<table>
<thead>
<tr>
<th>Date</th>
<th>SIGAR (deg)</th>
<th>$\Delta \theta/\Delta z$ ($^\circ$C/100m)</th>
<th>U (m/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>('69, 12Z, 0700 EST)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/1</td>
<td>10.8</td>
<td>-.075</td>
<td>13.</td>
</tr>
<tr>
<td>1/15</td>
<td>7.2</td>
<td>.98</td>
<td>8.</td>
</tr>
<tr>
<td>1/29</td>
<td>7.1</td>
<td>.93</td>
<td>11.</td>
</tr>
<tr>
<td>2/5</td>
<td>7.7</td>
<td>2.38</td>
<td>3.5</td>
</tr>
<tr>
<td>2/12</td>
<td>8.0</td>
<td>1.29</td>
<td>4.</td>
</tr>
<tr>
<td>('69, 00Z, 1900 EST)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/7</td>
<td>16.3</td>
<td>-.69</td>
<td>6.</td>
</tr>
<tr>
<td>7/14</td>
<td>17.2</td>
<td>-.88</td>
<td>6.</td>
</tr>
<tr>
<td>7/21</td>
<td>7.6</td>
<td>.83</td>
<td>4.</td>
</tr>
<tr>
<td>7/28</td>
<td>15.7</td>
<td>-.20</td>
<td>2.</td>
</tr>
<tr>
<td>8/4</td>
<td>14.5</td>
<td>-.90</td>
<td>14.</td>
</tr>
</tbody>
</table>
Table 2-7. Comparison of calculated SIGAR\(^2\) with values given by Dumbauld et al (19) (in parenthesis)

<table>
<thead>
<tr>
<th>Wind Speed at 18m (m sec(^{-1}))</th>
<th>STABILITY CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very Unstable ((\Delta T = -1.4^\circ C))</td>
</tr>
<tr>
<td>1-2</td>
<td>21.29</td>
</tr>
<tr>
<td></td>
<td>(25.)</td>
</tr>
<tr>
<td>2-4</td>
<td>18.13</td>
</tr>
<tr>
<td></td>
<td>(16.)</td>
</tr>
<tr>
<td>4-7</td>
<td>20.03</td>
</tr>
<tr>
<td></td>
<td>(12)</td>
</tr>
<tr>
<td>7-11</td>
<td>15.65</td>
</tr>
<tr>
<td></td>
<td>(10)</td>
</tr>
</tbody>
</table>

(1) \(\Delta T\) measured between 3m and 60m.

(2) SIGAR is the standard deviation of the wind azimuth angle measured over a 10-minute period.

(3) Very stable conditions cannot occur with KSC with such large wind speeds.
Testing of the computer code for SIGAR using the 1965 KSC Rawinsonde data has begun. Preliminary results for January and February data are summarized in Tables 2-8 and 2-9. Table 2-8 indicates that none of the computed values of SIGAR were less than 3 degrees, very few were greater than 18 degrees and most were between 6 and 9 degrees. Table 2-9 indicates that for a particular potential temperature gradient, SIGAR increases with decreasing wind speed. Table 2-9 should be expanded to cover more wind speed and potential temperature gradient intervals as the calculations based on all the 1965 data become available.

Table 2-8. Distribution of SIGAR computed from Rawinsonde Data (January, February 1965, 239 soundings)

<table>
<thead>
<tr>
<th>SIGAR (deg)</th>
<th>Percent Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;3</td>
<td>0</td>
</tr>
<tr>
<td>3-6</td>
<td>22.6</td>
</tr>
<tr>
<td>6-9</td>
<td>48.1</td>
</tr>
<tr>
<td>9-18</td>
<td>25.5</td>
</tr>
<tr>
<td>&gt;18'</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Table 2-9. Mean SIGAR (deg) as a Function of Potential Temperature Gradient for Two Wind Speed Intervals (January 1965, 26 cases)

<table>
<thead>
<tr>
<th>ΔΘ/ΔZ (°c/m)</th>
<th>U(m/sec)</th>
<th>&lt;.0017</th>
<th>-.0017</th>
<th>.0016</th>
<th>.0017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4-8</td>
<td>13.6</td>
<td>6.3</td>
<td>6.1</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td>2-4</td>
<td>17.5</td>
<td>*</td>
<td>*</td>
<td>7.7</td>
</tr>
</tbody>
</table>

*No Data
2.5 MODIFICATION TO THE UNIVAC 1108 VERSION OF THE REED DESCRIPTION AND THE CLOUD RISE PROGRAM

In the area of climatological assessment, one of the major tools is the NASA/MSFC REED Description (2). In the original mode of operation, a pre-processor program was required to read the meteorological data and calculate cloud rise and cloud location. This process has been automated so that the two programs are executed in one job stream. Instead of punching cards, the cloud rise program builds a disk file where each case processed is given a unique identifier. The REED program then executes with the capability of choosing any of the cases from the cloud rise file in any order. Additional flexibility is achieved by allowing the user to override any parameters set by the cloud rise program prior to the execution of the REED Description.

For the purposes of documentation and compact storage, the capability to produce a duplicate copy of all printer output on plot paper was added. This plotter output is much more suitable than printer output for 8½ by 11" documents, and is also more easily filed. In addition, the tapes from which these plots are made can be saved and used as data files from which additional calculations can be performed.

Finally, the capability to print a table summarizing the most critical parameters for each case in a particular run was added to the code. Thus in a run where many cases are processed, the user can quickly determine which cases are the more critical. This table can be conveniently used directly for documentation purposes.
2.5.1 **Screening Program Modification**

Modifications to provide additional capabilities for the Meteorological Data Screening Program* were completed. The MET Screening reads in cards, which indicate which soundings to search for, and reads from met data tapes, then it generates as output, plots, cards, and printout for each sounding processed.

The plots include a list of the cloud rise heights plus the following plots:

- Wind Speed versus Altitude
- Wind Direction versus Altitude
- Dry Bulb and Potential Temperature versus Altitude
- Temperature, Virtual Temperature, and Virtual Potential Temperature versus Altitude

The card output from the Screening Program is punched in the format needed for the pre-processor.

The printout has been expanded to include stability criteria. The data were tested against both TVA and AEC stability criteria. The results are printed in a table after the original output has been completed for each time. The following information is printed: The altitude interval, temperature interval, DTODS, AEC stability, potential temperature interval, DPTODZ, and TVA stability. DTODZ is defined as

\[
\frac{T_i - T_{i-1}}{Z_i - Z_{i-1}} \quad \text{where } i = 1 - \text{no. of altitudes}
\]

T is temperature (°C)

\[
Z_i \text{ is altitude (meters)}
\]

DPTODZ is defined as

\[
\frac{PT_i - PT_{i-1}}{Z_i - Z_{i-1}} \quad \text{PT is potential temperature}
\]

\[
\]

*Initially developed by Dr. Stephens and W. C. Campbell at MSFC.

2-37
The AEC and TVA stability criteria is listed in Table 2-10. This added printout aids in choosing the height of the surface transport layers needed for input into the cloud rise program.

Table 2-10. Stability Criteria

**ATOMIC ENERGY COMMISSION CRITERIA**

\[ X = \text{Gradient of Temperature} \]

<table>
<thead>
<tr>
<th>Classification</th>
<th>( X(\degree C/\text{Meter}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extremely unstable</td>
<td>( X &lt; -0.019 )</td>
</tr>
<tr>
<td>Moderately unstable</td>
<td>(-0.019 \leq X &lt; -0.017)</td>
</tr>
<tr>
<td>Slightly unstable</td>
<td>(-0.017 \leq X &lt; -0.015)</td>
</tr>
<tr>
<td>Neutral</td>
<td>(-0.015 \leq X &lt; -0.005)</td>
</tr>
<tr>
<td>Slightly stable</td>
<td>(-0.005 \leq X &lt; 0.015)</td>
</tr>
<tr>
<td>Moderately stable</td>
<td>(0.015 \leq X &lt; 0.040)</td>
</tr>
<tr>
<td>Extremely stable</td>
<td>(0.040 \leq X )</td>
</tr>
</tbody>
</table>

**TENNESSEE VALLEY AUTHORITY CRITERIA**

\[ Y = \text{Gradient of Potential Temperature} \]

<table>
<thead>
<tr>
<th>Classification</th>
<th>( Y(\degree C/\text{Meter}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstable</td>
<td>( \leq -0.0017)</td>
</tr>
<tr>
<td>Neutral</td>
<td>(-0.0017 \leq Y &lt; 0.0016)</td>
</tr>
<tr>
<td>Moderately stable</td>
<td>(0.0016 \leq Y &lt; 0.0070)</td>
</tr>
<tr>
<td>Very stable</td>
<td>(0.0070 \leq Y &lt; 0.0187)</td>
</tr>
<tr>
<td>Extremely stable</td>
<td>(0.0187 \leq Y )</td>
</tr>
</tbody>
</table>
2.5.2 Addition of New Vehicle and Updating of Constants

The characteristics of the newest solid motor in the Thor-Delta family of launch vehicles were added to the Multi-layer/pre-processor system. This new vehicle is known as the Thor-Delta 3914 and is the fifth vehicle that can be simulated by the code.

In conjunction with determining the values for constants associated with the 3914, the same constants were examined for the other four vehicles. These constants include the following:

- QC1, QC2, QC3 - total source output rates (g/sec) for the three types of launch respectively (i.e. normal, abnormal with one motor burning on the pad, abnormal where motors explode and burn on the ground).
- QT1, QT2, QT3 - total source strength (g) for the three types of launch respectively.
- HEATN, HEATM, HEATA - Heat output (cal/g) for the three types of launch respectively.
- a, b, c - Rocket rise parameters in the equation $T = az^b + c$ where T is the burn time and z is the altitude
- FRQ1 - Fractional distribution of material for HCl, CO, CO$_2$ and AL$_2$O$_3$.

Table 2-11 lists the preliminary values determined for these constants:
## Table 2-11. Preprocessor Program Constants

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Titan III C</th>
<th>Space Shuttle</th>
<th>Thor-Delta 2914</th>
<th>Minuteman II</th>
<th>Thor-Delta 3914</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC1</td>
<td>5.43752E6</td>
<td>1.52192E7</td>
<td>8.36068E5</td>
<td>4.68447E5</td>
<td>1.05755E6</td>
</tr>
<tr>
<td>QC2</td>
<td>2.71876E6</td>
<td>6.88296E6</td>
<td>9.09811E4</td>
<td>4.68447E5</td>
<td>1.48292E5</td>
</tr>
<tr>
<td>QC3</td>
<td>1.35938E6</td>
<td>3.44148E6</td>
<td>2.72943E5</td>
<td>1.17119E5</td>
<td>3.70731E5</td>
</tr>
<tr>
<td>QT1</td>
<td>3.26251E8</td>
<td>1.89479E9</td>
<td>2.88759E7</td>
<td>2.81068E7</td>
<td>6.70169E7</td>
</tr>
<tr>
<td>QT2</td>
<td>1.63125E8</td>
<td>8.56929E8</td>
<td>3.14229E6</td>
<td>2.81068E7</td>
<td>9.39861E6</td>
</tr>
<tr>
<td>QT3</td>
<td>3.26251E8</td>
<td>1.71385E9</td>
<td>1.88537E7</td>
<td>2.81068E7</td>
<td>4.69930E7</td>
</tr>
<tr>
<td>HEATN</td>
<td>2021.1</td>
<td>1479.7</td>
<td>1766.0</td>
<td>2055.9</td>
<td>1449.9</td>
</tr>
<tr>
<td>HEATM</td>
<td>1010.55</td>
<td>1062.35</td>
<td>1000.00</td>
<td>2055.9</td>
<td>1000.00</td>
</tr>
<tr>
<td>HEATA</td>
<td>1000.00</td>
<td>1000.00</td>
<td>690.0</td>
<td>1000.00</td>
<td>411.18</td>
</tr>
<tr>
<td>FRQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-HCl</td>
<td>.1931</td>
<td>.1782</td>
<td>.1218</td>
<td>.1977</td>
<td>.1589</td>
</tr>
<tr>
<td>-CO</td>
<td>.2665</td>
<td>.2021</td>
<td>.2055</td>
<td>.2380</td>
<td>.2783</td>
</tr>
<tr>
<td>-CO₂</td>
<td>.0222</td>
<td>.0286</td>
<td>.0156</td>
<td>.0318</td>
<td>.0331</td>
</tr>
<tr>
<td>-Al₂O₃</td>
<td>.2819</td>
<td>.2524</td>
<td>.2214</td>
<td>.2761</td>
<td>.1936</td>
</tr>
<tr>
<td>AA</td>
<td>.429580</td>
<td>.652213</td>
<td>.922156</td>
<td>.469982</td>
<td>1.245756</td>
</tr>
<tr>
<td>BB</td>
<td>.518422</td>
<td>.468085</td>
<td>.432703</td>
<td>.463333</td>
<td>.418095</td>
</tr>
<tr>
<td>CC</td>
<td>0.375</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

*Value used in report; other is up-dated reflecting latest result.
The original rocket rise equation \( T = az^b \) was modified to the form \( T = az^b + c \). The constant \( c \) was added to take into account the time lapse between ignition and lift-off. The parameters \( a \), \( b \), and \( c \) were obtained from least squares fits of empirical trajectory data. Plots of the trajectories generated by the old values and by the new values were made against the measured trajectories. The results are shown in Figures 2-7 through 2-11. For each of the original four vehicles, the trajectory generated by the new values is closer to the measurement than is the old trajectory.

Since the burn rate for solid propellant motors is influenced by the initial temperature of the propellant, the pre-processor program was modified to take into account this initial propellant temperature. A table of the mean temperatures at KSC for each month was added to the code. Based on the month in which meteorological data was taken, the default temperature is obtained from the table and used to compute a burn-rate factor (where 70° is the standard, yielding a burn-rate factor of one). The capability to over-ride this default table was also added to the code, so the initial propellant temperature, if known, can be input to the program.

Various runs were made with the UNIVAC 1108 REED Description/Cloud Rise system to check out all the modifications made to the code; however no production type runs have been performed.
Figure 2-7. Titan III C Trajectory

\[ T = a z^{b+c} \]
Figure 2-8. Space Shuttle Trajectory

\[ T = az^b + c \]
Figure 2-9. Minuteman II Trajectory

\[ T = az^b + c \]
Figure 2-10. Delta Thor 2914 Trajectory
\[ T = ax^b + c \]
Figure 2-11. Delta Thor 3914 Trajectory\(^{(21)}\)

\[ T = az^b + c \]
2.6 SURFACE DEPOSITION MODEL

Three major constituents in the SRB and SSME exhaust effluents are Al₂O₃, HCl, and H₂O. The HCl is of prime concern environmentally since it is potentially toxic in the gaseous phase and forms a strong acid in the aqueous phase. Two phases of Al₂O₃ can exist, the gamma phase which reacts strongly with HCl and the alpha phase which does not. The HCl reacts with and is absorbed by water droplets to form an aerosol. The formation of the aerosol, of course, reduces the concentration of gaseous HCl in the atmosphere. The aluminum oxide absorbs H₂O readily; it is used as a drying agent in laboratories. Rain falling through a cloud consisting of the rocket exhaust effluents and the entrained air can react chemically with the HCl and possibly the Al₂O₃ and can physically interact with the HCl aerosol and the Al₂O₃. Thus it can be seen that the Al₂O₃/HCl/H₂O system has a large number of physical and chemical interactions that can occur simultaneously. A consistent set of reactions and interactions must be developed to allow the calculation of the HCl and Al₂O₃ concentrations for a surface deposition model.

The phase of the aluminum oxide in the exhaust is not without question. Early work(25) indicated that the aluminum oxide present in a rocket exhaust is the alpha phase, which does not react with HCl. More recent data(26,27,28) has indicated that some of the gamma phase may be present. This may be important to the surface deposition depending on the aluminum particle size distribution in the rocket exhaust. The SRB exhaust will have relatively large particles; therefore, the amount of gamma phase will be less than found in small motor firings. Of course, for an equal weight, the
number and surface area for small particles is greater than for larger particles. The whole question of $\text{Al}_2\text{O}_3$ phase is undergoing intensive investigation and must be considered when developing a surface deposition model.

Using available experimentally measured $\text{Al}_2\text{O}_3$ particle size distributions (29) for solid propellant rocket motors of various sizes and making reasonable assumptions as to particle size growth as a function of throat diameter, a particle size distribution for the $\text{Al}_2\text{O}_3$ exhausting from the Space Shuttle SRB was determined. This is shown in Table 2-12. At the present time no realistic input to the REED Description surface deposition model is available for use; therefore, these effects, which may be significant for climatological predictions, have been neglected.

2.7 ABSORPTION AND SCAVENGING

Studies have been conducted on atmospheric scavenging of HCl which experimentally determined the washout coefficient (30,31). Washout involves several microprocesses, including the solubility of HCl in raindrops, the diffusion of HCl to the falling raindrops, and the physical parameters which characterize the rain. At higher relative humidities, washout of HCl aerosol must be considered in addition to the washout of gaseous HCl. The $\text{Al}_2\text{O}_3$ particles as well as salt or dust particles in the rocket exhaust may act as potential cloud droplet nuclei. The nucleating efficiency of $\text{Al}_2\text{O}_3$ particles is unknown at this time. The rain scavenging experimental results must be integrated into the surface deposition model.

The effects of absorption and scavenging, which may be significant for climatological predictions, have been neglected in this study because of the lack of a suitable, acceptable washout coefficient. (27,30,32,33)
Table 2-12

SRB Particle Size Distribution (29)

<table>
<thead>
<tr>
<th>Particle Diameter in Microns</th>
<th>Weight Percentage of the Particles of that Size Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 7.0</td>
<td>20.0</td>
</tr>
<tr>
<td>7.0 - 10.0</td>
<td>20.0</td>
</tr>
<tr>
<td>10.0 - 14.0</td>
<td>20.0</td>
</tr>
<tr>
<td>14.0 - 16.0</td>
<td>20.0</td>
</tr>
<tr>
<td>16.0 - 23.0</td>
<td>20.0</td>
</tr>
</tbody>
</table>
2.8 CONCLUSIONS

The long term objective of this study is to establish the relation between weather patterns of various scale and the environmental impact of the Space Shuttle exhaust effluents. To date, the synoptic weather patterns have been categorized and their relative frequency of occurrence have been calculated.

Concurrently the tools for calculating air quality assessments for large samples of KSC meteorological data have been developed. A large sample of Rawinsonde data are available for definition of the variability of calculated air quality assessments over time scales as small as six hours. This variability is associated with such phenomena as the development of the sea breeze and ground based stable layers. These phenomena strongly influence the critical meteorological input variables to the diffusion model.
3. EXHAUST CHEMISTRY

The calculation of the heat content of the plume, or more exactly the heat content of the rocket exhaust effluents, taking into account interactions with the ambient environment, is a well-defined problem. The problem has been attacked for many years by the propulsion community and a set of standard techniques have been devised and published for liquid engine performance and analysis by the Interagency Chemical Rocket Propulsion Group – Joint Army, Navy, NASA, Air Force, (ICRPG-JANNAF) Performance Standardization Working Group (34, 35, 36). The state-of-the-art of analysis for solid motors is not yet as advanced but an ICRPG-JANNAF Solid Performance Working Group has begun work.

The available techniques were adequate for analyzing the plume from the liquid propellant SSME rocket engine and the solid propellant SRB motors. The value of the effective heat release and the exhaust species concentrations were quantitatively satisfactory for both propulsion devices. During this study only the Space Shuttle SRB exhaust effluents were studied in detail. Solid propellant rocket motors have the phenomena of two-phase flow occurring in the combustion chamber, nozzle, and plume. The two phases are not in thermal or velocity equilibrium. In general, the particles, in this case solid and liquid aluminum oxide, are traveling slower than the gas, are at a higher temperature than the gas, and are at a greater flow angle than the gas. These phenomena make the characteristics of a two-phase flow field different than that of a single-phase flow field such as occurs in the liquid propellant SSME.
3.1 TWO-PHASE FLOW PHENOMENA

The analysis of the two-phase flow in the SRB rocket nozzle started with a one-dimensional thermochemical analysis of the solid propellant. As seen in Figure 3-1, when solid propellant combusts, the combustion products are at some pressure, $P_c$, and some flame temperature, $T_c$. The chamber pressure history is governed principally by the amount of burning surface exposed. The desired amount of burning surface (chamber pressure) can be obtained by the geometry of the propellant grain. Figure 3-2 shows the Space Shuttle altitude, Mach number, and Solid Rocket Motor chamber pressure history for the first 70 seconds of flight. As can be seen, the chamber pressure varies from 825 to 580 psia during this portion of the flight. With a knowledge of the propellant composition and the chamber pressure, the flame temperature and the concentrations of the combustion products were calculated as shown in Table 3-1. The flame temperature and the combustion products as a function of time (velocity and altitude are then known) were needed for input to subsequent steps. The calculations were performed on the NASA UNIVAC 1108 with a program written by NASA-Lewis Research Center (38) and modified by SAI.

By means of two-phase characteristic theory, the supersonic portion of the flow field of the SRB nozzle and plume was determined. With reference to Figure 3-1, the supersonic portion is bounded roughly upstream by the nozzle throat and downstream by the plume boundary. The nozzle analysis portion of the program basically terminates calculation along the last left-running characteristic, identified on Figure 3-1. This surface is significant in that no disturbance downstream of it will affect the pressure field along the nozzle wall. The program originally written by TRW personnel (38) and extensively modified by SAI (40) yields vital pieces of information along the last
Figure 3-1. Solid Rocket Motor, Nozzle, and Plume
left-running characteristic which are needed for subsequent steps in the modeling.

A two-phase flow field is dissipative and non-equilibrium in nature. There is, therefore, an entropy rise down the flow field and an entropy gradient radially across the flow fields since the particles and gas have a different history at every point in the flow field. The entropy rise and the loss in total pressure can be calculated from local flow properties. Figure 3-3 shows the total pressure loss and gradient along the last left-running characteristic for the Space Shuttle SRB nozzle with a single particle size of 12.0 micron diameter which represents an average particle size. The pressure loss varies from about 27 to 55 percent of the chamber pressure; thus, the species and energy content of the exhaust will vary across the nozzle exit. Because of the wide variation in properties across the exit, an integration scheme was incorporated into the program which integrates the mass flow and energy content and computes the average for a gross value of the energy content of the exhaust as it leaves the nozzle. The energy content of the exhaust was assumed to be composed of two parts: the sensible enthalpy and the kinetic energy of the gas. For a SRB operating at 780 psia chamber pressure, the average integrated value of the heat content of the plume is 2125 calories per gram. Figure 3-4 is a schematic of the SRB nozzle. The chamber pressure chosen, 780 psia, is an average value representative of the SRB when it is close to the launch pad, 0-3000 meters altitude.

3.2 AFTERBURNING AND MIXING ANALYSIS

Solid propellants normally are formulated to have an exhaust composition rich in underoxidized species, i.e., the carbon, C, is preferentially in the form of carbon monoxide, CO, rather than carbon dioxide, CO₂. This formulation technique gives higher specific impulse for the propellant. A
Figure 3-3. Ratio of Stagnation to Chamber Pressure vs Radial Distance Along Last Left-Running Characteristic

$R_t = 27.215$ inches
SPACE SHUTTLE SRB NOZZLE

LIMITING PARTICLE STREAMLINE
$D_p = 12.0$ micron

Figure 3-3. Ratio of Stagnation to Chamber Pressure vs Radial Distance Along Last Left-Running Characteristic
Figure 3-4. DIMENSIONLESS AXIAL COORDINATE (X/Rₜ)
SRB Nozzle Diagram

Rₜ = 27.215 inches
R_c/Rₜ = 2.00
Nozzle Exit

Center Line
jet exhausting into a stationary or moving atmosphere tends to entrain and mix the atmosphere with the jet exhaust.

When a hot exhaust with underoxidized species mixes with ambient air, the possibility for afterburning exists. This condition has been noted on several launch vehicles, and digital computer programs have been written to describe the phenomena with varying degrees of success (41-43). Several of the programs appear generally suitable for use in the proposed study. Based on such factors as ease of input, accuracy, computing time, and calculation technique, the program written by AeroChem Research Laboratories, Inc. (43) was chosen. Figure 3-5 shows the plume afterburning schematic as it applied to this situation. The only modification necessary for the program to be used for the problem under consideration is in the description of the the initial data line. The initial data line for the original program is assumed to be radial, normal to the axis at the exit of the nozzle. All species and the velocity at each grid point must be input. The output of the two-phase analysis program is along the last left-running characteristic. The velocity, entropy, and stagnation pressure are known at every point on the characteristic but not the species; therefore, a technique was devised which would fill the gap between the last left-running characteristic and the needed initial value line and which would calculate the species along the initial value line. There exists in general use in the aerothermodynamic community in this country, a computer program known as PLIMP (44) which calculates and outputs the species concentration, pressure, temperature, and velocity fields on surfaces immersed in a plume; therefore, if a flat plate is placed normal to the axis at the exit of the nozzle, all necessary quantities will be obtained.

The Aerochem mixing program calculated the required values of species concentration and amount of entrained air simultaneously. Stedman (27) in his work estimated the
Figure 3-5. Plume Afterburning Schematic
amount of mixed or entrained air from the work of Hart (45) and assumed uniform mixing and chemical equilibrium in the cloud. The Aerochem program not only estimated the amount of entrained air, which is not uniform across the jet, but also determined the species concentrations using finite-rate chemistry. This technique thus detailed the species, the reaction rates, and the temperature and pressure radially across the exhaust as well as in a downstream direction from the nozzle exit. An inventory of the constituents was thus maintained. Table 3-2 lists the reaction scheme utilized in this study. The scheme models the chlorine species production and destruction in detail. Figures 3-6 and 3-7 show the Space Shuttle SRB exhaust effluents as a function of distance from the nozzle exit. Table 3-3 shows the exhaust effluent weight fractions as a function of distance from the nozzle exit.

3.3 OTHER LOSSES

It should be noted a number of potentially important effects were neglected. The first effect neglected was the injection of water into the exhaust. Since the study was initiated, the decision was made to inject large quantities of water into the exhaust as a noise suppression technique. The water is not only to be injected when the Space Shuttle is sitting on the launch pad but the injection will continue until it clears the launch pad. Due to the afterburning and the large number of hot particles, a large luminous plume is formed; therefore, radiation losses may be important. Hart (45) using geometric flight and launch hardware radiant flux estimates, states that the radiation loss may be as large as one-fourth the total heat content. If this estimate is correct, radiation loss calculations are imperative. The radiation data for the exhaust effluents have been collected and tabulated, but the entire calculation has not yet been performed. During the time the SSME's are building up thrust and until shortly after SRB ignition, the Space Shuttle is held onto the launch pad. During this time and even after liftoff, for
Table 3-2
AFTERBURNING ANALYSIS
REACTIONS BEING CONSIDERED

<table>
<thead>
<tr>
<th></th>
<th>Reaction</th>
<th></th>
<th>Reaction</th>
<th></th>
<th>Reaction</th>
<th></th>
<th>Reaction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HCL + OH</td>
<td>=</td>
<td>H2O + CL</td>
<td>20</td>
<td>H + HO2</td>
<td>=</td>
<td>OH + OH</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>H + HCL</td>
<td>=</td>
<td>CL + H2</td>
<td>21</td>
<td>H + O2   + M</td>
<td>=</td>
<td>H2O2 + M</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OH + CL</td>
<td>=</td>
<td>HCL + O</td>
<td>22</td>
<td>O + H2</td>
<td>=</td>
<td>OH + H</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CL + HO2</td>
<td>=</td>
<td>HCL + O2</td>
<td>23</td>
<td>O + HO2</td>
<td>=</td>
<td>OH + O2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CLO + OH</td>
<td>=</td>
<td>HO2 + CL</td>
<td>24</td>
<td>OH + HO2</td>
<td>=</td>
<td>O2 + H2O</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>H + Cl2</td>
<td>=</td>
<td>HCL + CL</td>
<td>25</td>
<td>H2 + HO2</td>
<td>=</td>
<td>H2O + OH</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>O + HCL</td>
<td>=</td>
<td>CL + OH</td>
<td>26</td>
<td>H + OH   + M</td>
<td>=</td>
<td>H2O + M</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CL + O3</td>
<td>=</td>
<td>CLO + O2</td>
<td>27</td>
<td>H + HO2</td>
<td>=</td>
<td>H2 + O2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CL + Cl  + M</td>
<td>=</td>
<td>CL2 + O2</td>
<td>28</td>
<td>OH + H2</td>
<td>=</td>
<td>H2O + H</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>O + Cl  + M</td>
<td>=</td>
<td>CLO + M</td>
<td>29</td>
<td>N + O2</td>
<td>=</td>
<td>NO + O</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CLO + H</td>
<td>=</td>
<td>HCL + O</td>
<td>30</td>
<td>NO + O   + M</td>
<td>=</td>
<td>NO2 + M</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>O + CLO</td>
<td>=</td>
<td>CL + O2</td>
<td>31</td>
<td>NO + CLO</td>
<td>=</td>
<td>CL + NO2</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>H + Cl  + M</td>
<td>=</td>
<td>HCL + M</td>
<td>32</td>
<td>NO + O3</td>
<td>=</td>
<td>NO2 + O2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>O3 + O</td>
<td>=</td>
<td>O2 + O2</td>
<td>33</td>
<td>NO2 + H</td>
<td>=</td>
<td>NO + OH</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>O + O   + M</td>
<td>=</td>
<td>O2 + M</td>
<td>34</td>
<td>N + NO</td>
<td>=</td>
<td>N2 + O</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>O + H   + M</td>
<td>=</td>
<td>OH + M</td>
<td>35</td>
<td>CO + OH</td>
<td>=</td>
<td>CO2 + H</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>H + H   + M</td>
<td>=</td>
<td>H2 + M</td>
<td>36</td>
<td>CO + O   + M</td>
<td>=</td>
<td>CO2 + M</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>O + O</td>
<td>=</td>
<td>H2O + O</td>
<td>37</td>
<td>CO + HO2</td>
<td>=</td>
<td>CO2 + OH</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>H + O2</td>
<td>=</td>
<td>OH + O</td>
<td>38</td>
<td>NO + CLO</td>
<td>=</td>
<td>CL + NO2</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3-6. Space Shuttle Solid Rocket Motor Exhaust Effluents
Figure 3-7. Space Shuttle Solid Rocket Motor Exhaust Effluents
Table 3-3

Space Shuttle SRB Exhaust Effluents
Concentrations in Weight Percent

<table>
<thead>
<tr>
<th>Effluents</th>
<th>Distance From Nozzle Exit - Feet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Al₂O₃</td>
<td>30.32</td>
</tr>
<tr>
<td>CO</td>
<td>24.36</td>
</tr>
<tr>
<td>CO₂</td>
<td>3.33</td>
</tr>
<tr>
<td>Cl</td>
<td>0.246</td>
</tr>
<tr>
<td>ClO</td>
<td>0.006</td>
</tr>
<tr>
<td>Cl₂</td>
<td>0.008</td>
</tr>
<tr>
<td>HCl</td>
<td>21.41</td>
</tr>
<tr>
<td>H₂</td>
<td>2.09</td>
</tr>
<tr>
<td>N₂</td>
<td>8.78</td>
</tr>
<tr>
<td>NO</td>
<td>0.001</td>
</tr>
<tr>
<td>O₂</td>
<td>0.004</td>
</tr>
</tbody>
</table>
a short time, the exhaust effluents are flowing into the flame trench where they are mixed with a large amount of water and ducted away. This mass and energy ducted away represents a possible loss to the ground cloud which may be important. This portion of the problem has not yet been attacked. This study has concentrated primarily on the SRB exhaust effluents and has essentially neglected the SSME exhaust and the problem of the impingement and mixing between the SRB and SSME exhaust plumes. The SSME exhaust effluents have been calculated and are shown in Table 3-4.

3.4 CONCLUSIONS

This study has developed a technique that allows the exhaust effluent chemistry for the SRB to be determined with a state-of-the-art analysis. At this point the basic exhaust effluents have been calculated but a number of important losses such as plume impingement, radiation, flame trench, and water injection have not been addressed. The effluents from the SSME have been determined but the chemical and physical interactions between the two plumes has not been studied.
Table 3-4

SSME Exhaust Effluents

Engine Conditions

\[ P_c = 3000 \text{ psia} \]
\[ O/F = 6 \]
\[ A_e/A_t = 77.5 \]

<table>
<thead>
<tr>
<th>Species</th>
<th>Location in Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chamber</td>
</tr>
<tr>
<td></td>
<td>Concentrations in Mole Fractions</td>
</tr>
<tr>
<td>H</td>
<td>0.0270</td>
</tr>
<tr>
<td>H\textsubscript{2}</td>
<td>0.2477</td>
</tr>
<tr>
<td>H\textsubscript{2}O</td>
<td>0.6831</td>
</tr>
<tr>
<td>O</td>
<td>0.0024</td>
</tr>
<tr>
<td>OH</td>
<td>0.0373</td>
</tr>
<tr>
<td>O\textsubscript{2}</td>
<td>0.0026</td>
</tr>
</tbody>
</table>
4. CONVERSION PROGRAMS

In order to allow for the processing of various types of data, received from numerous sources (i.e., Marshall Space Flight Center, Kennedy Space Center, Vandenberg AFB, Point Mugu, Asheville, etc.) and generated on several different computer configurations (i.e., IBM 360, IBM 370, IBM 7094, UNIVAC 1108, CDC 3300 etc.); there becomes a specific need for software which provides the capability of converting the various and voluminous amount of data into the proper BCD/EBCDIC+ASCII character set and record format to make it compatible to the different computer systems (i.e., IBM, UNIVAC, REEDA), upon which the data will be processed by a variety of programs. The data and programs must be available on all NASA/MSFC machines since no machine outage should cause a lack of monitoring capability.

In this section various conversion programs that were developed are discussed. Section 4.1 describes the conversion programs which have generic applications while Section 4.2 discusses conversion programs developed for individual cases. The program listings are given in the Appendix.

4.1 CONVERSION PROGRAMS (GENERIC)

The most efficient means to load data or software on the REEDA System, which was generated on other computer configurations, is to generate a magnetic tape compatible with the REEDA System. The following is a list of the requirements that all magnetic tapes must satisfy to be usable on the REEDA System:

- 9-track magnetic tape
- 800 bits per inch
- Odd parity
- 7-bit ASCII (The 8th bit is always off; this limits the character set to 128 combinations)

however, most (if not all) of the data and software programs being processed on the REEDA System received from other computer
installations were recorded on magnetic tape in a format not compatible with the REEDA System. Thus, various conversion programs were generated to convert data recorded on IBM, UNIVAC, CDC, etc., computers to a usable format.

4.1.1 IBM 370/360 BCD + ASCII Conversion Programs

A conversion program was developed to convert data or programs recorded in BCD (on cards or magnetic tape) to a usable REEDA System ASCII character set. This program is written in IBM assembly language and will execute on either the IBM 370 or 360 configuration. It will accept as input either a 7-track or 9-track tape, or punched cards and convert each BCD character into a 7-bit ASCII character compatible with the REEDA System. This converted data is written onto a magnetic tape for REEDA utilization (i.e., 9-track, 800 BPI, ODD parity). An example flow of the conversion process is given in Figure 4-1. Note, only the control cards change when running this program on the IBM 370 or IBM 360.

![Diagram of conversion process]

Figure 4-1. IBM 370/360 BCD + ASCII Conversion Process
4.1.2 IBM 370/360 EBCDIC → ASCII Conversion Programs

A conversion program was developed to convert data or programs recorded in EBCDIC (on cards or magnetic tape) to a usable REEDA System ASCII character set. This program is identical to the BCD → ASCII conversion program except all EBCDIC characters are converted to ASCII. As shown in Figure 4-1, input can be either cards or magnetic tape with the output being a REEDA compatible 9-track ASCII tape. Once again only the control cards change from the IBM 370 and IBM 360 programs.

4.1.3 UNIVAC 1108 BCD → ASCII Conversion Program

A conversion program was written in UNIVAC assembly to allow for the conversion of BCD record data on the UNIVAC 1108. As with IBM conversion, data is accepted from either cards or 7-track or 9-track magnetic tape. Each character is then converted to the corresponding 7-Bit ASCII character. A REEDA System compatible 9-track ASCII tape is generated as shown in Figure 4-2.

![Diagram of UNIVAC 1108 BCD → ASCII Conversion Process](image-url)
4.1.4 UNIVAC 1108 EBCDIC → ASCII Conversion Program

A conversion program was also written in UNIVAC assembly language to allow for the conversion of EBCDIC recorded data on the UNIVAC 1108. This program is identical to the BCD + ASCII conversion program except all EBCDIC characters are converted to ASCII. As shown in Figure 4-2, input can be either punched cards or magnetic tape with the output being a 9-track ASCII REEDA compatible tape.

4.2 CONVERSION PROGRAMS (SPECIFIC)

All of the generic BCD/EBCDIC conversion programs allow data to be converted from card/tape to REEDA System compatibility in a one-pass operation. However, the conversion programs were developed with a prerequisite that all data records be 80 characters long (i.e., card size). Thus, in the event records being converted (from tape) are not card images, that is, longer or shorter than 80 characters, a pre-processor is needed to reformat the data into 80 character records. This data can then be used as input into the BCD/EBCDIC conversion programs as shown in Figure 4-3.

Here the reformatter programs were developed to reformat various data from KSC, JSC, Pt. Mugu, Vandenberg AFB, etc., generated on IBM 7094, IBM 360, CDC 3300, UNIVAC 1108, etc., computers to REEDA compatibility. These programs are discussed in Chapter 5.
4.2.1 HP EBCDIC + ASCII Conversion Program

A conversion program was written in HP assembly language to translate IBM 9-track, ODD parity, EBCDIC recorded tapes to the compatible HP format. The program is capable of converting all IBM EBCDIC characters into their 7-Bit ASCII equivalent. Each 32 bit IBM integer is translated to a 16 bit HP integer, and 32 bit IBM real numbers are translated into the HP 32 bit real number format. For this program, the user must define the record lengths and blocksize to the conversion program which then translate the tape as it is being processed. Since only one tape drive exists for the current REEDA configuration, it is not possible to convert the entire tape and rewrite it to another tape for subsequent processing, thus the UNIVAC 1108 and IBM 370/360 conversion programs prove more efficient in most instances.

4.2.2 1965 KSC Rawinsonde Data Conversion Program

The conversion of the 1965 KSC rawinsonde data tapes (18 tapes, four recordings a day for twelve months) for REEDA usability was performed. The BCD to ASCII conversion program for the UNIVAC 1108 was utilized to convert from 7-track to the 9-track REEDA format. In initial attempts to process the data, once it was converted to REEDA System format, it was noted (see Table 4-1) that a non-standard method of recording negative numbers was used when recording the original data. That is, a negative number was represented by over punching an (11) in the right most digit of the variable field. Thus, for the numeric values of 0 - 9 together with an (11) punch would give the visual representation as follows:
### Table 4-1. Representative Rawinsonde Data

<table>
<thead>
<tr>
<th>ALT FT</th>
<th>WDIR</th>
<th>WKTS</th>
<th>TEMP</th>
<th>DEW PT</th>
<th>PRESS</th>
<th>RH</th>
<th>AB HUM</th>
<th>DENIRVSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>160</td>
<td>08</td>
<td>194</td>
<td>178</td>
<td>10691</td>
<td>99</td>
<td>1504</td>
<td>11925</td>
</tr>
<tr>
<td>001000</td>
<td>172</td>
<td>035</td>
<td>199</td>
<td>182</td>
<td>09750</td>
<td>89</td>
<td>1539</td>
<td>11497</td>
</tr>
<tr>
<td>002000</td>
<td>180</td>
<td>040</td>
<td>184</td>
<td>154</td>
<td>09410</td>
<td>82</td>
<td>1301</td>
<td>11163</td>
</tr>
<tr>
<td>003000</td>
<td>193</td>
<td>036</td>
<td>173</td>
<td>143</td>
<td>09082</td>
<td>82</td>
<td>1210</td>
<td>10818</td>
</tr>
<tr>
<td>004000</td>
<td>198</td>
<td>035</td>
<td>157</td>
<td>138</td>
<td>08764</td>
<td>87</td>
<td>1177</td>
<td>10496</td>
</tr>
<tr>
<td>005000</td>
<td>200</td>
<td>035</td>
<td>140</td>
<td>118</td>
<td>08455</td>
<td>86</td>
<td>1040</td>
<td>10194</td>
</tr>
<tr>
<td>006000</td>
<td>203</td>
<td>035</td>
<td>119</td>
<td>099</td>
<td>08154</td>
<td>86</td>
<td>0923</td>
<td>09907</td>
</tr>
<tr>
<td>007000</td>
<td>208</td>
<td>036</td>
<td>093</td>
<td>073</td>
<td>07863</td>
<td>84</td>
<td>0781</td>
<td>09632</td>
</tr>
<tr>
<td>008000</td>
<td>214</td>
<td>034</td>
<td>074</td>
<td>054</td>
<td>07580</td>
<td>87</td>
<td>0693</td>
<td>09367</td>
</tr>
<tr>
<td>009000</td>
<td>218</td>
<td>036</td>
<td>064</td>
<td>01M</td>
<td>07305</td>
<td>58</td>
<td>0438</td>
<td>09076</td>
</tr>
<tr>
<td>010000</td>
<td>222</td>
<td>038</td>
<td>055</td>
<td>04R</td>
<td>07038</td>
<td>47</td>
<td>0330</td>
<td>08779</td>
</tr>
<tr>
<td>011000</td>
<td>224</td>
<td>040</td>
<td>039</td>
<td>03P</td>
<td>06780</td>
<td>57</td>
<td>0361</td>
<td>08503</td>
</tr>
<tr>
<td>012000</td>
<td>228</td>
<td>041</td>
<td>026</td>
<td>09J</td>
<td>06531</td>
<td>42</td>
<td>0249</td>
<td>08233</td>
</tr>
<tr>
<td>013000</td>
<td>232</td>
<td>041</td>
<td>005</td>
<td>11R</td>
<td>06288</td>
<td>33</td>
<td>0194</td>
<td>07931</td>
</tr>
<tr>
<td>014000</td>
<td>238</td>
<td>040</td>
<td>02K</td>
<td>13P</td>
<td>06053</td>
<td>41</td>
<td>0170</td>
<td>07773</td>
</tr>
<tr>
<td>015000</td>
<td>244</td>
<td>044</td>
<td>04P</td>
<td>14P</td>
<td>05825</td>
<td>45</td>
<td>0157</td>
<td>07551</td>
</tr>
<tr>
<td>016000</td>
<td>250</td>
<td>047</td>
<td>07K</td>
<td>11P</td>
<td>05603</td>
<td>71</td>
<td>0204</td>
<td>07327</td>
</tr>
<tr>
<td>017000</td>
<td>255</td>
<td>051</td>
<td>09P</td>
<td>11N</td>
<td>05389</td>
<td>87</td>
<td>0208</td>
<td>07114</td>
</tr>
<tr>
<td>018000</td>
<td>258</td>
<td>056</td>
<td>12M</td>
<td>13!</td>
<td>05179</td>
<td>95</td>
<td>0186</td>
<td>06909</td>
</tr>
<tr>
<td>019000</td>
<td>261</td>
<td>058</td>
<td>140</td>
<td>200</td>
<td>04976</td>
<td>62</td>
<td>0105</td>
<td>06698</td>
</tr>
<tr>
<td>020000</td>
<td>261</td>
<td>057</td>
<td>15P</td>
<td>22J</td>
<td>04780</td>
<td>58</td>
<td>0087</td>
<td>06464</td>
</tr>
</tbody>
</table>
Thus a program was developed to process the data tapes to restore the number back to usable numeric notation. These tapes were then used as input to the REEDA System to build a "single tape" data base containing all pertinent information from the existing 18 tapes. The "single tape" data base allows the user easier/faster access to any/all data which he desires to process, thus eliminating the need to keep a library of 18 tapes and the processing of data which is not desired (See Figure 4-4).

The "single tape" data base was created by processing each of the 18 tapes and eliminating all data above 20,000 feet in altitude for the standard and mandatory levels, thus eliminating a large portion of the data. This data base was updated after each tape was processed with an EOF (end-of-file) mark inserted at the end of each month. Thus a user can easily access any month from the "single tape" data base by skipping the appropriate number of files. It must be pointed out that the initial idea of a "single tape" data base actually turned out to be two tapes containing all selected information from the original 18 tapes. The first tape contains JAN - JUN 1965, while the second tape contains JUL - DEC 1965.

It should also be noted that each of the 18 tapes being processed require about 2 hours to process on the REEDA System. Thus the initial creation of the "single tape" data
Figure 4-4. 1965 Rawinsonde Data Base Generation and Processing
base was quite time consuming, yet since it occurred only once, while the end product (i.e., "single tape" data base) is being utilized quite frequently. The overall effect of the "single tape" data base for the REEDA System is to allow the user to feasibly process all or as much of the 1965 KSC Rawinsonde data as efficiently and as fast as possible and to eliminate the handling of unneeded and unusable data.

Various programs are now being developed to extensively process the 1965 data, such as MOD3B, METPL, and STAN5 which are all documented in Chapter 5.

4.2.3 1974 Vandenberg Rawinsonde Data Conversion Program

The 1974 Vandenberg AFB Rawinsonde data tape contained two soundings per day (0000Z and 1200Z) for the entire year. The initial task was to convert the data into a usable format for the REEDA System. The original tape was generated on an IBM 360/44 and had variable length/variable block size records with half word alignment. A preprocessor was written in FORTRAN to restructure the data into fixed length records to be used as input into the IBM 370 ECBDIC + ASCII conversion program. It was discovered that two types of data records existed on the tape, 1) PIBAL and 2) Rawinsonde. However, neither data record contained all the information that was required to process the data using the REEDA diffusion model program MOD3A. The PIBAL record contained pressure, altitude, wind speed, wind direction but not temperature. The Rawinsonde records contained pressure, altitude, temperature, dew point, but not wind speed or wind direction. Subsequently, code was generated to merge the two records by means of various interpolations and calculations. The program computed the best possible values at the nearest altitude, pressure, and temperature. Figure 4-5 gives a brief flow of operations for processing the 1974 Vandenberg AFB data.
Once the data was converted to a usable format for the REEDA System, the program MOD3B was used to process the data. It should be noted that the two soundings per day for the entire year of 1974 were contained on the initial Vandenberg AFB tape. Some 48 cases roughly a week apart at 1200Z hours were processed. The program MOD3B is identical to the program MOD3A except it operates on the Plasmascope, which is interfaced into the REEDA System. It allows faster processing due to the use of "Touch Panel" program options. The output of MOD3B was 48 center line concentration plots and 48 isopleth plots (see Figure 4-6 and 4-7 respectively). This same data will also be processed on the REEDA System utilizing the new version of the program MOD3A.
Figure 4-6. 1974 Vandenberg AFB Centerline Concentration and Dosage Plot
Figure 4-7. Vandenberg AFB Isopieth Contour Plot
4.2.4 1964 - 1970 Jimsphere Data Conversion Program

A conversion program was written in FORTRAN to convert three meteorological data tapes, (1964 - 1966 KSC, 1967 - 1970 KSC, 1965 - 1970 Point Mugu) containing Jimsphere wind data, to the REEDA System compatibility. The tapes were initially created on an IBM 7094 with 36 bit word and data written in both fixed point and floating point binary. Each record contained 298 words with 20 such records per file. Additionally, each tape contained from 266 to 294 files. The decision was made to only extract and convert the needed data to eliminate the cumbersome task of processing over and around data not needed for calculations. Only the time, date, altitude, wind direction, and scalar wind speed was deciphered from the original data. It should be noted that the wind speed and wind direction were recorded at equal intervals in altitude from 25 meters to 20,000 meters. Thus, some 800 data points for both wind speed and wind direction were recorded for every Jimsphere profile.

The conversion program was written for the UNIVAC 1108 utilizing both ENTRAN and ENCODE features to convert the tapes into a format usable by the previously built EBCDIC and ASCII program. This encompassed converting from 36-bit to 16-bit HP word size, restructuring data into 80 character fixed length records, eliminating unwanted data, and then converting to ASCII format as shown in Figure 4-8.

As can be seen, a program to process the Jimsphere wind data on the REEDA System was created called JIMPL which will be discussed in Chapter 5.
Figure 4-8. Jimsphere Data Conversion and Processing
4.3 CONCLUSIONS AND RECOMMENDATIONS

These conclusions and recommendations are based on the conversion programs documented in Sections 4.1 and 4.2 respectively. In reviewing the generic conversion programs that were developed, it seems that they have proven quite satisfactory in providing a mean for converting non-standard REEDA formatted data into a usable form. Data generated on almost any other computer configuration, either 7-track or 9-track, either BCD or EBCDIC, either fixed or variable length records, can be made compatible to the REEDA System via one or a combination of two or more of the conversion programs that have been developed. However, it is still probable that data will be acquired that cannot be directly converted by using just the available conversion programs to date. Consequently, additional conversion programs undoubtedly will be developed as required.

It is also recommended that in most instances, tape reformatting and tape converting be conducted on a large scale computer configuration where multiple tape drivers and faster operating speeds are available.
5. INTERACTIVE REEDA PROGRAMS

In this section all of the applicable interactive REEDA programs are discussed. A brief description, along with current and future applications of each program is given. The following is a list of all the current REEDA programs which are discussed in Sections 5-1 thru 5-7.

- MOD3A
- MOD3B
- METPL
- STAN5
- MIXH
- JWSPL
- JWDPL
- JIMPS
- SKEW T (Version I & II)

5.1 MOD3A

The HP 9820 breadboard version of the REED Description, Model 3, previously used to monitor launches (46-49) has been rewritten, liberally commented, and made research operational on the REEDA System as an interactive program to test human factors and provide a real-time research capability for surface air quality predictions. The program asks questions of the user and accepts answers in English words and phrases. Using an X-Y plotter, it draws concentration and dosage versus distance plots as well as isopleth contour plots. The equations used for the cloud rise and diffusion are in an extremely simplified form and are being expanded to give a more accurate representation of the cloud mechanics and the diffusion process. This version does not permit diffusion calculations aloft, does not allow for options like surface absorption, rain scavenging, or Al$_2$O$_3$ deposition.

*These have been merged into the NASA/MSFC REED Diffusion Model Program Version I.
The distinct advantages of having the diffusion model operational on the REEDA System are two-fold. First, because the system is dedicated, the program can be run in almost real-time, thereby allowing last minute analysis and decisions to be made. Secondly, because of the interactive nature of the program, it is not necessary to have a trained computer person run it. Any scientific person knowledgeable in diffusion theory can, with a brief orientation, successfully operate the program. Knowledge of diffusion theory is required because SIGAR and the top of the transport layer determination calculations have not yet been automated.

5.2 MOD3B

A version of MOD3A, called MOD3B, has been written for the REEDA System to use the Plasmascope installed on the system. Because of the Touch Panel feature on the scope, it is easier for the user to answer the yes/no type questions asked by the program. He need only touch the YES or NO area on the screen instead of typing in the answer. Further man-machine interface improvements using the Plasmascope are planned for MOD3B to make the program, both input and output, as simple and informative as possible.

5.3 JIMPL

A program, written in FORTRAN, to process the Jimsphere wind data was created on the REEDA System. This program produces both scalar wind speed and wind direction plots. An example of each is given in Figure 5-1 and Figure 5-2 respectively. This program requires as input the following data on altitudes ranging from 25 to 20,000 meters.

- Time
- Date
- Altitude
- Wind Direction
- Scalar Wind Speed
Figure 5-1. Example Jimsphere Scalar Wind Speed Plot
Figure 5-2. Example Jimsphere Wind Direction Plot
This program was used to initially process the KSC and Point Mugu Jimsphere wind data to determine temporal variation in the atmospheric kinematics to support climatic diffusion assessments. Over 150 plots were generated. Several modifications were found desirable such as the inclusion of a filter to eliminate bad data or noise, and provide the ability for the user to select only those specific profiles of interest. The ultimate desire was to be able to see a plotted profile without having to actually plot it. This would allow the user to process only data of interest and allows the creation of final Jimsphere profile plots without having to process the data several times.

Consequently, programs to process Jimsphere wind speed/direction (JWSPL, JWDPL and JIMPS) were developed for the REEDA Plasmascope which allows for "touch panel" control. Research or production options are available to allow the user to process all, or portions, of the data with graphic plots on the Plasmascope or hard copy plots. Program JIMPS allows the user to visually display a complete Jimsphere wind speed direction plot on the Plasmascope, thus allowing the user to quickly scan and edit the data before making a hard copy plot. This feature ensures the ability to only generate useful hard copy plots.

The programs JWSPL and JWDPL process the Jimsphere Wind Speed/Direction data respectively. Each allows the user to easily process Jimsphere data quickly through "touch panel" questions and answers, thus eliminating the possibility of erroneous keyboard input. An example scenario from the program JWSPL is given in Figure 5-3. Hopefully, it can be seen from the scenario that by using the Plasmascope "touch panel" control the possibility of making input errors (keyboard) are reduced. A noncomputer oriented user can easily be taught to use such a program within minutes.
**NASA/MSFC Jimsphere Wind Profile Program**

<table>
<thead>
<tr>
<th>Data Being Processed?</th>
<th>Cape Kennedy</th>
<th>Point Mugu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Desired?</td>
<td>Wind Speed</td>
<td>Wind Direction</td>
</tr>
<tr>
<td>Date of Sounding is:</td>
<td>December 29, 1967</td>
<td></td>
</tr>
<tr>
<td>New Date Desired?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time of Sounding is:</td>
<td>1300Z</td>
<td></td>
</tr>
<tr>
<td>Plot Desired?</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Turn on plotter -- Insert paper**

**Touch panel when ready**

**Plotting has been initialized**

<table>
<thead>
<tr>
<th>Time of Sounding is:</th>
<th>1500Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot desired?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Terminate Program?**

| Yes | No |

**Program JWSPL has terminated**

---

Figure 5-3. Example of Operating JWSPL Plasmascope Program

5-6
The overall results of the Plasmascope Jimsphere programs have been quite successful and time saving. Over 700 finalized Wind Speed and Wind Direction profiles have been generated from the KSC and Point Mugu data.

5.4 METPL

METPL is currently a stand alone research program generated to allow visual display of Wind Speed, Wind Direction, Dry Temperature, Potential Temperature, and Cloud Stabilization Height as one profile upon the Plasmascope. This program should be interfaced to MOD3B. As an example of the meteorological profile as it appears on the Plasmascope is shown in Figure 5-4. Various questions will appear at the bottom of the Plasmascope to direct the user as to the moving up or down of the top of the surface mixing layer to the desired height as well as giving the option for a hard copy plot of the generated profile. The meteorological profile is normally obtained from a Rawinsonde of the atmosphere. To obtain the entropy profile required for these soundings, the temperature and pressure are translated into the potential temperature in accordance with the following equation:

$$\theta = T \left( \frac{1000}{p} \right)^{0.288}$$

where the concept of a potential temperature ($\theta$) is introduced to reference the temperature to a specific pressure (1000 mb).

5.5 STAN5

Program STAN5 is a research stand alone program written in FORTRAN and operates on the REEDA System. This program should be interfaced to the MOD3B program. STAN5 calculates the standard deviation of the horizontal wind azimuth angle, SIGAR. Input data are the temperatures, pressures, and altitudes of the first three data levels of KSC Rawinsonde soundings. The levels are the first and second standard
Figure 5-4. Example of METPL Plasmascope Program
altitude levels (16 and 1,000 feet) and the first mandatory pressure level (1,000 mb). The roughness length along the air trajectory with the surface transport layer is also an input variable. The background for the calculation is described in Section 2.

The output of STAN5 includes the time and date of the Rawinsonde sounding, the input data, calculated nondimensional parameters, the gradient of potential temperature, and SIGAR.

5.6 MIXH PROGRAM

Program MIXH is a stand alone research program which operates on the REEDA system. MIXH selects a surface transport layer height based on criteria described in Chapter 2. The input data is a Rawinsonde sounding. MIXH calculates virtual temperature at each level and tests the data according to the prescribed criteria for virtual temperature gradient corresponding to the base of a stable layer and the top of a stable layer. The layers must have a thickness of at least 100 meters. The base of the stable layer nearest to the ground is offered as the height of the surface transport layer. If a stable layer is ground based, then the top of the stable layer is selected as the transport layer height. If no stable layers are found between the surface and 3000 meters, then the transport layer height is set equal to 3000 meters. The output of MIXH is the mixing height of the surface transport layer. If the theory is upheld by extensive test, it should be interfaced to the MOD3B program.
5.7 SKEW T

The existing SKEW T REEDA System program, originally written by Dr. J. B. Stephens of MSFC, was modified to enhance its capabilities in processing sounding data. It can currently accept sounding data from both magnetic tape or disc and in a variety of user specified formats. The SKEW T program generates logarithmic plots for both dew point and temperature as a function of altitude as shown in Figure 5-5.

The SKEW T program was used to process some 23 cases of Battelle Thiokol data for 1974. One additional modification was made to the SKEW T program to allow for processing the Battelle data, which was the calculation of dew point from a given relative humidity and temperature.

5.8 CONCLUSIONS AND RECOMMENDATIONS

These conclusions and recommendations are drawn from the discussion of the interactive REEDA software described in Section 5-1 thru 5-7. It should be evident that a variety of sophisticated interactive REEDA software has been generated and utilized during this contractual period. A vast amount of data from various sources have been processed, analyzed, plotted, etc., by the different REEDA programs. The REEDA software has proven effective, efficient, and invaluable in providing both fast/accurate results in both statistical and graphical form. The current REEDA software, especially the Plasmascope programs provide a means for even a non-computer oriented user to operate and get results with very little effort. The Plasmascope "Touch Panel" capability provides not only for faster user response (i.e., touch-vs-keyboard) but proves superior to the CRT program due to the fact it virtually eliminates or safeguards the user from entering an erroneous value/answer. In addition, due to the 512 by 512 raster dot resolution provided by the Plasmascope virtually
Figure 5-5. Example of SKEW T Plot
unlimited visual graphic display can be generated.

It is recommended that all existing stand alone REEDA software be extensively tested and validated to its fullest extent, with state-of-the-art Plasmascope technology being incorporated whenever and wherever feasible. Additional REEDA software should also be developed, utilizing the REEDA Plasmascope technology, to provide even more capabilities in processing both present and future sources of data.
6. NUMERICAL CLOUD RISE MODEL

Under contract with the Army Missile Command High Energy Laser Programs Office, SAI has recently developed a digital computer program (PUFF) representing a first-order mathematical model for describing the behavior of clouds produced by short-duration high temperature exhausts. In order to more clearly identify and understand the important features associated with the problem of cloud behavior, the cloud history was divided into three phases as depicted in Figure 6-1 and as tabulated in Table 6-1. As indicated in the figure and table, the cloud's history from the time of its initial formation until it reaches equilibrium altitude is contained within Phases I and II. PUFF was primarily designed to handle the problem of cloud behavior during these two phases.

The basic model upon which PUFF is based is the result of a study of relevant literature, both theoretical and experimental. In essence, the cloud is treated as an open thermodynamic system within which all properties are assumed to be uniform. The cloud shape is represented by a sphere and cylinder combination as shown in Figure 6-2. The cloud behavior is predicted by the simultaneous numerical solution of the

- Conservation equations for
  1) Mass
  2) Momentum (3 components)
  3) Energy
- Equation of state
- Volume and center of mass relations for cylinder and sphere combinations.

6-1
Figure 6-1. General Representation of the PUFF Program
Table 6-1. Phases of the PUFF Program

<table>
<thead>
<tr>
<th>Phase (Thermodynamic Phase)</th>
<th>Cloud Behavior</th>
<th>Dominant Effect</th>
<th>Other Effects Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Vortex ring with tail formed near vehicle</td>
<td>Exhaust momentum flux</td>
<td>Buoyancy, Drag, Diffusion</td>
</tr>
<tr>
<td>II</td>
<td>&quot;Tadpole&quot; shaped cloud rises through atmosphere</td>
<td>Buoyancy</td>
<td>Drag, Diffusion</td>
</tr>
<tr>
<td>III (Kinematic Phase)</td>
<td>Cloud reaches equilibrium altitude and spreads out</td>
<td>Diffusion</td>
<td>Drag</td>
</tr>
</tbody>
</table>
I'Exhaust

$L_t$ - length of tail
$r_c$ - radius of sphere
$r_e$ - radius of exhaust
$X_t$ - distance from exhaust to end of tail
$X_{cm}$ - distance from exhaust to center of mass

Figure 6-2. Cloud Shape
The resulting solution yields

- Position \((x,y,z)\)
- Velocity \((U_{c1}, U_{c2}, U_{c3})\)
- Temperature
- Density
- Shape*

6.1 MODIFICATIONS TO PUFF

Some modifications to PUFF are necessary to allow it to be applied to the situation shown in Figure 6-3. Some of the changes are general, relating to both the duct cloud and the ground cloud, while other changes deal with the ground cloud only.

6.1.1 General Changes

General changes to the program include (1) the introduction of atmospheric density and temperature profiles, (2) the calculation of energy released by chemical reaction, (3) the calculation of thermal radiation emitted by the exhaust products and (4) the calculation of the behavior of liquid droplets and solid particulates suspended in the exhaust gases.

The use of atmospheric density and temperature profiles would be based on atmospheric data obtained from soundings. Soundings are taken at regular time intervals before each firing and twice a day normally.

The calculations of the energy released by chemical reaction would involve maintaining an inventory of the chemical species present in the cloud and computing the rate and total amount of each significant reaction associated with the production or consumption of each species. The techniques used in the afterburning analysis and for maintaining an inventory of chemical species have been discussed in Section 3.2.

*In terms of length of cylindrical tail and radius of spherical body.
Figure 6-3. Exhaust Cloud Configurations
The spectral characteristics of the thermal radiation emitted by the exhaust products (gaseous, liquid droplets, and solid particulates) are quite complex and calculation of such characteristics is not a simple task. The total radiation emitted by the various constituents, however, can be calculated by standard engineering techniques and should prove adequate for the type of model under consideration.

The behavior of liquid droplets and/or solid particulates within the cloud depends upon the size of the droplets/particles and the velocity of the flow. Methods for predicting such two-phase flow phenomena have been presented in Subsections 3.1 and 3.3.

6.1.2 Ground Cloud Changes

In addition to the four general changes noted in the preceding subsection, there are certain modifications which relate specifically to the ground cloud alone. PUFF was not originally designed for the case where the rocket exhaust impinges on a solid surface. The program can be easily modified such that in the presence of a solid surface, a surface force is introduced into the momentum equations in such a way that the ground cloud center-of-mass cannot pass through the surface.

Although the ground cloud center-of-mass does not penetrate the surface, allowance must be made for mass, momentum, energy, and species to escape from the ground cloud through the flame trench entrance and ultimately into the duct cloud. All such losses to the ground cloud would be added to the duct cloud to satisfy basic conservation principles. Calculations of the magnitude of the losses would depend upon the height of the rocket engines above the flame trench entrance. As the launch vehicle ascends, the amount of exhaust gases passing through flame trench entrance decreases. This decrease results from the vertical rocket
exhaust plume cross section (at ground level) increasing with time while the velocity within the plume (as ground level) is decreasing.

In the original version of PUFF, the origin of the coordinate system was the center of the jet exit plane. For the case of the ground cloud, this origin would move upward as the launch vehicle ascends. Because it is desirable to have an origin which is stationary with respect to the ground level, PUFF must be modified such that the origin remains fixed at a point corresponding to the center of the rocket exhaust exit plane prior to liftoff. The rocket exhaust exit plane after liftoff will be programmed to move upward in accordance with the known trajectory of the launch vehicle.

Another modification to the program would involve the manner in which the buoyant force is calculated for the ground cloud. Currently in PUFF the buoyant force depends on the difference between the mean cloud density and the atmospheric density at ground level. The atmospheric density surrounding the tail of the ground cloud varies with altitude and thus the buoyant force should involve an integral of the density difference over the altitude interval from ground level to the end (top) of the ground cloud tail.

6.2 CURRENT STATUS OF PUFF PROGRAM

The PUFF Program was converted from the IBM 370 to execute on the REEDA System. Various software incompatibilities had to be resolved, such as:

- Label common not supported
- Multiple entry points not supported
- Initialization of common variables in data statements not supported
- Block data not supported
- Namelist read not supported
These were the initial problems which had to be resolved for a successful compilation. A benchmark run for PUFF on the REEDA System has been established. The necessary logic has been prepared to modify PUFF to account for:

- a variable atmosphere
- a moving exhaust nozzle
- a solid ground level

A mathematical model for calculating the jet stagnation length has been established. The necessary data for calculating the radiative emittance of the exhaust products have been collected. Once this was accomplished a benchmark comparison was made; however, some discrepancies were noted. When the REEDA version of PUFF was compared to an IBM 370 operational version identical results were obtained from the initial time until time was equal to 0.10 seconds using 0.01 sec time increments. At this point in the execution of the program, the time increment was increased to 0.10 second. Using the new time increment, significant differences begin to appear in the calculated results. Various modifications were made to try to eliminate the difference. All variables and calculations were changed to double precision. Various complex arithmetic computations were re-structured into less compound statements to eliminate possible loss of accuracy by truncation, etc. The above changes have not affected the final results. There still remained differences in the results when the time increment was increased to 0.10 second. Thus it was decided not to increase the time increment, but to leave it constant at .01 second for the entire duration of the program to determine if better accuracy is gained at larger times into the run; however, the results did not change. Consequently, analysis of the REEDA version of the PUFF Program will continue with appropriate modifications being made.
It should be noted, however, that due to the difference in hardware (i.e., IBM 370 vs HP 21MX REEDA), 32 bit vs 16 bit single precision words), 64 bit double precision vs 48 bit double precision words), (16 significant digits vs 11 significant digits), complete agreement between the results of the two machines may not be obtainable given the algorithms that exist currently in the PUFF Program.

6.3 CONCLUSIONS

A new numerical cloud rise program developed for another purpose has been investigated to see if it is suitable for use on the exhaust cloud from the Space Shuttle propulsion system. Development has been initiated to modify the original code and convert it for use on the REEDA system. The use of this code would allow the simultaneous determination of the cloud shape and size and the radiation loss from the exhaust effluents. No other known analyses can handle the situation as aptly as the PUFF code.
7. OVERALL CONCLUSIONS AND RECOMMENDATIONS

The study performed under NASA Contract NAS8-31806 has yielded large dividends in the technology learned, the basic algorithms developed, the meteorological knowledge about KSC brought to a useful form, and the large amount of software developed. New techniques utilizing the touch panel on the Plasmascope have yielded programs that are convenient and rapid to use. The effort has basically completed the necessary homework for a full scale climatological diffusion assessment; what is now required is to bring together the various models and start the development of an operational diffusion model useful not only for a climatological assessment but for monitoring operational launches. The technique for the calculation of the SRB exhaust effluents has been developed but so far losses due to plume impingement, radiation, the flame trench, and water injection have not been considered. A preliminary climatological diffusion assessment was performed to validate the techniques developed; the results have only limited validity and no conclusions can be drawn from the results of the study. The study assumed the Space Shuttle was a Titan type vehicle with only solid propellant boosters; the liquid propellant SSME and their interactions with the solid motor effluents were not considered.

7.1 RECOMMENDED STUDY

Ground-based stable layers and inversions are common over land areas near KSC during calm clear nights, especially in winter. The percent frequency of occurrence of ground based inversions for various thickness intervals by season at KSC (50) is given in Table 7-1.
Table 7-1. Percent frequency of occurrence of ground-based inversions by season at KSC during 1965 - 1969 at 0700 and 1900 EST.

<table>
<thead>
<tr>
<th>Thickness of Ground Based Inversion (m)</th>
<th>Dec</th>
<th>Mar</th>
<th>June</th>
<th>Sept</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;100</td>
<td>2.1</td>
<td>1.2</td>
<td>1.5</td>
<td>2.1</td>
</tr>
<tr>
<td>101 - 250</td>
<td>32.4</td>
<td>25.3</td>
<td>30.2</td>
<td>23.5</td>
</tr>
<tr>
<td>251 - 500</td>
<td>23.5</td>
<td>22.8</td>
<td>27.6</td>
<td>26.0</td>
</tr>
<tr>
<td>501 - 750</td>
<td>2.5</td>
<td>4.1</td>
<td>1.3</td>
<td>0.6</td>
</tr>
<tr>
<td>751 - 1000</td>
<td>1.6</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>1000 - 1500</td>
<td>3.5</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>&gt;1500</td>
<td>0.7</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
</tr>
</tbody>
</table>
The statistics in Table 7-1 are based on Rawinsonde data. Shallow ground based inversions (thickness less than 250 m) reported at KSC are based on a surface temperature (at 16 ft) and a temperature at the first mandatory pressure level (1,000 mb). Since the temperature at 16 ft is strongly influenced by local micrometeorological conditions the statistics of shallow inversions are not representative of other locations beyond a short distance from the measurement location. However if inversions are reported based on temperature observations at three or more altitudes (including the observation at 16 ft) there is more support for the argument that stable conditions exist near the ground over a wider area in the vicinity of the measurement site. Since a ground based stable layer at a particular location will effectively insulate that location from the stabilized SRB cloud, it is important to establish the applicability of the available KSC inversion statistics to the climatological impact analysis. The physical processes responsible for the formation of ground-based inversions in the areas surrounding KSC are influenced by the relative distribution of rural and urban topography and water bodies. Urban areas and water bodies during winters at KSC represent nocturnal heat sources which could contribute to the maintenance of a nocturnal mixed layer. Although a nocturnal mixed layer has been identified over large cities (51, 52) its existence has not been identified or correlated with nocturnal heat sources in the vicinity of KSC.

It is a reasonable hypothesis that the inversion statistics obtained from KSC Rawinsonde data are not necessarily representative of conditions at all locations of interest in the vicinity of KSC. It is suggested than an experimental study be implemented to establish the relative strength and frequency of occurrence of ground based stable layers and inversions over various locations of interest near KSC. Adequate results would be obtained by sampling...
temperatures aloft (to 1 km) daily, 1 hour before sunrise, during January and February over population centers (Titusville, Cocoa), working areas, and viewing areas within KSC boundaries. The purpose of the study will be the establishment of the degree of conservatism of air quality impact calculations based on the available large sample of Rawinsonde data at KSC.
8. REFERENCES CITED


REFERENCES CITED
(Continued)


REFERENCES CITED

(Continued)


REFERENCES CITED

(Continued)


REFERENCES CITED
(Continued)


REFERENCES CITED

(Concluded)


APPENDIX A

SOFTWARE SOURCE LISTINGS

This section contains the complete source listings of most of the software programs discussed in this report.
A-1.  Conversion Programs (Generic)

- IBM 370/360  BCD/EBCDIC → ASCII
- UNIVAC 1108  BCD/EBCDIC → ASCII
- HP 2100      EBCDIC → ASCII
IBM 370/360 BCD/EBCDIC + ASCII Conversion
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>SYMT</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PRINT NOGEN</td>
</tr>
<tr>
<td>000038 1821</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 HWPTAPE READY</td>
</tr>
<tr>
<td>00003A D503 204C C358 0004C 00358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12 RDJFCB (TAPE)</td>
</tr>
<tr>
<td>000040 4780 C05A 0005A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18 OPEN (CARDIN, TAPE, (OUTPUT))</td>
</tr>
<tr>
<td>00044 DC4F 2000 C198 00030 00198</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26 READCARD DS OH</td>
</tr>
<tr>
<td>000056 47FO C02E 0002E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>27 GET CARDIN</td>
</tr>
<tr>
<td>00005A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31 LR 2,1</td>
</tr>
<tr>
<td>000064 4820 C128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32 CLC 76(4+2)=C0 U</td>
</tr>
<tr>
<td>00006A 4120 2001 00001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33 BE CLOSE</td>
</tr>
<tr>
<td>00006E 4020 C128 00128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>34 TR 0180,21,0C0TAASC</td>
</tr>
<tr>
<td>000074 47FO C02E 0002E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35 PUT TAPE,(2)</td>
</tr>
<tr>
<td>000082</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40 B READCARD</td>
</tr>
<tr>
<td>000084</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>41 CLOSE DS OH</td>
</tr>
<tr>
<td>000086 4820 C128 00128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42 CLOSE (TAPE, LEAVE)</td>
</tr>
<tr>
<td>000094 4120 2001 00001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48 LH 2,JFCDAREA+60</td>
</tr>
<tr>
<td>00009E 4020 C128 00128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49 LA 2,1,20,2</td>
</tr>
<tr>
<td>0000A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50 STH 2,JFCDAREA+60</td>
</tr>
<tr>
<td>0000A4 47FO C02E 0002E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51 OPEN (TAPE, (OUTPUT, LEAVE), TYPE=J</td>
</tr>
<tr>
<td>0000A6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57 B RFAADCARD</td>
</tr>
<tr>
<td>0000A8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58 EDF DS OH</td>
</tr>
<tr>
<td>0000BA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59 CLOSE (CARDIN, TAPE)</td>
</tr>
<tr>
<td>0000BC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67 RET</td>
</tr>
<tr>
<td>LOC</td>
<td>OBJECT CODE</td>
<td>ADDR1</td>
<td>ADDR2</td>
<td>STAT</td>
<td>SOURCE STATEMENT</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>00004</td>
<td>75</td>
<td>JFCBAREA</td>
<td>DS</td>
<td>44F</td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td>76</td>
<td>LIST</td>
<td>DC</td>
<td>X'87', AL3(JFCBAREA)</td>
<td></td>
</tr>
<tr>
<td>00018</td>
<td>8001020304050607</td>
<td>77</td>
<td>BCTOASC</td>
<td>DC</td>
<td>X'000102030405060708090A083C9300E0F'</td>
</tr>
<tr>
<td>00019</td>
<td>78</td>
<td>OC</td>
<td>X'101112131415161718191A1B1C1D1E1F'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00020</td>
<td>79</td>
<td>DC</td>
<td>X'201222232425262728292A2B2C2D2E2F'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00021</td>
<td>80</td>
<td>OC</td>
<td>X'303132333435363738393A3B3C3D3E3F'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00022</td>
<td>81</td>
<td>DC</td>
<td>X'40'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00023</td>
<td>82</td>
<td>DC</td>
<td>X'50'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00024</td>
<td>83</td>
<td>DC</td>
<td>X'60'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00025</td>
<td>84</td>
<td>DC</td>
<td>X'70'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00026</td>
<td>85</td>
<td>DC</td>
<td>X'80'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00027</td>
<td>86</td>
<td>DC</td>
<td>X'90'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00028</td>
<td>87</td>
<td>DC</td>
<td>X'A0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00029</td>
<td>88</td>
<td>DC</td>
<td>X'B0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002A</td>
<td>89</td>
<td>DC</td>
<td>X'C0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002B</td>
<td>90</td>
<td>DC</td>
<td>X'D0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002C</td>
<td>91</td>
<td>DC</td>
<td>X'E0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002D</td>
<td>92</td>
<td>DC</td>
<td>X'F0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002E</td>
<td>93</td>
<td>DC</td>
<td>X'100'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002F</td>
<td>94</td>
<td>DC</td>
<td>X'200'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00030</td>
<td>95</td>
<td>DC</td>
<td>X'300'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00031</td>
<td>96</td>
<td>DC</td>
<td>X'400'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00032</td>
<td>97</td>
<td>DC</td>
<td>X'500'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00033</td>
<td>98</td>
<td>DC</td>
<td>X'600'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00034</td>
<td>99</td>
<td>DC</td>
<td>X'700'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00035</td>
<td>100</td>
<td>DC</td>
<td>X'800'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00036</td>
<td>101</td>
<td>DC</td>
<td>X'900'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00037</td>
<td>102</td>
<td>DC</td>
<td>X'A00'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00038</td>
<td>103</td>
<td>DC</td>
<td>X'B00'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00039</td>
<td>104</td>
<td>DC</td>
<td>X'C00'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

106 CARDIN DCB DSORG=PS, MACRF=GL, DONAME=CARDS, E3DAD=EOF, RECFM=FB, LRECL=80
160 TAPE DCB DSORG=PS, MACRF=PM, DONAME=TAPE, EXIST=LIST, RECFM=FB, LRECL=80

00000 214 END HWPITAPE
00035 215 =C* U*
<table>
<thead>
<tr>
<th>Line</th>
<th>Instructions</th>
<th>Address</th>
<th>Memory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A-7
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Instruction</th>
<th>Label</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>UUD133</td>
<td>73 12 16 00 0 UUGU20</td>
<td>LSSL</td>
<td>AI4:29</td>
</tr>
<tr>
<td>110</td>
<td>UUD134</td>
<td>40 00 02 00 0 UUGU32</td>
<td>OR</td>
<td>AI4:19</td>
</tr>
<tr>
<td>111</td>
<td>UUD135</td>
<td>19 12 01 01 0 UUGU50</td>
<td>LA,54</td>
<td>AI4:0,X1</td>
</tr>
<tr>
<td>112</td>
<td>UUD136</td>
<td>10 00 16 16 0 UUGU59</td>
<td>LA</td>
<td>AI4:FD2ASC+A1</td>
</tr>
<tr>
<td>113</td>
<td>UUD137</td>
<td>73 12 16 00 0 UUGU20</td>
<td>LSSL</td>
<td>AI4:16</td>
</tr>
<tr>
<td>114</td>
<td>UUD138</td>
<td>40 00 03 00 0 UUGU32</td>
<td>OR</td>
<td>AI4:14</td>
</tr>
<tr>
<td>115</td>
<td>UUD141</td>
<td>10 11 01 01 0 UUGU00</td>
<td>LA,55</td>
<td>AI4:0,X1</td>
</tr>
<tr>
<td>116</td>
<td>UUD142</td>
<td>10 00 16 16 0 UUGU54</td>
<td>LA</td>
<td>AI4:FD2ASC+A1</td>
</tr>
<tr>
<td>117</td>
<td>UUD143</td>
<td>73 12 16 00 0 UUGU10</td>
<td>LSSL</td>
<td>AI4:18</td>
</tr>
<tr>
<td>118</td>
<td>UUD144</td>
<td>40 00 09 00 0 UUGU32</td>
<td>OR</td>
<td>AI4:19</td>
</tr>
<tr>
<td>119</td>
<td>UUD145</td>
<td>10 10 01 01 0 UUGU00</td>
<td>LA,50</td>
<td>AI4:0,X1</td>
</tr>
<tr>
<td>120</td>
<td>UUD146</td>
<td>10 00 16 16 0 UUGU54</td>
<td>LA</td>
<td>AI4:FD2ASC+A1</td>
</tr>
<tr>
<td>121</td>
<td>UUD147</td>
<td>40 00 05 00 0 UUGU32</td>
<td>OR</td>
<td>AI4:19</td>
</tr>
<tr>
<td>122</td>
<td>UUD150</td>
<td>01 00 06 02 0 UUGU00</td>
<td>SA</td>
<td>AI4:0,X2</td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td>EVEN WORD</td>
</tr>
<tr>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>UUD151</td>
<td>06 02 01 00 0 UUGU12</td>
<td>SX,H1</td>
<td>AI4:TEST1D</td>
</tr>
<tr>
<td>126</td>
<td>UUD152</td>
<td>52 42 02 00 0 UUGU12</td>
<td>TE,H1</td>
<td>AI4:TEST1D</td>
</tr>
<tr>
<td>127</td>
<td>UUD153</td>
<td>74 54 00 00 0 UUGU21</td>
<td>J</td>
<td>LOOP</td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>UUD154</td>
<td>10 16 00 00 0 UUGU3</td>
<td></td>
<td>WTLINE</td>
</tr>
<tr>
<td>130</td>
<td>UUD155</td>
<td>01 11 00 00 0 UUGU54</td>
<td>LA,U</td>
<td>AI4:LINE</td>
</tr>
<tr>
<td>131</td>
<td>UUD156</td>
<td>10 16 00 00 0 UUGU4</td>
<td>LA,H2</td>
<td>AI4:RND4</td>
</tr>
<tr>
<td>132</td>
<td>U</td>
<td>72 11 00 00 0 UUGU00</td>
<td>ER</td>
<td>LNS8</td>
</tr>
<tr>
<td>133</td>
<td>U</td>
<td>74 41 00 00 0 UUGU00</td>
<td>J</td>
<td>LU CARD</td>
</tr>
<tr>
<td>134</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>UUD161</td>
<td>10 16 00 00 0 UUGU11</td>
<td>LA,U</td>
<td>AI4:11</td>
</tr>
<tr>
<td>136</td>
<td>UUD162</td>
<td>01 11 00 00 0 UUGU3</td>
<td>SA,52</td>
<td>AI4:TAPK</td>
</tr>
<tr>
<td>137</td>
<td>UUD163</td>
<td>10 16 00 00 0 UUGU4</td>
<td>LA,U</td>
<td>AI4:RND4</td>
</tr>
<tr>
<td>138</td>
<td>U</td>
<td>72 11 00 00 0 UUGU00</td>
<td>ER</td>
<td>LNS8</td>
</tr>
<tr>
<td>139</td>
<td>U</td>
<td>74 41 00 00 0 UUGU00</td>
<td>J</td>
<td>LU CARD</td>
</tr>
<tr>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>142</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>143</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>147</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>148</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>154</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A-9
ASMB,R,L,F

0001   THIS SUBROUTINE CONVERTS REAL NUMBERS IN IBM FORMAT TO HP FORMAT
0002   THE CALLING SEQUENCE IS: CALL IBHNP(IBML,IBM2,HP)
0003   WHERE: IBML - THE MOST SIGNIFICANT PART OF THE IBM REAL WORD
0004   IBM2 - THE LEAST SIGNIFICANT PART OF THE IBM REAL WORD
0005   HP - THE REAL WORD IN WHICH THE RESULT IS TO BE STORED
0006   IN HP FORMAT
0007   XAN IBMHP
0008   ENT IBMHP
0009   EXT .EHTR
0010   IBM1 BSS 1
0011   IBM2 BSS 1
0012   HP1 BSS 1
0013   IBHNP, NOP
0014   ENTRY/EXIT POINT
0015   JSB .EHTR
0016   GET ADDRESSES OF PARAMETERS INTO
0017   DEF IBM1
0018   IBML, IBM2, AND HP1
0019   LDA HP1
0020   SET UP THE ADDRESS OF THE SECOND
0021   ISZ A
0022   HALF OF THE HP REAL WORD
0023   STA HP2
0024   LDA IBM1, I
0025   GET FIRST PART OF IBM REAL WORD
0026   AND =8B77400
0027   MASK OFF THE EXPONENT
0028   ORL ALF
0029   SHIFT TO LOWER 8 BITS OF A
0030   AND AIB1 AS
0031   REMOVE IBM EXPONENT BIAS OF 64
0032   ADA A
0033   QUADRUPLE EXPONENT TO MAKE IT
0034   ADA A
0035   A POWER OF 2
0036   JMP NEGX
0037   JUMP IF NEGATIVE EXPONENT
0038   JMP STEX
0039   1 BIT RIGHT AND STORE
0040   NEGX
0041   JUMP IF NEGATIVE EXPONENT -- SHIFT IT 1
0042   AND =B0000376
0043   MASK OFF JUST THE EXPONENT
0044   SZA
0045   (SKIP IF EXACTLY ZERO)
0046   OR =B000001
0047   PUT IN SIGN BIT
0048   STEX STA TEMP
0049   STORE THE EXPONENT TEMPORARILY
0050   STA IBM1, I
0051   GET FIRST PART OF IBM REAL WORD
0052   RAL
0053   ROTATE IT LEFT 1 BIT
0054   AND =B000001
0055   MASK OFF MANTISSA SIGN BIT
0056   STA B
0057   STORE IN B REGISTER
0058   STA IBM2, I
0059   GET SECOND PART OF IBM REAL WORD
0060   ARS
0061   DROP LEAST SIGNIFICANT BIT OF MANTISSA
0062   AND =B077777
0063   AND CLEAR THE UPPER BIT
0064   SZB,RS5
0065   SKIP IF MANTISSA IS NEGATIVE
0066   JMP #2
0067   DON'T COMPLEMENT IF MANTISSA POSITIVE
0068   CMN,INA
0069   MANTISSA NEGATIVE, COMPLEMENT
0070   CMP A
0071   MANTISSA NEGATIVE, COMPLEMENT
0072   AND =B000377
0073   GET LOWEST EIGHT BITS
0074   ALF,ALF
0075   PUT IN UPPER PART OF WORD
0076   IOR TEMP
0077   OR IN THE EXPONENT
0078   STA HP2, I
0079   PUT IN SECOND HALF OF HP REAL WORD
LDA IB2.I
AND #$177400
ALF,ALF
STA TEMP
LDA IB1.I
AND #$0000377
ALF,ALF
IOR TEMP
ARDS
AND #$077777
SZS,RSS
JMP ++2
CMA
STA HP1.I
JMP IBMHPI.I
A EQU 0
B EQU 1
IBIAS DEC -64
TEMP, HP
HP2 BSS 1
END
SUBROUTINE E2A(Ia,LIA)
INTEGER EBCASC(256)
DIMENSION IA(1)
DATA EBCASC/0,1,2,3,0,9,0,127,0,0,0,11,12,13,14,15,16,17,18,0,
          0,0,8,0,24,25,9,0,0,0,0,0,28,0,0,0,23,27,
          0,0,0,0,5,6,7,0,0,22,0,0,30,0,4,0,0,0,0,
          20,21,0,26,32,0,0,0,0,0,0,0,0,0,0,0,46,60,40,43,0,
          38,0,0,0,0,0,0,0,0,0,33,36,42,41,59,94,45,47,0,0,
          0,0,0,0,0,0,124,44,37,95,62,63,0,0,0,0,0,0,0,0,
          0,96,58,35,64,39,61,34,0,97,
          98,99,100,101,102,103,104,105,0,0,
          0,0,0,0,0,106,107,108,109,110,
          111,112,113,114,0,0,0,0,0,0,
          0,126,115,116,117,118,119,120,121,122,
          0,0,0,0,91,0,0,0,0,0,0,0,
          0,0,0,0,0,0,0,0,0,93,
          0,0,123,65,66,67,68,69,70,71,
          72,73,0,0,0,0,0,125,74,
          75,76,77,78,79,80,81,82,0,0,
          0,0,0,0,92,0,83,84,85,86,
          87,88,89,90,0,0,0,0,0,0,
          48,49,50,51,52,53,54,55,56,57,
          0,0,0,0,0,0,0/ 
DO 7 I=1,LIA
INDEX1 = IAND(ISHIF(IA(I),-9),177B) + 1
INDEX2 = IAND(IA(I),177B) + 1
7 IA(I) = IOR(ISHIF(EBCASC(INDEX1),8),EBCASC(INDEX2))
RETURN
END
END$
A-2. Conversion Programs (Specific)

- 1965 KSC Rawinsonde Data Conversion Program
- 1974 Vandenberg Rawinsonde Data Conversion Program
- 1964-1970 Jimsphere Data Conversion Programs
1965 KSC Rawinsonde Data Conversion Program
FTN4, L

PROGRAM SOUND
DIMENSION IBUF(40), OBUF(40), SIZE(2)
DIMENSION NAME(3), IDCB(256)
INTEGER OBUF
DATA I99/2H/ 9/
DATA ISIZE/-1,40/
DATA IST/2HST/
DATA NAME/2H&5,2HDB,2H65/
C** CREATE DISC FILE TO STORE CONVERTED SOUNDING
CALL CREAT(IDCB, IERR, NAME, ISIZE, 3)
C
SET IL TO NUMBER OF WORDS TO BE WRITTEN = 39
IL=39
NC = 1
WRITE(6,320)
READ(8,15) OBUF
C
IF(OBUF(2).NE.IST) GO TO 56
C
NC=NC +1
C
IF(NC.LE.12) GO TO 56
WRITE(6,310) (OBUF(N), N=1,40)
CALL CODE
WRITE (IBUF,15) (OBUF(N), N=1,40)
CALL WRITF (IDCB, IERR, IBUF, IL)
88 DO 10 I=1,5
NC=NC+1
READ(8,15) OBUF
15 FORMAT(40A2)
WRITE(6,310) (OBUF(N), N=1,40)
CALL CODE
WRITE (IBUF,15) (OBUF(N), N=1,40)
CALL WRITF (IDCB, IERR, IBUF, IL)
10 CONTINUE
16 READ(8,20) ID, TD, TKTS, TIT, ITS, IDT, IDS, IPRESS, IR, IA
114, IDT, IVS, ID, IWS
20 FORMAT(16.3X,13,2X,13,3X,12,4X,12,2X,13,3X,12,14,1X
115, 3X,13,2X,13,5X)
CALL ISIGC(ITEMP, ITS, IDPT, IDS, TPT)
PRESS = IPRESS/10.
AB = IAB/100.
DEN = IDEN/10.
WRITE(6,350) IDALT, IWD, IWKT, TEMP, TPT, PRESS, IR, AB, DEN, IDT, IVS, IWS
350 FORMAT(1X,16,3X,13,5X,13,2X,F5.1,3X,F5.1,3X,F6.1,3X,F6.1,3X,F12,3X
1F5.2,2X,F6.1,1X,I3,13,1X,I3,13,1X)
CALL CODE
WRITE(IBUF,350) IDALT, IWD, IWKT, TEMP, TPT, PRESS, IR, AB, DEN,
1 IR, IWS
CALL WRITF (IDCB, IERR, IBUF, IL)
IF(IDALT.LE.19500) GO TO 16
17 READ(8,15) OBUF
IF(OBUF(4).NE.199) GO TO 17
WRITE(6,310) (OBUF(N),N=1,40)
CALL CODE
WRITE(6,310) (OBUF(N),N=1,40)
CALL WRITEF (IDCB,IERR,IBUF,IL)
WRITE(6,320)
DO 18 I=1,2
READ(8,15) OBUF
WRITE(6,310) (OBUF(N),N=1,40)
CALL CODE
WRITE(6,310) (OBUF(N),N=1,40)
CALL WRITEF (IDCB,IERR,IBUF,IL)
CONTINUE
18 READ(8,19) IALT, IWD, IWKTS,ITEMP, ITS, IDPT, IDS, IPRESS, IRH
FORMAT(I6,3X,I3,2X,I3,3X,I2,A2,4X,I2,A2,3X,I5,3X,I2)
PRESS = IPRESS/10.
CALL ISIGC(ITEMP,ITS,IDPT,IDS,TEMP,DPT)
WRITE(6,351) IALT, IWD, IWKTS,TEMP,DPT,PRESS,IRH
351 FORMAT(IX,16,3X,I3,5X,I3,2X,F5.1,3X,F5.1,3X,F6.1,3X,I2)
CALL CODE
WRITE(6,351) IALT, IWD, IWKTS,TEMP,DPT,PRESS,IRH
CALL WRITEF (IDCB,IERR,IBUF,IL)
IF(IALT.LE.19500) GO TO 21
READ(8,15) OBUF
IF(OBUF(2).NE.IST) GO TO 86
WRITE(6,320)
WRITE(6,310) (OBUF(N),N=1,40)
CALL CODE
WRITE(6,310) (OBUF(N),N=1,40)
CALL WRITEF (IDCB,IERR,IBUF,IL)
IF(NC.GT.700) GO TO 90
GO TO 88
86 FORMAT(IH1)
310 FORMAT(1X,40A2)
90 CALL CLOSE(IDCB,IERR)
END
SUBROUTINE ISIGC(ITEMP,ITS,IDPT,IDS,TEMP,DPT)
DIMENSION ICHR(10), INM(10)
DATA ICHR/ZHO,2H1,2H2,2H3,2H4,2H5,2H6,2H7,2H8,2H9/
DATA INM/2H!,2HJ,2HK,2HL,2HM,2HN,2HO,2HP,2HQ,2HR/
DO 10 I=1,10
A = I - 1
B = -1.
IF(ITS.EQ.ICHR(I)) TEMP =(FLOAT(ITEMP) + A/10.) * B
IF(IDS.EQ.ICHR(I)) DPT = (FLOAT(IDPT) + A/10.) * B
IF(ITS.EQ.INM(I)) TEMP = (FLOAT(ITEMP) + A/10.)
IF(IDS.EQ.INM(I)) DPT = (FLOAT(IDPT) + A/10.)
10 CONTINUE
RETURN
END
END$
1974 Vandenberg Rawinsonde Data Conversion Programs

- IBM 360/44 Variable Length → Fixed Length IBM 370
- Data Selection Program
- IBM 370 EBCDIC → ASCII (See pages A-3 through A-5)
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>J00014 47F0 C09E</td>
<td>0009E</td>
<td>00000</td>
<td>0044F</td>
<td>PRINT</td>
<td>NOGEN</td>
<td></td>
</tr>
<tr>
<td>J0001C 5820 1000</td>
<td>00004</td>
<td>0044F</td>
<td>00001</td>
<td>BRANCH</td>
<td>B OPEN</td>
<td></td>
</tr>
<tr>
<td>J0002E 5010 3000</td>
<td>00400</td>
<td>00000</td>
<td>0044F</td>
<td>L 2,011,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J0002F 1011</td>
<td>00400</td>
<td>00000</td>
<td>0044F</td>
<td>L 3,411,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00032</td>
<td>00000</td>
<td>0044F</td>
<td>00000</td>
<td>GET TAPE, (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00034</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>SR 1,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00035</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>ST 1,01,3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00036</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>RETURN DS OH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00037</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>RET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00044</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>DS OH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00046 96F0 C015</td>
<td>0044F</td>
<td>00000</td>
<td>0044F</td>
<td>DS OH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J00047</td>
<td>00001</td>
<td>0044F</td>
<td>00000</td>
<td>BRANCH+1, X'FO'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J0009A 47F0 C032</td>
<td>00400</td>
<td>00000</td>
<td>0044F</td>
<td>CEB</td>
<td>DSORG=PS,MACF=GM,UDNAME=SASTAPE,EODAD=EOF</td>
<td></td>
</tr>
</tbody>
</table>

A-21
COMPILER OPTIONS - NAME = MAIN,OPT=00,LINECNT=54,SIZE=0200K,
SOURCE=EBCDIC,NOLIST,NOECK,LOAD=MAP,NOEDIT,ID,NOXREF

ISN 0002  DIMENSION IMN(12)
ISN 0003  LOGICAL*1 IDATA(20), (BUF(2500)
ISN 0004  DIMENSION PRESS(78), PESSX(78), ALT(78), ALT(78), TEM(78)
ISN 0005  INTEGER*2 YR, M0, DA, HR, 08, 19D, PRESS, TEMP, OPT, WD, WS, NLEVEL
ISN 0006  INTEGER HT, STN, EOF
ISN 0007  EQUIVALENCE (BUF(5), NLEVEL), (BUF(9), STN), (BUF(13), YR),
ISN 0008  IF (BUF(10), YR), (BUF(17), DA), (BUF(19), HR),
ISN 0009  IF (BUF(25), O0), (DATA(1), IND), (DATA(3), PRESS),
ISN 0010  IF (DATA(5), HT), (DATA(9), TEMP), (DATA(11), OPT),
ISN 0011  IF (DATA(13), WD), (DATA(15), WS)
ISN 0012  DATA INN/4HJAN, 4HJFEB, 4HMAY, 4HJHAPR, 4HMAE, 4HJU, 4HJJuly, 4HJAug,
ISN 0013  14HSEP, 4HOLT, 4HNOV, 4HDEC, 2
C ** CALL TAPE READ ROUTINE
ISN 0014  ICNT = 0
ISN 0015  10 CALL MPTR(I, BUF, EOF)
ISN 0016  IF (EOF.EQ.1) GO TO 99
C ** CHECK FOR PLURAL LEVEL
C ** CHECK FOR RAWHINSUNDE DATA
ISN 0017  IF (DA.LT.1, OR, DA.GT.2) GO TO 10
C ** CHECK FOR SURFACE LEVEL
ISN 0018  DO 15 J = 1, 20
ISN 0019  15 N = I + 80
ISN 0020  IF (DATA(I) = BUF(N))
ISN 0021  CONTINUE
ISN 0022  IF (DATA(J) = BUF(J)) GO TO 10
C ** COMPUTE NUMBER OF DATA LEVELS
ISN 0023  ILEVEL = NLEVEL
ISN 0024  LEVELS = ILEVEL/20
ISN 0025  IF (BUF.EQ.1) LEVELS = LEVELS
ISN 0026  IF (BUF.EQ.2) LEVELS = LEVELS
C ** PROCESS ALL LEVELS OF DATA
ISN 0027  DO 20 I = 1, LEVELS
ISN 0028  II = I*20 + 61
ISN 0029  JJ = II + 19
ISN 0030  K = 0
ISN 0031  IF (K = 0)
ISN 0032  DO 25 J = II, JJ
ISN 0033  K = J + 1
ISN 0034  25 CONTINUE
ISN 0035  IF (III.EQ.1) GO TO 99
ISN 0036  PRESS(I) = PRESS
ISN 0037  PRESS(I) = PRESS(I)/10.
ISN 0038  ALT(I) = HT
ISN 0039  TEMP(I) = TEMP
ISN 0040  TEMP(I) = (TEMP(I)/10.) - 273.16
ISN 0041  WSP(I) = WD
ISN 0042  WSP(I) = WS/10.
ISN 0043  DPT(I) = DPT
**CONTINUE**

**PRESSR(1) = PRESS**

**PRESSR(1I) = PRESSR(1)/10.**

**ALTR(1) = HT**

**TEMPR(1) = TEMP**

**TEMP(1) = (TEMPR(1)/10.) - 273.16**

**WD(1) = WD**

**WS(1I) = WS/10.**

**OPT(1) = OPT**

**DA = DA**

**HR = HR**

**CONTINUE**

**SIN = SIN**

**YR = YR**

**MU = MO**

**DA = DA**

**HR = HR**

**WRITE DATA RECORDS TO TAPE**

*IF(UB.EQ.1I) GO TO 10*

*ICNT = ICNT + 1*

*IF(IDAP.NE.10.AND.HRP.NE.IHR). GO TO 10*

*IF(HR.PE.12). GO TO 10*

*IF(INC.IE.11). GO TO 10*

**CONTINUE**

**P**

**COMPUTE PIBAL Pressures**

**IF(PRESS(1).EQ.-1.AND.ALTP(I.EQ.ALTR(I)) PRESS(1) = PRESSR(1)**

**N = 1**

**DU 210 I = 2, LEVELP**

**IF(PRESS(1).NE.-1). GO TO 220**

**N = N + 1**

**GO TO 210**

**GO TO 210**

**IF(N.EQ.1I) GO TO 210**

**J = J - 1**

**A = ALTP(I) - ALTP(J)**

**Y = LOG(PRESSP(J))**

**Z = LOG(PRESSP(I))**

**C = Y - Z**

**NN = N - 1**

**DO 32 K = 1,NN**

**A = ALTP(J+K) - ALTP(I)**

**B = Y/X**

**D = 3*C**

**E = Y - D**

**PRESSP(J+K) = EXP(E)**

**CONTINUE**

**N = 1**

**CONTINUE**

**CONTINUE**

**COMPUTE RAOB ALTITUDES**
A-24
1964 - 1970 Jimsphere Conversion Programs

- UNIVAC 1108 Data Reduction
- UNIVAC 1108 BCD → ASCII (See pages A-6 through A-11)
*DIAGNOSTICS* - THE NAME A APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.

```
C0100 10  DIMENSION IDATA(30),WD14(19)  000000
C0103 20  DIMENSION A(18)  000001
C0104 30  READ(5,1000) NFILES  000002
C0107 40  DO 50 NFILE=1,NFILES  000003
C0112 50  WRITE(6,105) NF  000004
C0115 60  DO 40 NR=1,20  000005
C0120 70  CALL NTRAN(B*2,29*1,DATA,IERR*22)  000006
C0121 80  IF(IERR.EQ.-1) GO TO 10  000007
C0123 90  IF(IERR.LT.-1) CALL NTRAN(B*22)  000008
C0125 100  MDX = 16  000009
C0129 110  DO 30 IK=1,5  000010
C0131 120  ENCODER(BC,LLK,WD14,INUM) (DATA(MDX+7*K),K=1,8)  000011
C0137 130  MDX=MDX+56  000012
C0140 140  CALL NTRAN(9*1,14,WD14,IERR*22)  000013
C0141 150  IF(INF.GE.2) GO TO 25  000014
C0143 160  WRITE(6,164) (WD14(K),K=1,19)  000015
C0146 170  1040 FORMAT(1H,14A6)  000016
C0147 180  25 CONTINUE  000017
C0150 190  36 CONTINUE  000018
C0152 200  40 CONTINUE  000019
C0154 210  CALL NTRAN(B*7,1,22)  000020
C0155 220  56 CONTINUE  000021
C0157 230  CALL NTRAN(9*1)  000022
C0160 240  CALL NTRAN(9*11)  000023
C0161 250  STOP  000024
C0163 260  1000 FORMAT(1S)  000025
C0164 270  1630 FORMAT(8F10.1)  000026
C0164 280  1050 FORMAT(1H,10HF*,16)  000027
C0165 290  END  000028
```

END OF COMPILATION:  I DIAGNOSTICS.
Interactive REEDA Programs

- MOD3A
- MOD3B
- METPL
- STAN5
- MIXH
- JWSPL
- JWDPL
- JIMPS
- SKEW T (Version I & II)
- PUFF

*These have been merged into the NASA/MSFC REED Diffusion Model Program Version I.
Program MOD3A
PROGRAM MOD3A

C***********************************************************************
C
HASL/llSFC HULTILAYER DIFFUSION MODEL - MOD3A 04 MAY 1977
C***********************************************************************
C
COMMON BLOCK
C
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
ISMONC2), ISYEAR, IV2, JTOP, LAUHTD(10), LTIME, LTIM, LDAY,
LMON(2), LYEAR, LU, NUM, PI, P1OYR2, PI43, PRESS(31), PTEMP(31),
Q1, RADDEG, RADON, CLDRAD, R2, R3, SAYERA(30), SAYERB(30), SIGA,
SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIG20, SIGAP, SB, TEMP(31),
TOPSUR, TMUPI, ASDR, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
YPOS, IFLG1(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIME
INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
( QT1, VPAR(4)), ( QT2, VPAR(5)), ( QT3, VPAR(6)),
( AA, VPAR(7)), ( BB, VPAR(8)), ( CC, VPAR(9)),
( HEATN, VPAR(10)), (HEATM, VPAR(11)), ( HEATA, VPAR(12)),
( PCO, VPAR(14)), (PCO2, VPAR(15)),
( PAL203, VPAR(16)), (PHIL, VPAR(17)), ( GAMMAX, VPAR(18))

C

INPUT FORMAT STATEMENTS

100 FORMAT (12,1X,2A2,12)
101 FORMAT (10A2)
102 FORMAT (14,5X12,1X12,1X12,1X14)
103 FORMAT (14,3X12,1X2A2,1A1,1X14)
104 FORMAT (16,1X13,1XF4.1,F6.1,F6.1,F7.2,11XF7.2)

C

OUTPUT FORMAT STATEMENTS

200 FORMAT ("E&dB***NASA/MSFC MULTILAYER DIFFUSION MODEL - MOD3A"
4X"04 MAY 1977**")
201 FORMAT ("E&dBNUMBER OF RUNS AND COMMON DATA FILE NAME"
"(e.g. 01,DATA): E&dJ_")
202 FORMAT (5X"RUN "I2" WILL USE DATA FILE "3A2")
203 FORMAT ("E&dBRESEARCH OR E&dBPRODUCTION RUN: E&dJ_")
204 FORMAT (5X"RESEARCH RUN")
205 FORMAT (5X"PRODUCTION RUN")
206 FORMAT ("E&dBTOP OF SURFACE LAYER(M): E&dJ_")
207 FORMAT ("E&dBPSIGMA OF WIND AZIMUTH ANGLE: E&dJ_")

A-29
208 FORMAT ("TIME AND DATE"
  
  (e.g. 0800 EST 01 MAY 1976) : "dd-j")
209 FORMAT (5X,"LAUNCH TIME: "4A2/5X"LAUNCH DATE: "6A2)
210 FORMAT (""LAUNCH VEHICLE ("4212"
  
  "DELTA-THOR 2914",
  
  "DELTA-THOR 3914",
  
  "MINUTEMAN II")"
  
  ""Vehicles")
211 FORMAT (5X,"LAUNCH VEHICLE: "4A2)
212 FORMAT ("10(1H*)/1X,80(1H*)/1X,10(1H*)/60X,10(1H*)/
  
  1X,80(1H*)/1X,10(1H*)/60X,10(1H*)/
  
  0RUN "I2" USING DATA FILE "3A2/"
  
  "O3A2, A1" LAUNCH VEHICLE")
213 FORMAT (""I4 "E"A2,4X"DATE: "I2,1XA2,1,XI4")
214 FORMAT (""PREDICTION TIME: "4A2,4X"DATE: "I2,1XA2,1,XI4/
  
  "DATA FILE HEADER INFORMATION:"
  
  "NEXT RUN")
215 FORMAT (""O""(""OPEN ERROR "I4", PROCESSING CONTINUES WITH "
  
  "NEXT RUN")")
216 FORMAT (""O""(""READF ERROR "I4", PROCESSING CONTINUES WITH "
  
  "NEXT RUN")")
217 FORMAT (6X,40A2)
218 FORMAT (""5X"TIME: "4A2,4X
  
  DATE: "I2,1XA2,1,XI4")
219 FORMAT (""80(1H*)/1X,20(1HS),40X,20(1HS)/
  
  1X,20(1HS),16X"SOUNDING"16X,20(1HS)/
  
  1X,20(1HS),40X,20(1HS)/1X,80(1HS)/")
220 FORMAT (""80(1H*)/1X,20(1HF),40X,20(1HF)/
  
  1X,20(1HF),16X"FORECAST"16X,20(1HF)/
  
  1X,20(1HF),40X,20(1HF)/1X,80(1HF)/")
221 FORMAT (""SURFACE DENSITY (GM/M**3): "F8.2")
222 FORMAT (""LAYER ALTITUDE DIRECTION SPEED TEMP "
  
  "POT-TEMP D P TEMP PRESSURE"/
  
  "NO. (FEET) (METERS) (DEGREES) (H/SEC) "
  
  "(DEGREES CENTIGRADE) (MILLIBARS)"
  
  "(DEGREES CENTIGRADE) (MILLIBARS)"
  
  "SURFACE DENSITY (GM/M**3): "F8.2")
223 FORMAT (2XI2,17,2XI5,7XI3,5XF4.1,4XF4.1,4XF5.2,6XF4.1,6XF5.2)

C TYPE AND DIMENSION STATEMENTS

  C INTEGER BLANKS,FILE(3),DIGIT(50),RCHAR,VMNAME(4,5),
  
  1RUNUM,RA,FO,SDT,TE,ZEROO,GETTD(3),CLDRI(3)
  
  DIMENSION IPAR(5),VPARS(18,5),IVNAME(5),IDCB(144),IBUF(40),
  
  IALT(31),DPTEMP(31)

C DATA STATEMENTS

C C***********C************************C************************C************************
C*** VPARS(1-18)=SHUTTLE (19-36)=TITAN (37-54)=DELTA-THOR 2914
C*** (55-72)=DELTA-THOR 3914 (73-90)=MINUTEMAN II

A-30
C*************
DATA VPARS/1.521923E7,6.882968E6,3.441484E6,1.894794173E9,
  8.56929516E8,1.713859032E9,65221298911,4680846,
  .375,1479.7,1062.35,1000.0,.1970,2234,.0316,2791,
  0.002,.64,
  5.437528E6,2.718764E6,1.359382E6,3.2625168E8,
  1.6312584E8,3.2625168E8,4.29580469,5184223,
  5.0,2021.1,1010.55,1000.0,.1932,2665,.0222,
  .2819,.0002,.64,
  8.360685E5,9.09811E4,2.729434E5,2.887598E7,
  3.14229E6,1.885373E7,922156,.432703,.54,1766.0,
  1000.0,.690.0,1866.2055,.0156,.3391,.0002,.
  .50,
  1.057557E6,1.482923E5,.370731E5,.670269E7,
  9.398616E6,4.699308E7,1.2245756,.4180947,
  0.0,1449.9,1000.0,.411.18,1866.2055,.0156,.3391,
  .0002,.50,
  4.684476E5,4.684476E5,1.171119E5,.61806856E7,
  2.8106856E7,4.69982,.463333,.00,
  2055.9,2055.9,1000.0,1866.2055,.0156,.3391,
  .0002,.64/
DATA BLANKS/2H /, RCHAR/1HR/, RA/2HRA/, FO/2HFO/,
  SDT/2HDT/, TE/2HTE/, ZERO0/2H00/, NINE9/2HN9/,
  GETTD/2HGE,2HTT,1HD/, CLDRI/2HCL,2HDR,1HI/
DATA FDIGIT/2H01,2H02,2H03,2H04,2H05,2H06,2H07,2H08,2H09,2H10,
  2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H19,2H20,
  2H21,2H22,2H23,2H24,2H25,2H26,2H27,2H28,2H29,2H30,
  2H31,2H32,2H33,2H34,2H35,2H36,2H37,2H38,2H39,2H40,
  2H41,2H42,2H43,2H44,2H45,2H46,2H47,2H48,2H49,2H50/
DATA IYNAH/2HSH,2HTI,2HD2,2HD3,2HMM/
DATA VNAMES/2HSH,2HUT,CHTL,1HE,
  2HTI,2HTA,1HN,1H ,
  2HD-,2HT,2H29,2H14,
  2HD-,2HT,2H39,2H14,
  2HMI,2HMM,2HN,2HII/
C
C FIND THE LOGICAL UNIT NUMBER OF THE DEVICE TO BE USED FOR
C INPUT AND SET THE VARIABLE LU EQUAL TO IT
C
CALL RMPAR(IPAR)
LU = IPAR(1)
C
INITIALIZE SOME COMMON VARIABLES
C
LTIME = .FALSE.
YES = 1HY
PI = 3.141593
PI0VR2 = 0.5 * PI

A-31
PI43 = 1.3333333 * PI
TWOPI = 2.0 * PI
SQR2PI = SQRT(TWOPI)
DEGRAD = PI/180.0
RADDEG = 180.0/PI
DO 2 I=1,3
2 IOPT(I) = 0
ZB = 0.0
ZZ = 0.0
REFLEC = 1.0

WRITE THE HEADER OF THE CONSOLE

WRITE (LU,200)
READ (LU,100) NUMRUN,FILE(1),FILE(2),IOFF
NUMRUN = MINO(MAXO(NUMRUN,1),50)
IF(IOFF .GT. 0)IOFF = IFF - 1
IF(FILE(1) .NE. BLANKS)GO TO 5
FILE(1) = 2HDA
FILE(2) = 2HTA
IOFF = 0
5 IF(NUMRUN+IOFF .GT. 50)NUMRUN = 50 - IFF
DO 6 I=1,NUMRUN
    J = I + IFF
6 WRITE (LU,202) I,FILE(1),FILE(2),FDIGIT(J)

READ (LU,101) I
IF(I .EQ. RCHAR)IOPT(2) = 1
IF(IOPT(2) .EQ. 0)GO TO 7
WRITE (LU,204)
GO TO 12
7 WRITE (LU,205)

FOR PRODUCTION RUNS, READ IN THE TOP OF THE SURFACE LAYER
AND THE SIGMA OF THE WIND AZIMUTH ANGLE TO BE USED FOR ALL RUNS

WRITE (LU,206)
READ (LU,*),TOPSUR
WRITE (LU,207)
READ (LU,*),SIGA
READ IN AND WRITE OUT THE LAUNCH TIME AND DATE -- IF NOT
ENTERED, DO NOT WRITE ANYTHING OUT

12 WRITE (LU,209)
   READ (LU,101) (LAUNTD(I),I=1,10)
   IF(LAUNTD(1) .EQ. blanks)GO TO 17
   LTIME = .TRUE.
   CALL CODE
   READ (LAUNTD,102) LTIM,LDAY,LMON,LYEAR
   WRITE (LU,209) (LAUNTD(I),I=1,10)
   GO TO 21
17 LAUNTD(4) = SDT

READ IN THE LAUNCH VEHICLE, LET IT DEFAULT IF NOT ENTERED,
WRITE IT BACK OUT, AND FILL THE VPAR ARRAY WITH THE
APPROPRIATE VEHICLE PARAMETERS

21 WRITE (LU,210)
   READ (LU,101) J
   DO 24 I=1,5
   IF(J .EQ. IYNAME(I))GO TO 25
24 CONTINUE
   I = 1
25 IOPT(3) = I - 1
   WRITE (LU,211) (VNAME(J,I),J=1,4)
   DO 28 J=1,18
28 VPAR(J) = VPARS(J,I)

DO LOOP ON THE RUN NUMBER

DO 79 RUNNUM=1,NUMRUN

SET UP THE FILE NAME FOR THIS RUN, GET THE CURRENT TIME,
AND WRITE OUT THE HEADER

FILE(3) = FDIGIT(RUNNUM+1OFF)
ASSIGN 31 TO IRETRN
CALL EXEC(8,GETTD)
31 CONTINUE
   ITIME = ITIME + 100
   I = IOPT(3) + 1
   WRITE (6,212) RUNNUM,(FILE(J),J=1,3),(VNAME(J,I),J=1,4)
   IF(LTIME) WRITE (6,213) LTIM,LAUNTD(4),LDAY,LMON(1),LMON(2),LYEAR
   WRITE (6,214) ITIME,LAUNTD(4),IDAY,MONTH,LYEAR

OPEN THE DATA FILE FOR THIS RUN

CALL OPEN(IDCB,IER,FILE)
IF(IER .GE. 0)GO TO 32
WRITE (6,215) IERR
GO TO 79

READ THE HEADINGS FROM THE DATA FILE, SETTING UP THE
APPROPRIATE PARAMETERS

32 CALL READF(IDCB, IERR, IBUF, 40, LEN)
IF(IERR .GE. 0) GO TO 37

34 WRITE (6, 216) IERR
GO TO 79

37 IF(IBUF(1) .NE. TE) GO TO 32

39 WRITE (6, 217) (IBUF(I), I=1, LEN)
CALL READF(IDCB, IERR, IBUF, 40, LEN)
IF(IERR .LT. 0) GO TO 34

IF(IBUF(1) .NE. RA .AND. IBUF(1) .NE. FO) GO TO 39
IOPT(1) = 0
IF(IBUF(1) .EQ. FO) IOPT(1) = 1
WRITE (6, 217) (IBUF(I), I=1, LEN)
CALL READF(IDCB, IERR, IBUF, 40, LEN)
IF(IERR .LT. 0) GO TO 34
WRITE (6, 217) (IBUF(I), I=1, LEN)

READ THE SOUNDED/FORECAST TIME

CALL READF(IDCB, IERR, IBUF, 9)
IF(IERR .LT. 0) GO TO 34
CALL CODE
READ (IBUF, 103) ISTIM, ISDAY, ISMON(1), ISMON(2), ISYEAR

CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME

ISTIM = ISTIM - 500
IF(LAUNTD(4) .NE. 2HST) ISTIM = ISTIM + 100
IF(ISTIM .GT. 0) GO TO 41
ISTIM = 2400 + ISTIM
ISDAY = ISDAY - 1

WRITE OUT THE NEXT LINE OF THE HEADER

41 CALL READF(IDCB, IERR, IBUF, 40, LEN)
IF(IERR .LT. 0) GO TO 34
WRITE (6, 217) (IBUF(I), I=1, LEN)

WRITE OUT THE SOUNDED/FORECAST TIME

WRITE (6, 218) ISTIM, LAUNTD(4), ISDAY, ISMON(1), ISMON(2), ISYEAR

FIND THE FIRST DATA POINT WITH AN ALTITUDE OF 10 FEET
OR ABOVE

44 CALL READF(IDCB, IERR, IBUF, 40, LEN)
IF(IERR .LT. 0) GO TO 34
CALL B22(IBUF(1),J)
IF(J .LT. ZERO OR. J.GT.NINE) GO TO 44
CALL CODE
READ (IBUF,104) IALT(1),IDIR(1),SPEED(1),TEMP(1),DPTMP(1),
PRESS(1),SURDEN
IF(IALT(1) .LT. 10) GO TO 44
TRY TO FIND A TOTAL OF 30 DATA POINTS WITH ALTITUDES
BETWEEN 20 FT AND 10,000 FT INCLUSIVE

NUM = 1
DO 47 I=2,30
46 CALL READF(IDCB,IERR,IBUF,40,LEN)
IF(IERR.LT.0 AND. IERR.NE.-12) GO TO 34
IF(LEN .EQ. -1) GO TO 48
CALL B22(IBUF(1),J)
IF(J .LT. ZERO OR. J.GT.NINE) GO TO 46
CALL CODE
READ (IBUF,104) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTMP(I),
PRESS(I)
IF(IALT(I) .LT. 20 OR. IALT(I) .GT. 10000) GO TO 46
NUM = I
47  ZER0 OUT THE REMAINING ELEMENTS OF THE ARRAYS

48 NUM1 = NUM + 1
IF(NUM1 .GT. 30) GO TO 51
DO 49 I=NUM1,39
ALT(I) = 0.0
IDIR(I) = 0
SPEED(I) = 0.0
TEMP(I) = 0.0
DPTMP(I) = 0.0
PRESS(I) = 0.0
49

CONVERT TO METRIC UNITS

51 DO 52 I=1,NUM
ALT(I) = 0.3048 * FLOAT(IALT(I))
52 SPEED(I) = 0.515 * SPEED(I)

SORT ALL THE DATA POINTS SO THEY APPEAR IN ASCENDING
ORDER OF ALTITUDE

NUM1 = NUM - 1
DO 50 I=1,NUM1
JJ = NUM - I
DO 57 J=1,JJ
J1 = J + 1
IF(ALT(J) .LE. ALT(J1)) GO TO 57
ARG = ALT(J)
ALT(J) = ALT(J1)
ALT(J1) = ARG
IARG = IDIR(J)
IDIR(J) = IDIR(J1)
IDIR(J1) = IARG
ARG = SPEED(J)
SPEED(J) = SPEED(J1)
SPEED(J1) = ARG
ARG = TEMP(J)
TEMP(J) = TEMP(J1)
TEMP(J1) = ARG
ARG = DPTEMP(J)
DPTEMP(J) = DPTEMP(J1)
DPTEMP(J1) = ARG
ARG = PRESS(J)
PRESS(J) = PRESS(J1)
PRESS(J1) = ARG
57 CONTINUE
58 CONTINUE

CALCULATE THE POTENTIAL TEMPERATURE

DO 62 I=1,NUM
62 PTEMP(I) = (TEMP(I) + 273.15) * ((1000.0/PRESS(I))**0.288)

WRITE THE HEADER FOR SOUNDING OR FORECAST

IF(IOPT(I) .EQ. 1) GO TO 64
WRITE (6,219)
GO TO 65
64 WRITE (6,220)

WRITE THE SURFACE DENSITY AND ALL THE DATA POINTS

65 WRITE (6,221) SURDEN
WRITE (6,222)
DO 68 I=1,NUM
IALTF = 3.281 * ALT(I) + 0.5
IALTM = ALT(I) + 0.5
APTEMP = PTEMP(I) - 273.15
68 WRITE (6,223) I,IALTF,IALTM,IDIR(I),SPEED(I),TEMP(I),
APTEMP,DPTEMP(I),PRESS(I)

TRANSFER TO THE SEGMENT CLDRI -- THE CLOUD RISE MODEL

ASSIGN 75 TO IRETRN
CALL EXEC(8,CLDRI)
75 CONTINUE
**CLOSE THE DATA FILE**

CALL CLOSE(IDCDB)

**PROCESS THE NEXT RUN**

79 CONTINUE

STOP EXECUTION

STOP

END OF MOD3A

END

**SUBROUTINE DFEXP(J, CONC)**

******************************************************************************

**THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS**

J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER

CONC - CONCENTRATION TO BE TESTED

******************************************************************************

**COMMON BLOCK**

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,

IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,

ISMON(2), ISYEAR, IV2, JTOP, LAUNTD(10), LTIME, LTIM, LDAY,

LMON(2), LYEAR, LU, NUM, PI, PI0VR2, PI43, PRESS(31), PTEMP(31),

Q1, RADDEG, RATMC, CLDRAD, R2, R3, SAVPA(30), SAVER(30), SIGA,

SIGX0, SIGX, SPEED(31), SQRT2PI, SURDEN, SIGZ0, SIGAP, SB, TEMP(31),

TOPSUR, TWOPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,

YPOS, IFGLG(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIME

INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),

(QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),

(AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),

(HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),

(PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),

(P2L03, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

**CALCULATE SIGMA Z**

A-37
SIGZ = DIST * SIGAP + SIGZ0/1.28
R3 = 2.0 * SIGZ * SIGZ

CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION

TWOI = 2.0
ZT = ALT(J)
TEMP2 = CLDHT - ZZ
TEMP3 = CLDHT - 2.0 * ZB + ZZ
E1 = EXP(- TEMP2 * TEMP2/R3) +
     EXP(- TEMP3 * TEMP3/R3)

4 TEMP1 = TWOI * (ZT - ZB)
TEXPSM = E1
TEXP = (TEMP1 + TEMP2)**2/R3
IF(TEXP .LE. 120.0) E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 + TEMP2)**2/R3
IF(TEXP .LE. 120.0) E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 - TEMP3)**2/R3
IF(TEXP .LE. 120.0) E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 + TEMP3)**2/R3
IF(TEXP .LE. 120.0) E1 = E1 + EXP(- TEXP)
IF(TEHP .LE. 120.0) E1 = E1 + EXP(- TEXP)

IF(TEHP .LE. 0.0) GO TO 7
TWOI = TWOI + 2.0
GO TO 4

7 E1 = REFLEC * E1

CALCULATE SIGMA Y

S8 = DIST * SIGAP + SIGX
R2 = SQRT(S8 * S8 + (0.0040589 * FLOAT(IDIR(J) - IDIR(1)) * 
     DIST)**2)

CALCULATE CLOUD LENGTH

TEMP1 = SPEED(J) - SPEED(1)
AL1 = 0.28 * TEMP1 * DIST/ASPD
IF(TEMP1 .GE. 0.0) GO TO 11
IF(PTEMP(J)-PTEMP(1) .GT. 0.0) AL1 = 0.0
AL1 = ABS(AL1)

CALCULATE SIGMA X

11 SIGX = SQRT((AL1/4.3)**2 + SIGX0 * SIGX0)

IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
TO THE CALLING PROGRAM

IF(CONC .EQ. 1000.0) RETURN

CALCULATE CROSS WIND DISTANCE
Y1 = - 2.0 * R2 * R2 * ALOG(15.7496 * CONC * SIGX.* R2 *
        SIGZ/(Q1 * E1))
Y1 = SQRT(AMAX1(Y1, 0.0))
RETURN TO THE CALLING PROGRAM
RETURN
END OF DFEXP
END
SUBROUTINE ORGIN(IX0, IY0)

***********************************************************************

THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
FOR THE COMPLEX AND MAP SELECTED

************************************************************************

COMMON BLOCK

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
        IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
        ISMON(2), ISYEAR, IV2, JTUP, LUNUTD(10), LTIME, LTIM, LDAY,
        LMON(2), LYEAR, LU, NUM, PI, PIDVR2, PI43, PRESS(31), PTEMP(31),
        Q1, RADDEC, RATOMC, CLDRAD, R2, R3, SAVEA(30), SAYER(30), SIGA,
        SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SICAPA, SB, TEM(31),
        TOSUR, TWOPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
        YPOS, IFLG1(5), ZB, ZZ, REFLEC, IRETRH
LOGICAL LTIME
INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
        (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
        (DA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
        (HEATN, VPAR(10)), (HEATN, VPAR(11)), (HEATA, VPAR(12)),
        (PHCL, VPAR(13)), (PC0, VPAR(14)), (PCO2, VPAR(15)),
        (PAL2O3, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

INPUT FORMAT STATEMENT

100 FORMAT (I2, 1X, A1)

OUTPUT FORMAT STATEMENT

200 FORMAT ("E&dBENTER COMPLEX, E&DFSE&dBEA OR E&DFLE&dBAND MAP =
        "(e.g. 17, L); E&dJ-")
TYPE AND DIMENSION STATEMENTS

LOGICAL NOTIST
INTEGER SCHAR
DIMENSION IX(8), IY(8)

DATA STATEMENTS

DATA NOTIST/.FALSE./, SCHAR/1HS/
DATA IX/8730,4100,5411,4825,8750,4100,5450,4830/,
IY/8600,7300,8243,8050,2990,1700,2630,2465/

IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
COORDINATES, I, AGAIN

IF(NOTIST)GO TO 7

THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
AND THE DESIRED MAP, i.e. SEA OR LAND

WOTIST = .TRUE.
WRITE (LU,200)
READ (LU,100) I,J

CALCULATE I AS THE INDEX OF THE COORDINATES FOR THE COMPLEX
AND MAP ASKED FOR -- DEFAULT IS COMPLEX 17, LAND MAP

K = 0
IF(J .EQ. SCHAR)K = 4
J = I - 37
IF(J.LT.2 .OR. J.GT.4)J = 1
J = J + K

SET THE COORDINATES BASED ON THE INDEX I

IXO = IX(I)
IYO = IY(I)

RETURN TO THE CALLING PROGRAM

RETURN

END SUBROUTINE SYMBL(IWIDE,IHI,ISYMB)
IX=-IWIDE/2
IY=-IHI/2
SUBROUTINE B2Z(IAM, IB)
  IB = IAM(DIA, 1, 177400B)
  IF(IB .EQ. 000000B) IB = 030000B
  IC = IAM(DIA, 000037B)
  IF(IC .EQ. 000004B) IC = 000060B
  IB = IB .OR. IC
  RETURN
END

C*************************************************************************
C
C THIS SEGMENT RETURNS THE CURRENT TIME, DAY, MONTH, AND YEAR

C*************************************************************************

COMMON BLOCK

COMMON ALT(31), AL(1), CMAX, CONCPK, DEGRAD, ADIR, DOSPK, E(1), CLDHT,
  IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTD, ISDAY,
  ISMN(2), IYEAR, IY2, JTOP, LAUNTD(10), LTME, LTIME, LDAY,
  LMON(2), LYEAR, LU, NUM, P1, P2VQ2, PI, 43, PRESS(31), PTEMP(31),
  P1, RADEG, RATOMIC, CLDRAQ, R2, R3, SAYCA(30), SAVR(30), SIGA,
  SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
  TPOSUR, TWOPI, ASPD, VPAR(18), CTIME(31), DIST, YES, Y1, NUMRUN,
  YPOS, IFLG1(5), ZB, ZZ, REFLEC, RETRN
LOGICAL LTME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
  (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
  (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
  (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
  (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
  (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

C

TYPE AND DIMENSION STATEMENTS

INTEGER DAYMON(12)
DIMENSION MONTHS(2, 12), IT(5)

C

DATA STATEMENTS
CALL EXEC TO RETURN CURRENT TIME, JULIAN DAY, AND YEAR

CALL EXEC(11, IT, IYEAR)

USE JUST HOURS AND MINUTES FOR THE TIME

ITIME = 100 * IT(4) + IT(3)

MAKE APPROPRIATE ADJUSTMENTS IF THIS IS A LEAP YEAR

DAYMON(2) = 28
I = IYEAR/4
IF(4*I .EQ. IYEAR) DAYMON(2) = 29

CONVERT THE JULIAN DAY INTO A MONTH AND A DAY

IDAY = IT(5)
DO 7 I=1, 12
    IDAY = IDAY - DAYMON(I)
    IF(IDAY .LE. 0) GO TO 12
7 CONTINUE
12 IDAY = IDAY + DAYMON(I)
MONTH(1) = MONTHS(1, I)
MONTH(2) = MONTHS(2, I)

RETURN TO THE APPROPRIATE PLACE IN MOD3A

GO TO IRETRN
17 CALL MOD3A

END OF GETTD

END PROGRAM CLDRI, 5

******************************************************************************

CLOUD RISE PROGRAM -- A SEGMENT OF THE MOD3A PROGRAM

******************************************************************************

COMMON BLOCK
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOPSK, E1, CLDHT,
. IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
. ISMON(2), ISYEAR, IV2, JTOP, LAVNTD(10), LTIME, LTIM, LDAY,
. LMON(2), LYEAR, LU, NUM, PI, PI0VR2, PI43, PRESS(31), PTEMP(31),
. Q1, RADIEM, RCAI0CH, CLRAD, R2, R3, SAV42, SAV52, SAV7(30), SIGA,
. SIGX, SIGY, SIGZ, SQR2I, SDRW, SIGQ, S8, TEMPS(31),
. TOPSUR, TU0PI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
. YPOS, IFLG1(5), ZB, ZZ, REFLEC, IRETRM

LOGICAL LTIME
INTEGER YES

EQUIVALENCE ((QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
. (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
. (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
. (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
. (PHCLS, VPAR(13)), (PCOS, VPAR(14)), (PCO2, VPAR(15)),
. (PAL203, VPAR(16)), (PHO, VPAR(17)), (GAMMAX, VPAR(18))

C
C OUTPUT FORMAT STATEMENTS
C
C 200 FORMAT (*1"27X"EXHAUST CLOUD"/*0LEVEL"4X"ALTITUDE"17X
. "RISE TIME"5X"RANGE"6X"DIRECTION"/1OX"(METERS)"17X
. ":4X"(DEGREES)"
C 201 FORMAT (2X13,5XF7.1,5X"ADIABATIC"5XF6.1,6XF7.1,7XF5.1)
C 202 FORMAT (2X13,5XF7.1,6X"STABLE"7XF6.1,6XF7.1,7XF5.1)
C 203 FORMAT ("0****CLOUD STABILIZATION****/
. 6X"HEIGHT(M): "F6.1/
. 6X"STABILIZATION TIME AFTER LAUNCH(SEC): "F5.1/
. 6X"RANGE FROM PAD(M): "F7.1/
. 6X"DIRECTION FROM PAD(DEC): "F5.1)
C 204 FORMAT ("ESTIMATED TOP OF SURFACE LAYER(M): "F6.1)
C 205 FORMAT ("ESTIMATED TOP OF SURFACE LAYER METEOROLOGICAL PARAMETERS"
. "****/
. 6X"HEIGHT(M): "F6.1/
. 6X"WIND DIRECTION(DEC): "I3/
. 6X"WIND SPEED(M/SEC): "F4.1)
C 206 FORMAT ("0****DIFFUSION PARAMETERS****/
. 6X"MEAN SPEED(M/SEC): "F4.1/
. 6X"MEAN TRANSPORT DIRECTION(DEC): "F5.1)
C 207 FORMAT ("SIGMA OF WIND AZIMUTH ANGLE, SIGA: "F4.1)
C 208 FORMAT ("SIGMA OF WIND AZIMUTH ANGLE, SIGA: "F4.1)
C 209 FORMAT ("SIGMA OF WIND AZIMUTH ANGLE, SIGA: "F4.1)
C 210 FORMAT ("0EFFECTIVE CLOUD HEIGHT(M): "F6.1)

C
C TYPE AND DIMENSION STATEMENTS
C
INTEGER CONC(3)
DIMENSION IAS(31)
C
DATA STATEMENT
DATA CONC/2HCO,2HNC,2H

INITIALIZE SOME LOCAL VARIABLES

CRTIME( ) - CLOUD RISE TIME
IAS( ) - 0 = ADIABATIC
1 = STABLE
ALTINC - ALTITUDE INCREMENT
ITERAT - ITERATION COUNTER

RNGY = 0.0
RNGX = 0.0
CRTIME(1) = 0.0
ALTINC = 0.0
SAVER(I) = 0.0
SAVEA(I) = 0.0
ITERAT = 0

WRITE OUT THE EXHAUST CLOUD HEADER

WRITE (6,200)

CALCULATE SOME QUANTITIES TO BE USED IN SUBSEQUENT DO LOOP

ALPHAC = 5.12913086E-2 * (TEMP(1) + 273.15) * SURDEN *
        GAMMAX**3/(HEATH * QC1)
GT = 9.8/(TEMP(1) + 273.15)

DO LOOP TO CALCULATE EXHAUST CLOUD PARAMETERS

DO 9 I=2,NUM

IM1 = I - 1
IAS(I) = 1

CALCULATE SLOPE OF POTENTIAL TEMPERATURE, SPEED, AND DIRECTION IN LAYER

DALT = ALT(I) - ALT(IM1)
GPTEMP = (PTEMP(I) - PTEMP(IM1))/DALT
GSPEED = (SPEED(I) - SPEED(IM1))/DALT
GDIR = FLOAT(IDIR(I) - IDIR(IM1))/DALT

CALCULATE METEOROLOGICAL AND ENERGY FACTOR

Z = ALT(I) - ALT(IM1)
ALPHA = ALPHAC * Z**4/(AA * Z**BB + CC)

CALCULATE POTENTIAL TEMPERATURE FACTOR
STAB = GT * (PTEMP(I) - ALTINC * GPTEMP - PTEMP(I)) / (ALT(I) - ALTINC - ALT(I) + 1.0E-7)

CALCULATION FOR ADIABATIC RISE

IF(STAB .GT. 0.000001) GO TO 4
CRTIME(I) = SQRT(ALPHA)
IAS(I) = 0
GO TO 6

CALCULATION FOR STABLE CLOUD RISE

C2 - ARGUMENT OF ARC COSINE (MUST BE LESS THAN -1)

4 C2 = 1.0 - 0.5 * ALPHA * STAB
IF(C2 .LT. -1.0) GO TO 5
C3 = C2/SQRT(1.0 - C2 * C2)
CRTIME(I+ITERAT) = (PI0VR2 - ATAN(C3)) / SQRT(STAB)
IF(ITERAT .EQ. 1) GO TO 11
GO TO 6

ITERATE IN LAYER

5 ALTINC = ALTINC + 5.0
ITERAT = 1
GO TO 2

CALCULATE RANGE AND DIRECTION

6 DELRNG = - 0.5 * (SPEED(IM1) + SPEED(I)) * (CRTIME(IM1) - CRTIME(I))
DELDIR = 0.00872665 * FLOAT(IDIR(I) + IDIR(IM1))
RNGY = RNGY - DELRNG * SIN(DELDIR)
RNGX = RNGX - DELRNG * COS(DELDIR)
AZMUTH = RADDEG * ATAN2(RNGY, RNGX)
IF(AZMUTH .LT. 0.0) AZMUTH = AZMUTH + 360.0
DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
SAVER(I) = DELRNG
SAVER(A) = AZMUTH

WRITE OUT THE VARIABLES WITH THE APPROPRIATE FORMAT STATEMENT BASED OF WHETHER OR NOT CLOUD IS ADIABATIC OR STABLE

IF(IAS(I) .NE. 0) GO TO 8
WRITE (6,201) I, ALT(I), CRTIME(I), DELRNG, AZMUTH
GO TO 9

8 WRITE (6,202) I, ALT(I), CRTIME(I), DELRNG, AZMUTH
9 CONTINUE
CALCULATE AND WRITE OUT STABILIZATION HEIGHT AND TIME

11 DELRNG = 0.5 * (SPEED(IM1) - ALTINC * GSPEED + SPEED(I)) * (CRTIME(I + 1) - CRTIME(IM1))
DALT = 0.00872665 * (FLOAT(IDIR(I) + IDIR(IM1)) - GDIR * ALTINC)
RNGY = RNGY - DELRNG * SIN(DALT)
RNGX = RNGX - DELRNG * COS(DALT)
AZMUTH = RADDEG * ATAN2(RNGY, RNGX)
IF(AZMUTH .LT. 0.0) AZMUTH = AZMUTH + 360.0
DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
ALT(31) = ALT(I) - ALTINC
WRITE (6,203) ALT(31),CRTIME(I+1),DELRNG,AZMUTH

STORE THE INDEX OF THE ESTIMATED TOP OF THE SURFACE LAYER
JTOP = I + 1

LOAD THE CLOUD RISE TIME ARRAY
CRTIME(31) = CRTIME(JTOP)
DO 15 J=I,NUM
15 CRTIME(I) = CRTIME(31)

IS THIS A RESEARCH OR A PRODUCTION RUN?
IF(IOPT(2) .NE. 0) GO TO 22

PRODUCTION RUN -- IF TOPSUR IS UNDEFINED, USE JTOP AS ESTIMATED
17 IF(TOPSUR.LE.0.0) GO TO 24

CALCULATE JTOP BASED ON VALUE OF TOPSUR
LEASTD = 9999999.9
DO 19 I=1,NUM
DIFF = ABS(ALT(I) - TOPSUR)
IF(DIFF.GT. LEASTD) GO TO 19
LEASTD = DIFF
JTOP = I
19 CONTINUE
GO TO 24

WRITE OUT THE ESTIMATED TOP OF SURFACE LAYER -- READ IN THE ONE TO BE USED -- CALCULATE JTOP
22 WRITE (LU,204) ALT(JTOP)
WRITE (LU,205)
READ (LU,*) TOPSUR
GO TO 17
WRITE OUT THE TOP OF THE SURFACE LAYER AND WIND DIRECTION
AND SPEED AT THE TOP

24 TOPSUR = ALT(JTOP)
WRITE (6,206) TOPSUR,IDIR(JTOP),SPEED(JTOP)

CALCULATE SOURCE STRENGTH

Q1 = 1.289E9 * (TEM(1) + 273.15)/PRESS(1) * TOPSUR**0.4837

CALCULATE AND WRITE OUT THE MEAN WIND SPEED, ASPD, AND
DIRECTION, ADIR

DO 28 I=2,JTOP
IF(IABS(IDIR(I) - IDIR(I - 1)) .LT. 180)GO TO 28
DO 27 J=1,JTOP
27 IF(IDIR(J) .LT. 180)IDIR(J) = IDIR(J) + 360
GO TO 31
28 CONTINUE

ASPD = 0.0
ADIR = 0.0

DO 32 I=2,JTOP
IM1 = I - 1
DALT = ALT(I) - ALT(IM1)
ASPD = ASPD + 0.5 * (SPEED(I) + SPEED(IM1)) * DALT
ADIR = ADIR + 0.5 * FLOAT(IDIR(I) + IDIR(IM1)) * DALT

DO 34 I=1,JTOP
34 IF(IDIR(I) .GT. 360)IDIR(I) = IDIR(I) - 360

DALT = ALT(JTOP) - ALT(1)
ASPD = ASPD/DALT
ADIR = ADIR/DALT
IF(ADIR .GT. 180.0)GO TO 35
ADIR = ADIR + 180.0
GO TO 36
35 ADIR = ADIR - 180.0

36 WRITE (6,207) ASPD,ADIR

IS THIS A RESEARCH OR A PRODUCTION RUN?

IF(IOPT(2) .EQ. 0)GO TO 45

RESEARCH RUN -- READ IN SIGA

WRITE (LU,208)
READ (LU,*) SIGA
WRITE OUT SIGA, THE SIGMA OF THE WIND AZIMUTH ANGLE

45 WRITE (6,209) SIGA

SIGAP = 0.0087266 * SIGA

CALCULATE THE HORIZONTAL AND VERTICAL CLOUD DIMENSIONS,

i.e. SIGXO AND GSPACE

SIGXO = 0.297674 * ALT(31)
GSPACE = 0.232558 * ALT(31)

CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, CLDHT

CLDHT = ALT(31)
CLUDRAD = 2.15 * SIGXO
IV2 = 0
IF(CLDRAD+ALT(31) .GE. ALT(JTOP)) IV2 = 1
SIGZO = SIGXO
IF(IV2 .EQ. 1) SIGZO = (ALT(JTOP) - ALT(31) + CUDRAD)/4.3
IF(SIGZO .LT. 0.0) GO TO 47
CLUDHT = 0.5 * ALT(JTOP)
SIGZO = 0.64 * CLDHT/2.15
GO TO 49
47 IF(IV2 .EQ. 1) CLDHT = 0.5 * (ALT(JTOP) + ALT(31) - CLDRAD)

49 WRITE (6,210) CLDHT

CALL THE SEGMENT CONC

CALL EXEC(8,CONC)

END OF CLDRI

END PROGRAM CONC,5

*******************************************************************************

CONCENTRATION AND DOSAGE PROGRAM -- A SEGMENT OF THE MOD3A PROGRAM

*******************************************************************************

COMMON BLOCK

COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLUDHT,
IDIR(31),I0PT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
ISNON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
LOGICAL LTIME
INTEGER YES
EQUVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
         (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
         (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
         (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
         (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
         (PAL203, VPAR(16)), (PHO, VPAR(17)), (GAMMAX, VPAR(18))

C
C  INPUT FORMAT STATEMENT

100 FORMAT (A1)

C
C  OUTPUT FORMAT STATEMENTS

201 FORMAT (//'E&dBCENTERLINE CONCENTRATION PLOT DESIRED?"
         
         "(E&dFY&dIBES OR E&dFN&dIBO): E&dJ_"

202 FORMAT (5X"NO")
203 FORMAT (5X"YES")
204 FORMAT ("1"12X"CLOUD CONCENTRATIONS AND DOSAGES"/
         
         "0DISTANCE"4X"CONCENTRATION"5X"DOSAGE"6X
         "TIME AFTER LAUNCH(SEC)"/
         
         "(METERS)"8X"(PPM)"8X"(PPM SEC)"8X"START"3X"FINISH")
205 FORMAT (1XF7.1,8XF7.3,8XF7.3,9XF5.1,3XF5.1)
206 FORMAT ("0****POINT OF MAXIMUM CONCENTRATION****"/
         
         6X"RANGE FROM PAD(M): "F8.1/
         
         6X"DIRECTION(DEG): "F5.1/
         
         6X"HEIGHT(M): 2.0"/
         
         6X"MAXIMUM CONCENTRATION(PPM): "F6.3")
207 FORMAT ("E&dBOFF-CENTER CONCENTRATIONS DESIRED?"
         
         "(E&dFY&dIBES OR E&dFN&dIBO): E&dJ_"
208 FORMAT ("0****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE"
         
         "UNCERTAINTIES****")
209 FORMAT ("E&dBRANGE(M), AZIMUTH(DEG)"
         
         "(O RANGE TERMINATES PROCEDURE): E&dJ_"
210 FORMAT ("0"5X"RANGE(M): "F7.1/
         
         6X"AZIMUTH(DEG): "F5.1/
         
         6X"MATERIAL"5X"CONCENTRATION(PPM)"11X"DOSAGE(PPM)"
211 FORMAT (415,12)
212 FORMAT (7X3A2,6XF8.3/,+- "F8.3, 4XF8.3" /+- "F8.3")
213 FORMAT ("E&dBISOPLETH PLOT DESIRED?"
         
         "(E&dFY&dIBES OR E&dFN&dIBO): E&dJ_"

C
C  TYPE AND DIMENSION STATEMENTS

A-49
C
LOGICAL IGRAF
DIMENSION FACT(3),CMPL(3),DMPL(3),MAT(3,5),ISOPO(3)
C
DATA STATEMENTS
C
DATA FACT/0.0,-0.174533,0.174533/
DATA MATS/2H,2HHC,2HL,2H,2H,C,H2O,
         2H,2HC0,2H2,2H,A,2HL2,2H03,
         2H,2H,H,2H0 /
DATA ISOPO/2H1S,2HOP,1H0/
C
IF THIS IS A RESEARCH RUN, DETERMINE IF PLOTTING IS DESIRED
C
IF(IOPT(2).EQ.0)GO TO 55
WRITE (LU,201)
READ (LU,100) I
IF(I.EQ.YES)GO TO 54
IGRAF = .FALSE.
WRITE (LU,202)
GO TO 55
54 IGRAF = .TRUE.
WRITE (LU,203)
C
DO LOOP FOR CONCENTRATION AND DOSAGE CALCULATIONS
C
DIST - RANGE FROM STABILIZATION
DOSPK - DOSAGE
DOSMAX - MAXIMUM DOSAGE
CONCPK - CONCENTRATION
CONMAX - MAXIMUM CONCENTRATION
C
55 CONMAX = 0.0
DOSMAX = 0.0
ACTVOL = PI43 * CLDRAD * CLDRAD * CLDRAD
TOTVOL = ACTVOL
IF(IY2.EQ.1)ACTVOL = PI * (ALT(JTOP) + CLDRAD - ALT(31))**2 *
(2.0 * CLDRAD - ALT(JTOP) + ALT(31))/3.0
Q1 = Q1 * ACTVOL/TOTVOL
C
WRITE (6,204)
C
DO 59 I=0,20000,250
C
DIST = I
C
CALL DFEXP(JTOP,1000.0)
C
DOSPK = Q1 * E1/(TWOPI * R2 * ASPD * SQRT(0.5 * R3))

A-50
CONCPK = DOSPK * ASPD / (SQR2PI * SIGX)

IF(IGRAPH)CALL CPLOT

DOSMAX = AMAX1(DOSPK, DOSMAX)

IF(CONCPK .LE. CONMAX) GO TO 58
RATOMC = DIST
CONMAX = CONCPK
SGMAX = SIGX
SGYMAX = SIGY

58 IF(AMOD(DIST, 1000.0) .NE. 0.0) GO TO 59

ARC1 = CRTIME(31) + (DIST - AL1) / ASPD
ARC2 = CRTIME(31) + (DIST + AL1) / ASPD
WRITE (6, 205) DIST, CONCPK, DOSPK, ARC1, ARC2

59 CONTINUE

CALCULATE AND WRITE OUT THE POINT OF MAXIMUM CONCENTRATION

ARG1 = DEGRAD * ADIR
DIST = RATOMC * COS(ARG1)
Y1 = RATOMC * SIN(ARG1)

DO 62 I = 2, JTOP
IF(CLDHT .LE. ALT(I)) GO TO 63
62 CONTINUE
I = JTOP

63 IM1 = I - 1
RANGSR = SAYER(IM1) + (CLDHT - ALT(IM1)) * (SAVER(I) - SAYER(IM1)) / (ALT(I) - ALT(IM1))

ARG1 = SAYEA(I) - SAYEA(IM1)
IF(ABS(ARG1) .LT. 180.0) GO TO 66
IF(ARG1 .GT. 0.0) GO TO 65
SAVEA(I) = SAYS(A) + 360.0
GO TO 66
65 SAYS(A) = SAYS(A) + 360.0

66 AZCS = SAYS(A) + (CLDHT - ALT(IM1)) * (SAYS(A) - SAYS(A)) / (ALT(I) - ALT(IM1))
IF(AZCS .GE. 360.0) AZCS = AZCS - 360.0

ARG1 = DEGRAD * AZCS
X2 = RANGSR * COS(ARG1)
Y2 = RANGSR * SIN(ARG1)
X = DIST + X2

A-51
\[ Y = Y_1 + Y_2 \]

\[ \text{RNGE} = \sqrt{X \times X + Y \times Y} \]

\[ \text{DIR} = \text{RADDEG} \times \text{ATAN2}(Y, X) \]

IF (DIR .LT. 0.0) DIR = DIR + 360.0

WRITE (6, 206) RNGE, DIR, CONMAX

IF THIS IS A PRODUCTION RUN, SKIP THE OFF CENTER CONCENTRATION SECTION AND THE CALL OF SEGMENT ISOPO -- IF PLOTTING WAS NOT REQUESTED, JUST SKIP THE OFF CENTER CONCENTRATION SECTION

IF (ICGRAF) GO TO 68

IF (IOPT(2) .EQ. 0) GO TO 88

GO TO 81

OFF CENTER CONCENTRATIONS SECTION

68 CALL LABEL(JTOP)

ARE OFF CENTER CONCENTRATIONS DESIRED?

WRITE (LU, 207)

READ (LU, 100) I

IF (I .NE. YES) GO TO 78

OFF CENTER CONCENTRATIONS ARE DESIRED

WRITE (LU, 203)

WRITE (6, 208)

CALL ORGIN(IXSET, IYSET)

ARG1 = 0.0

IF (ADIR .GT. 180.0) ARG1 = 360.0

BETAF = DEGRAD * (180.0 + ARG1 - ADIR)

ARG1 = 0.0

IF (AZCS .GT. 180.0) ARG1 = 360.0

BETAS = DEGRAD * (180.0 + ARG1 - AZCS)

XP = RANGSR * COS(BETAS)

YP = RANGSR * SIN(BETAS)

ITER = 0

LOOP ON OFF CENTER CONCENTRATION REQUESTS

71 ITER = ITER + 1

READ IN AND WRITE OUT THE RANGE AND AZIMUTH FOR THE OFF CENTER CONCENTRATION CALCULATION -- ENTERING A RANGE OF 0

A-52
C TERMINATES THE PROCEDURE
C
WRITE (LU,209)
READ (LU,*) RP,AZP
IF(RP .LE. 0.0)GO TO 81
WRITE (6,210) RP,AZP
C
ARG1 = 0.0
IF(AZP .GT. 180.0)ARG1 = 360.0
RP = DEGRAD * (180.0 + ARG1 - AZP)
XS = RP * COS(AP)
YS = RP * SIN(AP)
C
ON THE PLOTTER, WRITE OUT AN ASTERISK AND THE ITERATION
NUMBER AT THE LOCATION WHERE THE OFF CENTER CONCENTRATION
CALCULATION IS DESIRED
C
IX = IXSET + 0.2631 * XS
IY = IYSET + 0.3545 *YS
WRITE (12) -1,1,IX,IY
CALL SYMBLC(100,125,1H*)
IX = IX + 75
WRITE (12) -1,1,IX,IY
WRITE (12,211) 100,0,0,0,125,ITER
C
CALCULATE THE CONCENTRATIONS AND DOSAGES AT THIS POINT PLUS
10 DEGREES UNCERTAINTIES ON EITHER SIDE
C
XHAT = XS - XP
YHAT = YS - YP
C
DO 74 I=1,3
ARG1 = BETAF - FACT(I)
Y = - XHAT * SIN(ARG1) + YHAT * COS(ARG1)
CALL DFEXP(JTOP,1000.0)
DOS = Q1 * E1 * EXP(- Y * Y/(2.0 * R2 * R2))/
     (TWOPI * R2 * ASPD * SQRT(0.5 * R3))
CONC = DOS * ASPD/(SQR2PI * SIGX)
CMNPL(1) = CONC
74 QHNPL(1) = DOS
C
CALCULATE AND WRITE OUT THE CONCENTRATION AND DOSAGE FOR
EACH MATERIAL
C
DELC = ABS(0.5 * (2.0 * CMNPL(1) - CMNPL(2) - CMNPL(3)))
DELD = ABS(0.5 * (2.0 * DMNPL(1) - DMNPL(2) - DMNPL(3)))
WRITE (6,212) (MATS(1,1),I=1,3),CMNPL(1),DELC,DMNPL(1),DELD
C
ARG1 = PCO/PHCL
CONC = ARG1 * CMNPL(1)

A-53
DLC = ARG1 * DELC
DOS = ARG1 * DMHPL(1)
DLD = ARG1 * DELD
WRITE (6,212) (MATS(I,2),I=1,3),CONC,DLC,DOS,DLD

ARG1 = PC02/PHCL
CONC = ARG1 * CMNPL(1)
DLC = ARG1 * DELC
DOS = ARG1 * DMNPL(1)
DLD = ARG1 * DELD
WRITE (6,212) (MATS(I,1),I=1,3),CONC,DLC,DOS,DLD

ARG1 = PAL203/PHCL * 0.43882420 * PRESS(1)/
       (TEMP(1) + 273.16)
CONC = ARG1 * CMNPL(1)
DLC = ARG1 * DELC
DOS = ARG1 * DMNPL(1)
DLD = ARG1 * DELD
WRITE (6,212) (MATS(I,3),I=1,3),CONC,DLC,DOS,DLD

ARG1 = PCN0/PHCL
CONC = ARG1 * CMNPL(1)
DLC = ARG1 * DELC
DOS = ARG1 * DMNPL(1)
DLD = ARG1 * DELD
WRITE (6,212) (MATS(I,4),I=1,3),CONC,DLC,DOS,DLD

REQUEST ANOTHER POINT FOR AN OFF CENTER CONCENTRATION
CALCULATION
GO TO 71
OFF CENTER CONCENTRATIONS ARE NOT DESIRED
78 WRITE (LU,202)

IS AN ISOPLETH PLOT DESIRED?
81 WRITE (LU,213)
READ (LU,100) I

IF AN ISOPLETH PLOT IS DESIRED, CALL THE SEGMENT ISOPO

IF(I .'NE. YES)GO TO 87
WRITE (LU,203)
CALL EXEC(8,ISOPO)
87 WRITE (LU,202)
RETURN TO THE APPROPRIATE PLACE IN MOD3A
88 GO TO IRETRN
89 CALL MOD3A

C END OF CONC C

END

SUBROUTINE CPLOT

C**********************************************************************
C
C THIS SUBROUTINE PLOTS THE CONCENTRATION AND DOSAGE CENTERLINE
C
C**********************************************************************

COMMON ALI~(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
. IDIR~31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
. ISMON(2), ISYEAR, IY2, JTOP, LAUNTD(10), LTIME, LTIM, LDAY,
. LMON(2), LNEAR, LU, NUM, PI, PI0VR2, P143, PRESS(31), PTEMP(31),
. Q1, RADDEG, RATOMIC, CLDRAD, R2, R3, SAVEA(30), SAVER(30), SIGA,
. SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
. TOPSUR, TV0PI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
. YPOS, IFLG1(5), ZB, Z2, REFLEC, IRETRN

LOGICAL LTIME
INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
. (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
. (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
. (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
. (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
. (PAL203, VPAR(16)), (PHO, VPAR(17)), (GAMMAX, VPAR(18))

IEXPC=0
IEXPD=IEXPC+2
IX=DIST*9295./30000.+725.
IYC=CONCPK*8231./10.**(IEXPC+1)+1040.
IYD=DOSPK*8231./10.**(IEXPD+1)+1040.
IF(DIST.NE.0.) GO TO 30
WRITE(12) -1,1,IX,IYD
CALL SYMBOL(100, 100, 25400B)
WRITE(12) -1,1,IX,IYC

RETURN TO THE CALLING PROGRAM

RETURN
RETURN TO THE CALLING PROGRAM

RETURN

END OF CPLOT

END

SUBROUTINE LABEL(J2)

C**********************************************************
C
C THIS SUBROUTINE LABELS THE CONCENTRATION AND DOSAGE CENTERLINE
PLOTS

C**********************************************************
C
COMMON BLOCK

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
. IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
. ISMON(2), IYEAR, IV2, JTOP, LAUHTD(10), LTIME, LTIM, LDAY,
. LMON(2), LYEAR, LU, NUM, PI, P10VR2, PI43, PRESS(31), PTEMP(31),
. QI, RADIUS, RATOMC, CLDRAD, R2, R3, SAVEA(30), SAVR(30), SIGA,
. SIGX0, SIGX, SPEED(31), SQRT2PI, SURDEN, SIGZ0, SIGAP, S0, TEMP(31),
. TOPSUR, TWOPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
. YPOS, IFILG1(5), ZB, ZZ, REFLAC, IRETRN

LOGICAL LTIM

INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
. (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
. (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
. (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
. (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
. (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

C

OUTPUT FORMAT STATEMENTS

200 FORMAT (415, I2)
201 FORMAT (415, F5.0)
202 FORMAT (415, F5.2)
203 FORMAT (415, I4, ES*A2, 2X*I2, 1X*A2, A1, 1X*I4)

C

LABEL THE PLOT
IEXP=0
IEXPD=IEXP+2
NEXP=-IEXP
NEXPD=-IEXPD
WRITE (12) -1,1,300,5000
WRITE (12,200) 0,150,-100,0,NEXP
WRITE (12) -1,1,300,6500
WRITE (12,200) 0,150,-100,0,NEXPD
WRITE (12) -1,1,3700,8950
WRITE (12,201) 125,0,0,125,CLDHT
WRITE (12) -1,1,3700,8745
WRITE (12,201) 125,0,0,125,CRTIME(31)
WRITE (12) -1,1,3700,8540
WRITE (12,202) 125,0,0,125,CONMAX
WRITE (12) -1,1,3700,8335
WRITE (12,201) 125,0,0,125,ALT(J2)
WRITE (12) -1,1,3700,8130
WRITE (12,201) 125,0,0,125,0.
WRITE (12) -1,1,3700,7925
WRITE (12,201) 125,0,0,125,0.
IF(IOPT(1) .EQ. 1) GO TO 4
WRITE (12) -1,1,5625,8980
WRITE (12) 1,1,6125,8980
GO TO 7
4 WRITE (12) -1,1,5025,8980
WRITE (12) 1,1,5525,8980
WRITE (12) -1,1,5725,8950
WRITE (12,203) 125,0,0,125,ISTIM,LAUNTD(4),ISDAY,ISMON,IYEAR
7 WRITE (12) -1,1,5725,8695
WRITE (12,203) 125,0,0,125,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
WRITE (12) -1,1,5725,8490
IF(LTIME)WRITE (12,203) 125,0,0,125,LTIM,LAUNTD(4),LDAY,LMON,LYEAR
RETURN TO CONC
RETURN
END OF LABEL
END
PROGRAM ISOP0,5

******************************************************************************

ISOPLETH PLOTTING PROGRAM -- A SEGMENT OF THE MOD3A PROGRAM

******************************************************************************

A-57
COMMON BLOCK

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSP, E1, CLDHT, 
.IDIRC31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY, 
.ISMON(2), ISYEAR, IV2, JTOP, LAUNTD(10), LTIME, LTIM, LDAY, 
.LMON(2), LYEAR, LU, NUM, PI, PiovR2, PI43, PRESS(31), PTEMP(31), 
.Q1, RADDEG, RATOMC, CLDRAD, R2, R3, SAVEA(30), SAYER(30), SIGA, 
.SIGX0, SIGX, SPEED(31), SQRR2PI, SURDEN, SIGZ0, SIGAP, SB, TEMP(31), 
.TOPSUR, TWDPI, ASPD, VPAR(18), C RTIME(31), DIST, YES, Y1, NUMRUN, 
.YPOS, IFLG1(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIME

INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)), 
(QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)), 
(AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)), 
(HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)), 
(PHCL, VPAR(13)), (PCO, VPAR(14)), (PC02, VPAR(15)), 
(PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

INPUT FORMAT STATEMENTS

100 FORMAT (A1)

OUTPUT FORMAT STATEMENTS

200 FORMAT ("1"20X"CLOUD LOCATION AND DIMENSIONS"/
, " TIME FROM CLOUD STABILIZATION"5X"RANGE"5X"AZIMUTH"
, 8X"DIAMETERS (METERS)"/
, 11X"(MINUTES)"14X"(METERS)"4X"(DEG)"6X"CROSS WIND"
, 4X"ALONG WIND")
201 FORMAT (12XF6.2, 16XF8.1, 4XF5.1, 7XF7.1, 7XF7.1)
202 FORMAT (""&dDEFAULT ISOPLETH CONCENTRATION VALUES?"
, "&dDEFAULT ISOPLETH CONCENTRATION VALUES"
, (NEGATIVE VALUE TERMINATES PROCEDURE): &dJ-)
203 FORMAT (""&dDEFAULT ISOPLETH CONCENTRATION VALUE"
, "<NEGATIVE VALUE TERMINATES PROCEDURE): &dJ-")
204 FORMAT (415, I4 "E"*A2, 2XI2, 1XA2, A1, 1XI4)
205 FORMAT (415, A1)
206 FORMAT (415, F5.2")
207 FORMAT (415", "F5.2")

TYPE AND DIMENSION STATEMENTS

LOGICAL DFALT

DIMENSION CONC(10)

Determine the origin on the map for this plot and move the
pen there

CALL ORGIN(IX0, IY0)
WRITE (12) -1,1,IX0,IY0

DETERMINE THE INDEX IN THE ALTITUDE DATA ARRAY THAT HAS THAT ALTITUDE JUST LOWER THAN THE EFFECTIVE CLOUD HEIGHT, CLDHT

DO 4 I=2,JTOP
IF(CLDHT.GT. ALT(I))GO TO 4
ICLDHT = I - 1
GO TO 5
4 CONTINUE
ICLDHT = JTOP

DRAW THE LINE DEPICTING CLOUD MOVEMENT ALONG THE GROUND AS FAR AS THE CLOUD STABILIZATION POINT

X = 0.0
Y = 0.0
DO 9 I=2,ICLDHT
IM1 = I - 1
RANGE = 0.5 * (CRTIME(I) - CRTIME(IM1)) * (SPEED(I) + SPEED(IM1))
DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1))
IF(IABS(IDIR(I) - IDIR(IM1)).GT. 180)DIR = DIR - 180.0
IF(DIR.LT. 0.0)DIR = DIR + 360.0
DIR = DEGRAD * (360.0 - DIR)
X = X + RANGE * COS(DIR)
Y = Y + RANGE * SIN(DIR)
IX = INT(0.2631 * X) + IX0
IY = INT(0.3545 * Y) + IY0
IF(IX.LT.0 .OR. IX.CT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 11
9 WRITE (12) 1,1,IX, IY

MAKE THE CALCULATIONS NECESSARY TO WRITE OUT THE CLOUD LOCATION AND DIMENSIONS

ALT1 = 0.5 * (CLDHT + ALT(ICLDHT))
ICLDIP1 = ICLDHT + 1
ARG1 = ALT(ICLDIP1) - ALT(ICLDHT)
ARG2 = (CLDHT - ALT(ICLDHT))/ARG1
SPCEN = ALT(ICLDH) + (SPEED(ICLDIP1) - SPEED(ICLDHT)) * ARG2
RANGE = SPCEN * (CRTIME(ICLDIP1) - CRTIME(ICLDHT)) * ARG2
IF(IABS(IDIR(ICLDIP1) - IDIR(ICLDHT)).LT. 180)GO TO 14
IF(IDIR(ICLDIP1).LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
IF(IDIR(ICLDIP1).LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
14 DIR = FLOAT(IDIR(ICLDHT)) + (ALT1 - ALT(ICLDHT)) * (FLOAT(IDIR(ICLDIP1) - IDIR(ICLDHT))/ARG1
IF(DIR.GT. 360.0)DIR = DIR - 360.0
IF(DIR.GT. 180.0)GO TO 17
DIR = DIR + 180.0
GO TO 18
17 DIR = DIR - 180.0
10 DIR = 180.0 - DIR

ARG1 = DEGRAD * DIR
X = X + RANGE * COS(ARG1)
Y = Y + RANGE * SIN(ARG1)
R = SQRT(X * X + Y * Y)
DELR = 300.0 * ASPD

DACRS = 4.30 * SIGX0
DALNG = 4.30 * SIGX0

ARG1 = 180.0
IF(DIR .GT. 180.0) ARG1 = 540.0
AZ = ARG1 - DIR

ARG1 = 180.0
IF(ADR .GT. 180.0) ARG1 = 540.0
DAZ = ARG1 - ADR
ARG1 = DEGRAD * DAZ
DELX = DELR * COS(ARG1)
DELY = DELR * SIN(ARG1)

DELU = ABS(SPEED(ICLDHT) - SPEED(1))

DELT = IDIR(JTOP) - IDIR(1)

T1M = 0.0
R1 = 0.0
KC = K
YC = Y
TXL = 0.28 * DELU/ASPD
SIGX02 = SIGX0 * SIGX0
S82 = S8 * S8
WRITE (6,200)

DO 22 I=1,13
WRITE (6,201) TIM,R,AZ,DACRS,DALNG
T1M = TIM + 5.0
R1 = R1 + DELR
KL = R1 * TXL
SIGX = SQRT((KL/4.30)**2 + SIGX02)
DACRS = 4.30 * SIGX
SIGY = SQRT(S82 + (0.0040589 - 3.0 * DELT * R1)**2)
DALNG = 4.30 * SIGY
KC = KC + DELX
YC = YC + DELY
R = SQRT(KC * KC + YC * YC)
22 AZ = 180.0 - RADDEG * ATAN2(YC,KC)

LABEL THE CLOUD STABILIZATION POINT WITH A +

A-60
IX = INT(0.2631 * X) + Ixo
IY = INT(0.3545 * Y) + IY0
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 77
IXX = IX
IYY = IY
WRITE (12) 1,1,IX,IY
CALL SYMBL(150,150,1H+)

C C LABEL THE POINT OF MAXIMUM CONCENTRATION WITH A O C
C
DIR = DEGRAD * (180.0 - ADIR)
CDIR = COS(DIR)
SDIR = SIN(DIR)
IX1 = INT(0.2631 * (X + RATOMIC * CDIR)) + Ixo
IY1 = INT(0.3545 * (Y + RATOMIC * SDIR)) + IY0
WRITE (12) -1,1,IX1,IY1
CALL SYMBL(150,150,1H+)

C C DRAW THE LINE OF CLOUD MOVEMENT ALONG THE GROUND FROM C
C THE CLOUD STABILIZATION POINT ON C
C
WRITE (12) -1,1,IXX,IYY
RANGE = 1000.0
27 X = X + RANGE * CDIR
Y = Y + RANGE * SDIR
IX = INT(0.2631 * X) + Ixo
IY = INT(0.3545 * Y) + IY0
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 29
WRITE (12) 1,1,IX,IY
GO TO 27
29 WRITE (12) -1,1,IXX,IYY
C C ARE DEFAULT CONCENTRATION VALUES GOING TO BE USED C FOR THE PLOTS C
C
WRITE (LU,202)
READ (LU,100) I
DFALTc = .FALSE.
IF(I .NE. YES)GO TO 35
C C YES -- SET UP THE DEFAULT VALUES C
C
DFALTc = .TRUE.
CONC(1) = 0.1 * CONMAX
CONC(2) = 0.5 * CONMAX
CONC(3) = 0.75 * CONMAX
CONC(4) = - 1.0
C C DO LOOP OVER THE 10 POSSIBLE CONCENTRATION VALUES FOR THE PLOTS C
35 DO 50 I=1,10
C IF DEFAULT CONCENTRATION VALUES ARE NOT BEING USED,
C READ IN THE VALUE FOR THIS PLOT
C
IF(DFALTC)GO TO 37
WRITE (LU,203)
READ (LU,*) CONC(I)
37 IF(CONC(I) .LT. 0.0)GO TO 61
C
ITERATE TO FIND THE LOCATION OF THIS CONCENTRATION
ON THE PLOT
C
DIST = 0.0
DINC = 1000.0
C
41 CALL DFEQP(JTOP,CONC(I))
IF(Y1 .GT. 0.0)GO TO 42
DIST = DIST + DINC
GO TO 41
C
42 IF(DINC .LE. 100.0)GO TO 43
DIST = DIST - 900.0
DINC = 100.0
GO TO 41
C
43 IF(DINC .LE. 10.0)GO TO 44
DIST = DIST - 90.0
DINC = 10.0
GO TO 41
C
PLOT OUT THE CONCENTRATION LINE ON BOTH SIDES
C
44 DIST = DIST - 10.0
IX1 = INT(0.2631 * DIST * CDIR) + IXX
IY1 = INT(0.3545 * DIST * SDIR) + IYY
IF(IX1.LT.0 .OR. IX1.GT.9999 .OR. IY1.LT.0 .OR. IY1.GT.9999)GO TO 58
WRITE (12) -1,1,IX1,IY1
C
DIST = DIST + 10.0
IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 58
WRITE (12) 1,1,IX,IY
C
WRITE (12) -1,1,IX1,IY1
C
IX2 = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
IY2 = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY

A-62
IF(IX2.LT.0 .OR. IX2.GT.9999 .OR. IY2.LT.0 .OR. IY2.GT.9999) GO TO 58

C

46 WRITE (12) 1,1,IX2,IY2
WRITE (12) -1,1,IX,IY

C

IX1 = IX2
IY1 = IY2
DIST = DIST + 500.0
CALL DFEXP(JTOP,CONC(I))
IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 58
WRITE (12) 1,1,IX,IY

C

IF(Y1 .GT. 0.0)GO TO 54
WRITE (12) 1,1,IX2,IY2
GO TO 58

C

54 WRITE (12) -1,1,IX1,IY1
IX2 = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
IY2 = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
IF(IX2.LT.0 .OR. IX2.GT.9999 .OR. IY2.LT.0 .OR. IY2.GT.9999)GO TO 58
GO TO 46

C

58 CONTINUE
C

ON THE PLOT, CROSS OUT EITHER THE WORD FORECAST OR SOUNDING

C

61 IF(IOPT(1) .NE. 0)GO TO 62
WRITE (12) -1,1,707,604
WRITE (12) 1,1,1174,604
GO TO 64

C

62 WRITE (12) -1,1,1269,604
WRITE (12) 1,1,1760,604

C

PRINT OUT THE PREDICTION TIME ON THE PLOT

C

64 WRITE (12) -1,1,1869,319
WRITE (12,204) 100,0,0,150,ITIME,LANTD(4),IDAY,MOMTH,KEYEAR
C

IF THE LAUNCH TIME WAS ENTERED, PRINT IT OUT ON THE PLOT

C

IF(.NOT. LTME)GO TO 67
WRITE (12) -1,1,1869,112
WRITE (12,204) 100,0,0,150,LTIM,LANTD(4),LDAY,LMON,LYEAR

C

ON THE PLOT, PRINT OUT THE CHARACTERS + AND @ FOR THE LEGEND
C
67 WRITE (12) -1,1,1041,1342
WRITE (12,205) 150,0,0,150,1H+
WRITE (12) -1,1,1041,1104
WRITE (12,205) 150,0,0,150,1H0
C
C FOR THE LEGEND ON THE PLOT, PRINT OUT THE CONCENTRATION VALUES
C FOR WHICH CONTOURS WERE DRAWN
C
WRITE (12) -1,1,1066,9587
DO 75 I=1,10
IF(CONC(I) .LT. 0.0)GO TO 77
IF(I .NE. 1)GO TO 72
WRITE (12,206) 125,0,0,150,CONC(I)
GO TO 75
72 WRITE (12,207) 125,0,0,150,CONC(I)
75 CONTINUE
C
RETURN TO THE APPROPRIATE PLACE IN MOD3A
C
77 GO TO IRETRN
78 CALL MOD3A
C
END OF ISOPO
C
END
END$
Program REED
PROGRAM REED

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CDHIT,
   . IDIR(31), I1OPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
   . ISMONC(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
   . LMON(2), LYEAR, LU, NUM, PI, PIOVR2, PI043, PRESS(31), PTEMP(31),
   . SIGCLE, RADDEG, RATOMIC, CLDRAD, R(2), R3, SAVEA(30), SAVER(30), SIGA,
   . SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
   . TOPSUR, TUOPI, ASDP, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
   . YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   . (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   . (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   . (HEAT, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
   . (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
   . (PAL203, VPAR(16)), (PN0, VPAR(17)), (CNMAX, VPAR(18))
DIMENSION ILINE(32), IDATAF(10), IERS(32), JTIM(5)

INPUT FORMAT STATEMENTS

100 FORMAT (I2, 1X, 2A2, I2)
101 FORMAT (A1)
102 FORMAT (10A2)
103 FORMAT (I4, 5XI2, 1XA2, A1, 1XI4)
104 FORMAT (I4, 3XI2, 1XA2, A1, 1XI4)
105 FORMAT (I6, 1XI3, 1XF4.1, F6.1, F6.1, F7.2, 11XF7.2)
106 FORMAT (I4, "C"A2, 1XI2, 1X, A2, A1, 1X, I4)
107 FORMAT (I2, 1X, A2, A1, 1X, I4)
108 FORMAT (F7.2)
109 FORMAT (I2)
110 FORMAT (F4.1)

OUTPUT FORMAT STATEMENTS

200 FORMAT ("!"80(1H+)/1X, 60(1H+)/1X, 10(1H+), 60X, 10(1H+)/
   1X, 10(1H+), "NASA/MSFC MULTILAYER DIFFUSION MODEL -- ",
   "REED"4X", 21 JUN 1977 ", 10(1H+), 1X, 10(1H+), 60X, 10(1H+),

A-66
**TYPE AND DIMENSION STATEMENTS**

```
INTEGER BLANKS,FILE(3),FDIGIT(50),RCHAR,TCHAR,SCHAR,
VNAAMES(4,3),RUNNUM,RA,FO,SDT,TE,ZER00,RCLDR(3)
DIMENSION IPAR(5),VPARS(18,5),IDCB(272),IBUF(40),IALT(31),
DPTEM(31),NAME(3),NAMEF(3)
```

**DATA STATEMENTS**

```
DATA NAME/036522B,2HEE,1HD/ DATA IERS/32*2H /
DATA NAMEF/2H?R,2HEE,1HD/
```

**VPARS (1 THRU 18) = SHUTTLE (19 - 36) = TITAN (37 - 54) = DELTA (55 - 72) = DELTA 3914 (73 - 90) = MINUTEMAN**

```
DATA VPARS/1.521923E7,6.882968E6,3.441484E6,1.894794173E9,  
8.56929516E8,1.713859032E9,6.522129891,4.6800846,  
.375,1479.7,1062.35,1000.0,1970,2234,0.316,2.791,  
.0002,.64,  
5.437528E6,2.718764E6,1.359382E6,3.2625168E8,  
1.6312584E9,3.2625168E8,4.29580469,5.184223,  
5.0,2021.1,1.1010.55,1000.0,1.932,2665,0.0222,
```
DATA BLANKS/2H /, RCHAR/1HR/, TCHAR/1HT/, SCHAR/1HS/, RA/2HRA/, FO/2HF0/, SDT/2HDT/, TE/2HTE/, ZER00/2H00/, NINE9/2H99/, RCLDR/2HRC, 2HLD, 1HR/
DATA FDIGIT/2H01, 2H02, 2H03, 2H04, 2H05, 2H06, 2H07, 2H08, 2H09, 2H10, 2H11, 2H12, 2H13, 2H14, 2H15, 2H16, 2H17, 2H18, 2H19, 2H20, 2H21, 2H22, 2H23, 2H24, 2H25, 2H26, 2H27, 2H28, 2H29, 2H30, 2H31, 2H32, 2H33, 2H34, 2H35, 2H36, 2H37, 2H38, 2H39, 2H40, 2H41, 2H42, 2H43, 2H44, 2H45, 2H46, 2H47, 2H48, 2H49, 2H50/
DATA VNAMES/2HSH, 2HUT, 2HTL, 1HE, 2HTI, 2HTA, 1HN, 1H, 2HDE, 2HLT, 1HA, 1H /

Call GRAF to initialize scope (only applicable if using PLUSASCOPE)

Call GRAF(1)

Find the logical unit number of the device to be used for input and set the variable LU equal to it

Call RMPAR(IPAR)

LU = IPAR(1)

Begin processing of new data by clearing scope

1 Call CLEAR

Initialize some common variables

LTIME = .FALSE.
YES = 1HY
PI = 3.141593
PIOVR2 = 0.5 * PI
PI43 = 1.3333333 * PI
TWOPI = 2.0 * PI
SQR2PI = SQRT(TWOPI)
DEGRAD = PI/180.0
RADDEG = 180.0/PI
DO 2 I=1,3
2 IOPT(I) = 0
JBOT = 0
ZB = 0.0
ZZ = 0.0
REFLEC = 1.0
WRITE THE HEADER OF THE CONSOLE
CALL CLEAR
YPOS = 490.
CALL DREAD(NAMEF, 2, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE, 64, 0, 0)
CALL GETTD(LTIM, LDAY, LMON, LYEAR)
CALL CODE
WRITE (IDATAF, 107) LDAY, LMON, LYEAR
CALL CHAR(368., YPOS, 0, IDATAF, 11, 2, 0)
YPOS = YPOS - 32.
READ IN THE NUMBER OF RUNS TO BE MADE AND THE FIRST FOUR CHARACTERS OF THE DATA FILE NAMES FOR THOSE RUNS
CALL DREAD(NAMEF, 3, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE, 43, 3, 0)
CALL CHAR(384., YPOS, 0, ILINE(25), 8, 3, 0)
CALL CHAR(464., YPOS, 0, ILINE(30), 6, 0, 0)
CALL INC1(JTYPE, 0., 0., 0., 0, 0, 31, 0, 31, IX, IY)
CALL CHAR(0., YPOS, 0, ILINE, 64, 0, 0)
IF (IX .LE. 25) CALL CHAR(464., YPOS, 0, IERS, 6, 0, 0)
IF (IX .GT. 25) CALL CHAR(384., YPOS, 0, IERS, 8, 0, 0)
YPOS = YPOS - 32.
IF (IX .GE. 28) IOPT(2) = 1
IF (IOPT(2) .EQ. 0) GO TO 4
CALL DREAD(NAMEF, 4, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE, 64, 0, 0)
YPOS = YPOS - 16.
CALL DREAD(NAMEF, 5, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE, 22, 3, 0)
NIN = 9
CALL BLANK(IDATAF, 10)
CALL INC(0., JTYPE, 175., YPOS, 0, IDATAF, NIN, 0, 31, 0, 31, IX, IY)
CALL CHAR(0., YPOS, 0, ILINE, 22, 0, 0)
YPOS = YPOS - 32.
CALL CODE

A-69
READ (IDATLF, 100) HUMRUN, FILE(1), FILE(2), IFOFF
HUMRUN = MIN( MAX(HUMRUN, 1), 50)
IF(IFOFF .GT. 0) IFOFF = IFOFF - 1
IF(FILE(1), NE. BLANKS) GO TO 5
4 FILE(1) = 2HDA
FILE(2) = 2HTA
IFOFF = 0
HUMRUN = 1
5 IF(HUMRUN + IFOFF .GT. 50) HUMRUN = 50 - IFOFF
IFOFF = IFOFF + 1
IFLAG(3) = 0
IF(FILE(1).EQ.2HYA .AND. FILE(2).EQ.2HND) IFLAG(3) = 1
IF(FILE(1).EQ.2HTA .AND. FILE(2).EQ.2HPE) IFLAG(3) = 2
IF(FILE(1).EQ.2HDA .AND. FILE(2).EQ.2HTA) IFLAG(3) = 3
IFLAG(4) = '1HE
IF(IFLAG(3).EQ. 1) IFLAG(4) = '1HP

FIND OUT IF THESE RUNS ARE TO BE RESEARCH RUNS (INTERACTION
AND PLOTTING ALLOWED) OR PRODUCTION RUNS

CALL DREAD(NAMEF, 7, ILINE)
CALL LERS(YPOS)
CALL CHAR(0, YPOS, 0, ILINE, 11, 3, 0)
CALL CHAR(128, YPOS, 0, ILINE(9), 12, 3, 0)
CALL CHAR(240, YPOS, 0, ILINE(16), 32, 0, 0)
CALL IN(1, JTYPE, 0, 0, 0, 0, 0, 0, 31, 0, 31, IX, IY)
CALL CHAR(0, YPOS, 0, ILINE, 64, 0, 0)
IF(IX .LT. 12) CALL CHAR(224, YPOS, 0, IERS, 34, 0, 0)
IF(IX .LT. 12) IFLAG(1) = 1
IF(IX.GE.12 .AND. IX.LT.19) CALL CHAR(120, YPOS, 0, IERS, 16, 0, 0)
IF(IX.GE.12 .AND. IX.LT.19) IFLAG(1) = 2
IF(IX.GE.12 .AND. IX.LT.19) CALL CHAR(368, YPOS, 0, IERS, 16, 0, 0)
IF(IX.GE.19) CALL CHAR(120, YPOS, 0, IERS, 30, 0, 0)
IF(IX.GE.19) IFLAG(1) = 3
YPOS = YPOS - 32.
IF(IX.LT.19)ILOPT(2) = 1
IF(IOPT(2).EQ. 0) GO TO 7
GO TO 12
7 CONTINUE

FOR PRODUCTION RUNS, READ IN THE TOP OF THE SURFACE LAYER
AND THE SIGMA OF THE WIND AZIMUTH ANGLE TO BE USED FOR ALL RUNS

CALL DREAD(NAMEF, 11, ILINE)
CALL LERS(YPOS)
CALL CHAR(0, YPOS, 0, ILINE, 33, 3, 0)
MIN = 6
CALL BLANK(IDATAF, 10)
CALL IN(0, JTYPE, 263, YPOS, 0, IDATAF, MIN, 0, 31, 0, 31, IX, IY)
CALL CHAR(0, YPOS, 0, ILINE, 33, 0, 0)
CALL CODE
READ (IDATAF,108) TOPSUR
YPOS = YPOS - 32.
CALL DREAD(NAMEF,12,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,37,3,0)
NIN = 2
CALL BLANK(IDATAF,10)
CALL IN(JTYPE,295.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
CALL CHAR(0.,YPOS,0,ILINE,37,0,0)
CALL CODE
READ (IDATAF,109) ISIGA
IF(NIN.EQ.1) ISIGA = ISIGA/10
SIGA = FLOAT(ISIGA)
YPOS = YPOS - 32.

READ IN AND WRITE OUT THE LAUNCH TIME AND DATE -- IF NOT
ENTERED, DO NOT WRITE ANYTHING OUT

CALL DREAD(NAMEF,8,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,28,3,0)
CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
CALL GETTD(LTIM,LDAY,LMON,LYEAR)
CALL CODE
WRITE (IDATAF,106) LTIM,SDT,LDAY,LMON,LYEAR
CALL CHAR(224.,YPOS,0,IDATAF,20,0,0)
CALL IN(JTYPE,0.,0,0,0,0,31,0,31,IX,IY)
CALL CHAR(0.,YPOS,0,ILINE,28,0,0)
CALL CHAR(384.,YPOS,0,ILINE(25),15,0,0)
IF(IX .LE. 25) CALL CHAR(464.,YPOS,0,IER,6,0,0)
IF(IX .GT. 25) CALL CHAR(384.,YPOS,0,IER,8,0,0)
YPOS = YPOS - 32.
IF(IX .LE. 25) GO TO 17
CALL DREAD(NAMEF,9,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,26,3,0)
NIN = 20
CALL BLANK(IDATAF,10)
CALL IN(JTYPE,207.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
CALL CHAR(0.,YPOS,0,ILINE,26,0,0)
CALL CODE
READ (IDATAF,102) (LAUNTD(I),I=1,10)
YPOS = YPOS - 32.
IF(LAUNTD(1).EQ. BLANKS) GO TO 17
LT = .TRUE.
CALL CODE
READ (LAUNTD,103) LTIM,LDAY,LMON,LYEAR
GO TO 21
17 LAUNTD(4) = SDT

C READ IN THE LAUNCH VEHICLE, LET IT DEFAULT IF NOT ENTERED,
C WRITE IT BACK OUT, AND FILL THE VPAR ARRAY WITH THE
C APPROPRIATE VEHICLE PARAMETERS

21 CALL DREAD(NAMEF,10,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,24,3,0)
CALL CHAR(192.,YPOS,0,ILINE(13),24,0,0)
CALL CHAR(416.,YPOS,0,ILINE(27),11,3,0)
CALL IN(1,JTYPE,0.,0.,0,0,0,31,0,31,IX,IY)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
IF(IX .LE. 15)CALL CHAR(312.,YPOS,0,IERs,24,0,0)
IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(192.,YPOS,0,IERs,12,0,0)
IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(416.,YPOS,0,IERs,11,0,0)
IF(IX .GE. 24)CALL CHAR(192.,YPOS,0,IERs,24,0,0)
YPOS = YPOS - 32.
IF(IX .LE. 15)GO TO 25
IF(IX .LE. 23)GO TO 24

C************************************************************************
C*** IF IOPT(3) = 0 THEN IT IS A SHUTTLE LAUNCH. ********
C************************************************************************
C*** IF IOPT(3) = 1 THEN IT IS A TITAN LAUNCH. ********
C************************************************************************
C*** IF IOPT(3) = 2 THEN IT IS A DELTA LAUNCH. ********
C************************************************************************

25 IOPT(3) = 2
26 I = IOPT(3) + 1

C FILL THE VPAR ARRAY

DO 28 J=1,18
28 VPAR(J) = VPARS(J,I)

C CHANGE IN BOTTOM LAYER WITH TOTAL REFLECTION?

CALL DREAD(NAMEF,15,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
YPOS = YPOS - 32.
CALL DREAD(NAMEF,16,ILINE)
CALL LERS(YPOS)
CALL CHAR(32.,YPOS,0,ILINE(3),11,3,0)
CALL CHAR(160., YPOS, 0, ILINE(11), 43, 0, 0)
CALL IN(1, JTYPE, 0., 0., 0., 0., 0., 31, 0, 31, IX, IY)
CALL CHAR(0., YPOS, 0, IERS, 64, 0, 0)
YP = YPOS
YPOS = YPOS - 32.0

CHECK FOR SURFACE -- STABILIZATION -- SOMETHING ELSE

IF(IX .GE. 9) GO TO 29
IFLAG(2) = 0
CALL CHAR(0., 0, YP, 0, ILINE, 16, 0, 0)
GO TO 38
29 IF(IX .GE. 20) GO TO 30
IFLAG(2) = 1
CALL CHAR(160., 0, YP, 0, ILINE(11), 16, 0, 0)
JBOT = 2
ZB = ALT(JBOT)
GO TO 38

DEFAULT HEIGHT CALCULATION Zt?

30 IFLAG(2) = 2
CALL CHAR(320., YP, 0, ILINE(20), 18, 0, 0)
CALL DREAD(NAMEF, 17, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE, 42, 3, 0)
CALL CHAR(384., YPOS, 0, ILINE(25), 8, 3, 0)
CALL CHAR(464., YPOS, 0, ILINE(30), 6, 0, 0)
CALL IN(1, JTYPE, 0., 0., 0., 0., 0., 31, 0, 31, IX, IY)
CALL CHAR(0., YPOS, 0, IERS, 42, 0, 0)
CALL CHAR(0., YPOS, 0, ILINE, 42, 0, 0)
IF(IX .LE. 25) GO TO 37
CALL CHAR(384., YPOS, 0, IERS, 8, 0, 0)
YPOS = YPOS - 32.

ENTER HEIGHT Z\(z\)

CALL DREAD(NAMEF, 18, ILINE)
CALL LERS(YPOS)
CALL CHAR(47., YPOS, 0, ILINE(4), 10, 3, 0)
NIN = 6
CALL BLANK(IDATAF, 10)
CALL IN(0, JTYPE, 128., YPOS, 0, IDATAF, NIN, 0, 31, 0, 31, IX, IY)
CALL CODE
READ (IDATAF,*), ZZ
CALL CHAR(0., YPOS, 0, IERS, 16, 0, 0)
CALL CHAR(47., YPOS, 0, ILINE(4), 10, 0, 0)
YPOS = YPOS - 32.

ENTER SURFACE REFLECTION?
CALL DREAD(NAMEF, 19, ILINE)
CALL LERS(YPOS)
CALL CHAR(31, YPOS, 0, ILINE, 45, 3, 0)
CALL CHAR(384, YPOS, 0, ILINE(25), 8, 3, 0)
CALL CHAR(472, YPOS, 0, ILINE(30), 6, 0, 0)
CALL INC(1, JTYPE, 0, 0, 0, 0, 0, 31, 0, 31, IX, IY)
CALL CHAR(0, YPOS, 0, ILINE, 64, 0, 0)
IF(IX .LE. 25) REFLEC = 1.0
IF(IX .LE. 25) CALL CHAR(464, YPOS, 0, IERS, 6, 0, 0)
IF(IX .LE. 25) CALL CHAR(384, YPOS, 0, IERS, 0, 0, 0)
YPOS = YPOS - 32.

WRITE OUT RF VALUES FOR SELECTION

CALL DREAD(NAMEF, 20, ILINE)
CALL LERS(YPOS)
CALL CHAR(0, YPOS, 0, ILINE, 64, 3, 0)
MIN = 4
CALL BLANK(IDATAF, 10)
CALL INC(2, JTYPE, 440, YPOS, 0, IDATAF, MIN, 0, 31, 0, 31, IX, IY)
IF(JTYPE .NE. 0) GO TO 31
CALL CODE
READ (IDATAF, *) REFLEC
GO TO 32
31 IX = IX/2
IF(IX .EQ. 1) REFLEC = 0.8
IF(IX .EQ. 3) REFLEC = 0.7
IF(IX .EQ. 5) REFLEC = 0.5
IF(IX .EQ. 7) REFLEC = 0.3
IF(IX .EQ. 9) REFLEC = 0.1
IF(IX .EQ. 11) REFLEC = 0.0
CALL CODE
WRITE (IDATAF, 110) REFLEC
GO TO 32
32 CALL CHAR(0, YPOS, 0, IERS, 64, 0, 0)
CALL CHAR(48, YPOS, 0, ILINE, 6, 0, 0)
CALL CHAR(464, YPOS, 0, IDATAF, 4, 0, 0)
IF(JTYPE .NE. 0) GO TO 34
CALL CODE
READ (IDATAF, *) REFLEC
34 YPOS = YPOS - 32.

DEFAULT HEIGHT OF BASE LAYER?

CALL DREAD(NAMEF, 21, ILINE)
CALL LERS(YPOS)
CALL CHAR(0, YPOS, 0, ILINE, 46, 3, 0)
CALL CHAR(384, YPOS, 0, ILINE(25), 8, 3, 0)
CALL CHAR(464, YPOS, 0, ILINE(30), 6, 0, 0)
CALL IN(1,JTYPE,0.,0.,0.,0.,0.,0.,31.,0.,31.,IX,IFY)
CALL CHAR(0.,YP0S,0,ILINE,64.,0.,0.)
IF(IX .LE. 25) CALL CHAR(464.,YP0S,0,IIERS,6.,0.,0.)
IF(IX .LE. 25) GO TO 36
CALL CHAR(384.,YP0S,0,IIERS,8.,0.,0.)
YP0S = YP0S - 32.
CALL DREAD(NAMEF,22,ILINE)
CALL IIERS (YP0S)
CALL CHAR(47.,YP0S,0,ILINE(4),10.,3.,0.)

ENTER HEIGHT OF BASE LAYER

NIN = 6
CALL BLANK(IDATAF,10)
CALL INC(0,JTYPE,144.,YP0S,0,IDATAF,NIN,0.,31.,0.,31.,IX,IFY)
CALL CODE
READ (IDATAF,*) ZB
CALL CHAR(47.,YP0S,0,ILINE(4),10.,0.,0.)
YP0S = YP0S - 32.
GO TO 38
36 ZB = 0.0
GO TO 38
37 CALL CHAR(0.,YP0S,0,IIERS,64.,0.,0.)
CALL CHAR(0.,YP0S,0,ILINE,58.,0.,0.)
YP0S = YP0S - 32.
38 CONTINUE

DO LOOP ON THE RUN NUMBER

DO 79 RUNNUM=1,NUMRUN

SET UP THE FILE NAME FOR THIS RUN, GET THE CURRENT TIME,
AND WRITE OUT THE HEADER

FILE(3) = FDIGIT(RUNNUM+IFOFF)
CALL GETTD(CTIME,IDAY,MONTH,IFYEAR)
I = IOPT(3) + 1
WRITE (6,200) RUNNUM,(FILE(J),J=1,3),(YNAK(ME,J,I),J=1,4)
IF(LTIME) WRITE (6,201) LTIME,LAUNTD(3),LAUNTD(4),LDAY,LMDN(1),
LMON(2),LYEAR
WRITE (6,202) CTIME,LAUNTD(4),IDAY,MONTH,IFYEAR

IF THE DATA IS ON A DISK FILE, READ FROM DISK -- IF IT
IS ON TAPE, READ IT AS KSC 1965 DATA IN SUBROUTINE KSC65

IF(IFLAG(3) .NE. 2) GO TO 39
CALL KSC65(IBUF,IALT,DPTEMP,IFOFF1,IEOF)
IFOFF1 = 1
IF(IEOF .EQ. 1) GO TO 81
IF(IEOF .EQ. 2) GO TO 79

A-75
GO TO 48

OPEN THE DATA FILE FOR THIS RUN

39 CALL OPEN(IDCB, IERR, FILE, 0, 0, 0, 272)
   IF(IERR .GE. 0) GO TO 40
   WRITE (6, 203) IERR
   GO TO 79

READ THE HEADINGS FROM THE DATA FILE, SETTING UP THE
APPROPRIATE PARAMETERS

40 CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .GE. 0) GO TO 42
   WRITE (6, 204) IERR
   GO TO 79
41 IF(IBUF(1) .NE. TE) GO TO 40
42 WRITE (6, 205) (IBUF(I), I=1, LEN)
   CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .LT. 0) GO TO 41
   IF(IBUF(1) .NE. RA .AND. IBUF(1) .NE. FO) GO TO 43
   IOPT(1) = 0
   IF(IBUF(1) .EQ. FO) IOPT(1) = 1
   WRITE (6, 205) (IBUF(I), I=1, LEN)
   CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .LT. 0) GO TO 41
   WRITE (6, 205) (IBUF(I), I=1, LEN)

READ THE SOUNDING/FORECAST TIME

CALL READF(IDCB, IERR, IBUF, 9)
   IF(IERR .LT. 0) GO TO 41
   CALL CODE
   READ (IBUF, 104) ISTIM, ISDAY, ISMON(1), ISMON(2), ISYEAR

CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME

ISTIM = ISTIM - 500
   IF(IFLAG(3) .EQ. 1) ISTIM = ISTIM - 300
   IF(LAUNTD(4) .NE. 2HST) ISTIM = ISTIM + 100
   IF(ISTIM .GT. 0) GO TO 44
   ISTIM = 2400 + ISTIM
   ISDAY = ISDAY - 1

WRITE OUT THE NEXT LINE OF THE HEADER

44 CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .LT. 0) GO TO 41
   WRITE (6, 205) (IBUF(I), I=1, LEN)
WRITE OUT THE SOUNDING/FORECAST TIME
WRITE (6, 206) ISTD, IFLAG(4), LAUNTD(4), ISDAY, ISMON(1), ISMON(2), ISYEAR

FIND THE FIRST DATA POINT WITH AN ALTITUDE OF 10 FEET OR ABOVE

45 CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .LT. 0 ) GO TO 41
   CALL B2Z(IBUF(1), J)
   IF(J .LT. ZERO0 .OR. J.GT.NINE9) GO TO 45
   CALL CODE
   READ (IBUF, 105) IALT(1), IDIR(1), SPEED(1), TEMP(1), DPTEMP(1),
     PRESS(1), SURDEN
   IF(IALT(1) .LT. 10 ) GO TO 45

TRY TO FIND A TOTAL OF 30 DATA POINTS WITH ALTITUDES BETWEEN 20 FT AND 10,000 FT INCLUSIVE

NUM = 1
DO 47 I = 2, 30
46 CALL READF(IDCB, IERR, IBUF, 40, LEN)
   IF(IERR .LT. 0 .AND. IERR .NE. -12) GO TO 41
   IF(LEN .EQ. -1) GO TO 48
   CALL B2Z(IBUF(1), J)
   IF(J .LT. ZERO0 .OR. J.GT.NINE9) GO TO 46
   CALL CODE
   READ (IBUF, 105) IALT(I), IDIR(I), SPEED(I), TEMP(I), DPTEMP(I),
     PRESS(I)
   IF(IALT(I) .LT. 20 .OR. IALT(I) .GT. 10000) GO TO 46
47 NUM = I
C
ZERO OUT THE REMAINING ELEMENTS OF THE ARRAYS
C
48 NUM1 = NUM + 1
   IF(NUM1 .GT. 30) GO TO 51
   DO 49 I = NUM1, 30
      ALT(I) = 0.0
      IDIR(I) = 0
      SPEED(I) = 0.0
      TEMP(I) = 0.0
      DPTEMP(I) = 0.0
      PRESS(I) = 0.0
49 C
C
CONVERT TO METRIC UNITS
C
51 DO 52 I = 1, NUM
      ALT(I) = 0.3048 * FLOAT(IALT(I))
52 SPEED(I) = 0.515 * SPEED(I)

A-77
SORT ALL THE DATA POINTS SO THEY APPEAR IN ASCENDING ORDER OF ALTITUDE

NUM1 = NUM - 1
DO 58 I=1,NUM1
JJ = NUM - I
DO 57 J=1, JJ
J1 = J + 1
IF(ALT(J) .LE. ALT(J1)) GO TO 57
ARG = ALT(J)
ALT(J1) = ALT(J)
ALT(J) = ARG
IARG = IDIR(J)
IDIR(J1) = IDIR(J1)
IDIR(J1) = IARG
ARC = SPEED(J)
SPEED(J1) = SPEED(J)
SPEED(J) = ARC
ARG = TEMP(J)
TEMP(J1) = TEMP(J)
TEMP(J) = ARG
ARC = DPTEMP(J)
DPTEMP(J1) = DPTEMP(J)
DPTEMP(J) = ARC
ARG = PRESS(J)
PRESS(J1) = PRESS(J)
PRESS(J) = ARG
57 CONTINUE
58 CONTINUE

CALCULATE THE POTENTIAL TEMPERATURE

DO 62 I=1,NUM
62 PTEMP(I) = (TEMP(I) + 273.15) * ((1000.0/PRESS(I))**0.288)

WRITE THE HEADER FOR SOUNDING OR FORECAST

IF(IOPT(1) .EQ. 1) GO TO 64
WRITE (6,207)
GO TO 65
64 WRITE (6,208)

WRITE THE SURFACE DENSITY AND ALL THE DATA POINTS

65 WRITE (6,209) SURDEN
WRITE (6,210)
DO 68 I=1, NUM
IALTF = 3.281 * ALT(I) + 0.5
IALTM = ALT(I) + 0.5

A-78
AP TEMP = P TEMP(I) - 273.15
68 WRITE (6, 211) I, I ALTF, I ALTM, IDIR(I), SPEED(I), TEMP(I), 
       A TEMP, D TEMP(I), PRESS(I)

C

CLOSE THE DATA FILE

CALL CLOSE(IDCB)

C

TRANSFER TO THE PROGRAM RCLDR -- THE CLOUD RISE PROGRAM

CALL NGRAF
CALL RWDIS(NAME, 1)
CALL EXEC(9, RCLDR)
CALL RWDIS(NAME, 0)
CALL GRAF(1)

C

PROCESS THE NEXT RUN

79 CONTINUE

C

TERMINATE OF PROCESS MORE DATA?

81 CALL DREAD(NAMEF, 14, ILINE)
   CALL LERS(YPOS)
   CALL CHAR(0., YPOS, 0, ILINE(1), 24, 3, 0)
   CALL CHAR(224., YPOS, 0, ILINE(15), 14, 3, 0)
   CALL CHAR(352., YPOS, 0, ILINE(23), 12, 0, 0)
   CALL IN(1, JTYPE, 0, .0, 0, 0, 0, 31, 0, 31, IX, IY)
   IF(IX .LT. 20) GO TO 82

C

PROCESS MORE DATA

CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0, ILINE(1), 28, 0, 0)
CALL CHAR(352., YPOS, 0, ILINE(23), 12, 0, 0)
YPOS = 458.
GO TO 1

C

TERMINATE EXECUTION

82 CALL DREAD(NAMEF, 13, ILINE)
   CALL LERS(YPOS)
   YPOS = YPOS - 32.
   CALL CHAR(0., YPOS, 0, ILINE, 64, 3, 0)

C

DELAY BEFORE CLEARING SCOPE

CALL EXEC(11, JJTIM)
JJ = JJTIM(2)
IF(JJ .GT. 55) JJ = 5

A-79
'85 CALL EXEC(11, JJTIM)
  IF(JJ+5 .GT. JJTIM(2)) GO TO 85

C REINITIALIZE SCOPE NORMAL OPERATION AND STOP
C
CALL NGRAF
STOP

END

SUBROUTINE KSC65(IBUF, IALT, DPTEMP, IWANT, IEOF)

******************************************************************************

C THIS SUBROUTINE READS IN DATA FOR THE REED DIFFUSION
C MODEL FROM MAG TAPE IN KSC 1965 FORMAT

******************************************************************************

C DATA READING

COMMON BTCM, AL1, CO, COGMAX, CONCPK, DEGRAD, ADIR, DOP PK, E1, CLDHT,
   IDIR(31), IOLT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
   ISMON(2), ISYER, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
   LMON(2), LYEAR, LU, NUM, PI, PI0VR2, PI43, PRESH(31), PTEMP(31),
   SIGHCL, RADDEG, RATMC, CLDRAD, R2, R3, SAVEA(30), SAVER(30), SIGA,
   SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIG20, SIGAP, SB, SEMP(31),
   TOPSUR, T00PI, ASPD, VPAR(18), CTIME(31), DIST, YES, Y1, NUMRUN,
   VPOS, IFLAG(5), ZB, Z, REFLEC, IRETRN

LOGICAL LTIM
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATF, VPAR(12)),
   (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCD, VPAR(15)),
   (PAL02, VPAR(16)), (PND, VPAR(17)), (GAMMA, VPAR(18))

C INPUT FORMAT STATEMENTS

1000 FORMAT (40A2)
1001 FORMAT (I4, 3X12, 1XA2, A1, 1XI4)
1002 FORMAT (1X16, 3X13, 5XF3.0, 2XF5.1, 3XF5.1, 3XF6.1, 15XF6.1)

C OUTPUT FORMAT STATEMENT

2000 FORMAT ("O"5X"TIME: ": I4, 1XA1, A2, 4X"DATE: ": I2, 1XA2, A1, 1XI4)

C DIMENSION STATEMENT

A-80
DIMENSION IBUF(1), IALT(1), DPTEMP(1).

INITIALIZE THE COUNTER FOR THE NUMBER OF SETS OF DATA TO 0
IGOT = 0

READ DATA FROM TAPE

4 READ (8,1000) (IBUF(I), I=1,40)

IF AN EOF ON TAPE, SET THE EOF FLAG AND RETURN
CALL EXEC(13,8,IEQTS)
IEOF = IAND(ISHIF(IEQTS,-7),1)
IF(IEOF .EQ. 1) RETURN

KEEP READING UNTIL THE STANDARD LEVEL DATA IS FOUND

IF(IBUF(2) .NE. 2HST) GO TO 4
7 READ (8,1000) (IBUF(I), I=1,40)
IF(IBUF(1) .NE. 2HST AND IBUF(2) .EQ. 2HST) GO TO 7

READ THE SOUNDING/FORECAST TIME
READ (8,1001) ISTIM, ISDAY, ISMON(1), ISMON(2), ISYEAR

CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
ISTIM = ISTIM - 500
IF(IFLAG(3) .EQ. 1) ISTIM = ISTIM - 300
IF(LAUNCH(4) .NE. 2HST) ISTIM = ISTIM + 100
IF(ISTIM .GT. 2400) GO TO 11
ISTIM = 2400 + ISTIM
ISDAY = ISDAY - 1

FIND THE KEY WORD ALTITUDE

11 READ (8,1000) (IBUF(I), I=1,40)
IF(IBUF(2) .EQ. 2HST) GO TO 7
IF(IBUF(1) .NE. 2HAL) GO TO 11

LIMIT DATA TO 30 POINTS -- READ THE STANDARD LEVEL DATA

DO 19 I=1,30
15 READ (8,1002) IALT(I), IDIR(I), SPEED(I), TEMP(I), DPTEMP(I), PRESS(I), SURDH
IF(SPEED(I) .EQ. 999.0 OR IDIR(I) .EQ. 999) GO TO 15
IF(IDIR(I) .EQ. 360) IDIR(I) = 0
IF(I .EQ. 1) SURDEH = SURDH
IF(IALT(1) . GT. 10000) GO TO 22
19 CONTINUE
22 NUM = 1
IF(NUM . GT. 30) GO TO 34
C C
FIND THE KEY WORD MANDATORY
C
25 READ (8,1000) (IBUF(1),I=1,40)
   IF(IBUF(2) . EQ. 2HST) GO TO 7
   IF(IBUF(10) . NE. 2HOR . AND. IBUF(15) . NE. 2HOR) GO TO 25
   READ (8,1000) I
C C
LIMIT DATA TO 30 POINTS -- READ THE MANDATORY LEVEL DATA
C
DO 29 I=NUM,30
27 READ (8,1002) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),PRESS(I)
   IF(SPEED(I) . EQ. 999.0 . OR. IDIR(I) . EQ. 999) GO TO 27
   IF(IDIR(I) . EQ. 360) IDIR(I) = 0
   IF(IALT(I) . GT. 10000) GO TO 34
29 CONTINUE
C C
NUM IS THE NUMBER OF DATA POINTS
C
34 NUM = I - 1
C C
INCREMENT THE COUNTER -- IF THIS IS THE SET OF DATA DESIRED,
WRITE OUT THE SOUNDING/FORECAST TIME -- OTHERWISE GET THE NEXT
SET
C
IGOT = IGOT + 1
IF(IGOT . LT. IWANT) GO TO 4
C C
WRITE OUT THE SOUNDING/FORECAST TIME
C
WRITE (6,2000) ISTIM,IFLAG(4),LAUNTD(4),ISDAY,ISMON(1),ISMON(2),
   ISYEAR
C C
THERE MUST BE 5 OR MORE DATA POINTS FOR THIS TO BE A VALID SET
OF DATA -- IF THERE IS NOT, RETURN WITH IEOF=2
C C
IF(NUM .GE. 5) RETURN
IEOF = 2
RETURN
C C
END OF KSC65
C C
END
SUBROUTINE RWDIS(NAME, JJ)
COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPKE1,CLDHT,
   IDIR(31),IOPT(3),ITIME,IDAY,MNTH(2),IYEAR,ISTIM,ISDAY,
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
(QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
(AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
(HEAT, VPAR(10)), (HEATM, VPAR(11)), (HEAT, VPAR(12)),
(PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
(PAL203, VPAR(16)), (PM0, VPAR(17)), (GAMMA, VPAR(18))

INTEGER ODCB(144), OBUF(669)
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1), ALT(1))
CALL OPEN(ODCB, IERR, NAME, 0)
IF(JJ.EQ.1)CALL WRITF(ODCB, IERR, OBUF, 669)
IF(JJ.EQ.0)CALL READF(ODCB, IERR, OBUF, 669)
CALL CLOSE(ODCB, IERR)
RETURN

SUBROUTINE GETTD(ITIME, IDAY, MONTH, IYEAR)

C******************************************************************************
C
C THIS SUBROUTINE RETURNS THE CURRENT TIME, DAY, MONTH, AND YEAR
C******************************************************************************
C
C TYPE AND DIMENSION STATEMENTS

INTEGER DAYMON(12)
DIMENSION MONTH(2), MONTHS(2, 12), IT(5)

DATA STATEMENTS

DATA MONTHS/2HJA, 1HN, 2HFE, 1HB, 2HMA, 1HR, 2HAP, 1HR,
2HMA, 1HY, 2HJU, 1HN, 2HJU, 1HL, 2HAU, 1HG,
2HSE, 1HP, 2HOC, 1HT, 2HNO, 1HY, 2HDE, 1HC/
DATA DAYMON/31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /

CALL EXEC TO RETURN CURRENT TIME, JULIAN DAY, AND YEAR

CALL EXEC(11, IT, IYEAR)

USE JUST HOURS AND MINUTES FOR THE TIME

A-83
ITIME = 100 * IT(4) + IT(3)

MAKE APPROPRIATE ADJUSTMENTS IF THIS IS A LEAP YEAR

DAYMON(2) = 28
I = IYEAR/4
IF(4*I .EQ. IYEAR)DAYMON(2) = 29

CONVERT THE JULIAN DAY INTO A MONTH AND A DAY

IDAY = IT(5)
DO 7 I=1,12
   IDAY = IDAY - DAYMON(I)
   IF(IDAY .LE. 0) GO TO 12
7 CONTINUE

IDAY = IDAY + DAYMON(I)
MONTH(1) = MONTHS(1,I)
MONTH(2) = MONTHS(2,I)

RETURN TO THE CALLING PROGRAM

RETURN

END

SUBROUTINE B2Z(IA,IB)
IB = IAND(IA,177400B)
IF(IB .EQ. 020000B)IB = 030000B
IC = IAND(IA,000377B)
IF(IC .EQ. 000040B)IC = 000060B
IB = IOR(IB,IC)
RETURN

END

SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32),IPAR(5)
CALL RMPAR(IPAR)
LU = IPAR(1)
CALL OPEN(IDCBIIPAR,IBF,NAMEF,0)
LOOP = LNUM - 1
DO 10 I=1,LOOP
   CALL BLANK(IBUF,40)
   CALL READF(IDCBIIPAR,IBF)
10 CONTINUE
   CALL BLANK(IBUF,40)
   CALL READF(IDCBIIPAR,IBF)
   CALL CODE
   READ(IBUF,100) (ILINE(I),I=1,32)
100 FORMAT(32A2)
   CALL CLOSE(IDCBIIPAR)
RETURN
END
SUBROUTINE BLANK(IBUF,II)
DIMENSION IBUF(40)
DATA IBLK/2H /
DO 10 I=1,II
    IBUF(I) = IBLK
RETURN
END
SUBROUTINE IERS(YPOS)
DIMENSION IERS(32)
DATA IERS/32*2H /
IF(YPOS.LE.48) YPOS = 458.0
CALL CHAR0.,YPOS,0,IERS,64,0,0)
CALL CHAR0.,YPOS-16.,0,IERS,64,0,0)
RETURN
END
END$
**FTN4.L**

**PROGRAM RCLDR**

*COMMON BLOCK*

```plaintext
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT, 
   IDIR(31), IOTP(3), LATIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY, 
   ISMON(2), ISYEAR, IV2, JTOP, JBOT, LAUHTD(10), LTIME, LTIM, LDAY, 
   LMON(2), LYEAR, LU, NUM, PI, PI0YR2, PI43, PRESS(31), PTEMP(31), 
   SIGC, RADDEC, RATONC, CLDRAD, R2, R3, SAVEA (30), SAYER(30), SIGA, 
   SIGX, SIGY, SPEED(31), SQR2PI, SURDEN, SIGZ, SIGAP, S8, TEMP(31), 
   TOPSUR, TWOP1, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN, 
   YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETBN 
LOGICAL LTIME 
INTEGER YES 
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)), 
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)), 
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)), 
   (HEATM, VPAR(10)), (HEATM, VPAR(11)), (HEATM, VPAR(12)), 
   (PHCL, VPAR(13)), (PH0, VPAR(14)), (PC02, VPAR(15)), 
   (PAL203, VPAR(16)), (PHO, VPAR(17)), (GAMMAX, VPAR(18))

INTEGER RMETP(3) 
DIMENSION *IAS(31), NAME (3), NAMEF(3), ILINE(32), IDATAF(10), 
   IERS(80), ISURFP(3)

**OUTPUT FORMAT STATEMENTS**

```plaintext
100 FORMAT (F7.2) 
101 FORMAT (I2) 
102 FORMAT (F3.1) 
200 FORMAT ("1*27X"EXHAUST CLOUD"/"LEVEL"4X"ALTITUDE"6X"RISE TIME"5X"RANGE"6X"DIRECTION"/10X"(METERS)"17X 
   "SECONDS")4X"(METERS)"4X"(DEGREES)"
201 FORMAT (2X13.5F7.1, 5X"ADIABATIC"5XF6.1, 6XF7.1, 7XF5.1)
202 FORMAT (2X13.5F7.1, 6X"STABLE"7XF6.1, 6XF7.1, 7XF5.1)
203 FORMAT ("/*/O***CLOUD STABILIZATION***"/
   6X"HEIGHT(M)" "F6.1/ 
   6X"STABILIZATION TIME AFTER LAUNCH(SEC)" "F5.1/
   6X"RANGE FROM PAD(M)" "F7.1/
   6X"DIRECTION FROM PAD(DEG)" "F5.1")
204 FORMAT (F6.1)

A-86
205 FORMAT (//"O****TOP OF SURFACE LAYER METEOROLOGICAL PARAMETERS"
   ***/
     6X"HEIGHT(M): "F6.1/
     6X"WIND DIRECTION(DEG): "I3/
     6X"WIND SPEED(M/SEC): "F4.1)
206 FORMAT (//"O****DIFFUSION PARAMETERS****"/
     6X"MEAN SPEED(M/SEC): "F4.1/
     6X"MEAN TRANSPORT DIRECTION(DEG): "F5.1)
207 FORMAT (F3.0)
208 FORMAT (//"O_SIGMA OF WIND AZIMUTH ANGLE, SIGMA: "F4.1)
209 FORMAT (//"O_EFFECTIVE CLOUD HEIGHT(M): "F6.1)
C
C TYPE AND DIMENSION STATEMENTS
C
INTEGER RCONC(3)
C DATA STATEMENT
C
DATA NAME/036522B,2HEE,1HD/,NAMEF/2HR,2HCL,2HDR/
DATA RMETP/PHRM,2HET,1HP/
DATA RCONC/2HRC,2HON,1HC/
DO 1 I=1,80
1 IERS(I) = 2H
C
** CALL GRACF1) TO INITIALIZE PLASMASCOPE GRAPHIC MODE
CALL GRACF1)
** INITIALIZE THE Y POSITION OF THE CALL CHARACTER STATEMENTS
C ** ON THE PLASMASCOPE.
C YPOS=490.
C *** READ COMMON DATA FILE ***
C CALL RWDIS(NAME,0)
C
C INITIALIZE SOME LOCAL VARIABLES
C
CRTIME( ) - CLOUD RISE TIME
IAS( ) - 0 = ADIABATIC
       1 = STABLE
ALTINC - ALTITUDE INCREMENT
ITERAT - ITERATION COUNTER
C
RNGY = 0.0
RNGX = 0.0
CRTIME(1) = 0.0
ALTINC = 0.0
SAVER(1) = 0.0
SAVEA(1) = 0.0
ITERAT = 0
C
WRITE OUT THE EXHAUST CLOUD HEADER
C
A-87
WRITE (6,200)

CALCULATE SOME QUANTITIES TO BE USED IN SUBSEQUENT DO LOOP

ALPHAC = 5.12913006E-02*(TEMP(1) + 273.15)*SURDEN*GAMMAX**3

ALPHAC = ALPHAC/(HEATN * QC1)

GT = 9.8/(TEMP(1) + 273.15)

DO LOOP TO CALCULATE EXHAUST CLOUD PARAMETERS

DO 9 I=2,NUM

IM1 = I - 1
IAS(I) = 1

CALCULATE SLOPE OF POTENTIAL TEMPERATURE, SPEED, AND DIRECTION IN LAYER

DALT = ALT(I) - ALT(IM1)

GPTEMP = (PTEMP(I) - PTEMP(IM1))/DALT

GSPEED = (SPEED(I) - SPEED(IM1))/DALT

GDIR = FLOAT(IDIR(I) - IDIR(IM1))/DALT

CALCULATE METEOROLOGICAL AND ENERGY FACTOR

2 Z = ALT(I) - ALT(I) - ALTINC

ALPHA = ALPHAC * Z**4/(AA * Z**BB + CC)

CALCULATE POTENTIAL TEMPERATURE FACTOR

STAB = GT * (PTEMP(I) - ALTINC * GPTEMP - PTEMP(I))/

(ALT(I) - ALTINC - ALT(1) + 1.0E-7)

CALCULATION FOR ADIABATIC RISE

IF(STAB .GT. 0.000001)GO TO 4

CRTIME(I) = SQRT(ALPHA)

IAS(I) = 0

GO TO 6

CALCULATION FOR STABLE CLOUD RISE

C2 - ARGUMENT OF ARC COSINE (MUST BE LESS THAN -1)

4 C2 = 1.0 - 0.5 * ALPHA * STAB

IF(C2 .LT. -1.0)GO TO 5

C3 = C2/SQRT(1.0 - C2 * C2)

CRTIME(I+ITERAT) = (PI0VR2 - ATAN(C3))/SQRT(STAB)

IF(ITERAT .EQ. 1)GO TO 11

GO TO 6

A-88
ITERATE IN LAYER

5 ALTINC = ALTINC + 5.0
ITERAT = 1
GO TO 2

CALCULATE RANGE AND DIRECTION

6 DELRNG = -0.5 * (SPEED(IM1) + SPEED(I)) * (CRTIME(IM1) - CRTIME(I))
DELDIR = 0.00872665 * FLOAT(IDIR(I) + IDIR(IM1))
RNGY = RNGY - DELRNG * SIN(DELDIR)
RNGX = RNGX - DELRNG * COS(DELDIR)
AZMUTH = RADDEG * ATAN2(RNGY, RNGX)
IF(AZMUTH.LT.0.0)AZMUTH = AZMUTH + 360.0
SAYSAV(1) = DELRNG
SAYSEA(1) = AZMUTH

WRITE OUT THE VARIABLES WITH THE APPROPRIATE FORMAT STATEMENT
BASED OF WHETHER OR NOT CLOUD IS ADIABATIC OR STABLE

IF(IAS(I).NE.0)GO TO 8
WRITE (6,201) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
GO TO 9
8 WRITE (6,202) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
9 CONTINUE

CALCULATE AND WRITE OUT STABILIZATION HEIGHT AND TIME

11 DELRNG = 0.5 * (SPEED(IM1) - ALTINC * GSPEED + SPEED(I)) * (CRTIME(I + 1) - CRTIME(IM1))
DALT = 0.00872665 * (FLOAT(IDIR(I) + IDIR(IM1)) - GDIR * ALTINC)
RNGY = RNGY - DELRNG * SIN(DALT)
RNGX = RNGX - DELRNG * COS(DALT)
AZMUTH = RADDEG * ATAN2(RNGY, RNGX)
IF(AZMUTH.LT.0.0)AZMUTH = AZMUTH + 360.0
DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
ALT(31) = ALT(I) - ALTINC
WRITE (6,203) ALT(31),CRTIME(I+1),DELRNG,AZMUTH

STORE THE INDEX OF THE ESTIMATED TOP OF THE SURFACE LAYER

JTOP = I

LOAD THE CLOUD RISE TIME ARRAY

CRTIME(31) = CRTIME(JTOP)
DO 15 J=I,NUM
15 CRTIME(I) = CRTIME(31)

IS THIS A RESEARCH OR A PRODUCTION RUN?

IF(IOPT(2) .NE. 0)GO TO 21

PRODUCTION RUN -- IF TOPSUR IS UNDEFINED, USE JTOP AS ESTIMATED

17 IF(TOPSUR .LE. 0.0)GO TO 26

CALCULATE JTOP BASED ON VALUE OF TOPSUR

LEASTD = 9999999.9
DO 19 I=1,NUM
DIFF = ABS(ALT(I) - TOPSUR)
IF(DIFF .GT. LEASTD)GO TO 19
LEASTD = DIFF
JTOP = I
19 CONTINUE
GO TO 26

WRITE OUT THE ESTIMATED TOP OF SURFACE LAYER -- READ IN
THE ONE TO BE USED -- CALCULATE JTOP

21 CALL DREAD(NAMEF,2,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
CALL CODE
WRITE(ISURTP,204) ALT(JTOP)
TOPSUR = ALT(JTOP)
CALL CHAR(320.,YPOS,0,ISURTP,6,0,0)
YPOS = YPOS - 32.
IF(IFLAG(1) .EQ. 3)GO TO 26
IF(IFLAG(1) .EQ. 1)GO TO 24
CALL DREAD(NAMEF,3,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,6,3,0)
CALL CHAR(56.,YPOS,0,IER5,1,3,0)
CALL CHAR(64.,YPOS,0,ILINE(5),9,3,0)
CALL CHAR(160.,YPOS,0,ILINE(11),44,0,0)
NIN=6
CALL BLANK(IDATAF,10)
CALL INC(2,JTYPE,463.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
IF(JTYPE .EQ. 1)GO TO 22
CALL CHAR(0.,YPOS,0,ILINE,6,0,0)
CALL CHAR(47.,YPOS,0,IER5,40,0,0)
YPOS=YPOS-32.
CALL CODE
READ(IDATAF,100) TOPSUR
ALT(JTOP) = TOPSUR

A-90
GO TO 17
22 IF(IX .GT. 9) GO TO 23
   CALL CHAR(0.,YPOS,0.,ILINE,18,0,0)
   CALL CHAR(143.,YPOS,0.,IERS,46,0,0)
   YPOS = YPOS - 32.
   GO TO 17
23 CALL CHAR(0.,YPOS,0.,ILINE,6,0,0)
   CALL CHAR(56.,YPOS,0.,IERS,10,0,0)
   CALL CHAR(360.,YPOS,0.,IERS,18,0,0)
   YPOS = YPOS - 32
   CALL MET PROFILE, SUBROUTINE RMETP, TO DETERMINE LAYER VALUE
C
24 CALL NGRAF
   CALL RWDIS(NAME,1)
   CALL EXEC(9,RMETP)
   CALL RWDIS(NAME,0)
   CALL GRAY(1)
   CALL CLEAR
   YPOS = 474.
   CALL DREAD(NAMEF,5,ILINE)
   CALL LERS(YPOS)
   CALL CHAR(0.,YPOS,0.,ILINE,50,0,0)
   CALL CODE
   WRITE (IDATAF,100) TOPSUR
   CALL CHAR(400.,YPOS,0.,IDATAF,7,0,0)
   ALT(JTOP) = TOPSUR
   YPOS = YPOS - 32.
   GO TO 17
C
WRITE OUT THE TOP OF THE SURFACE LAYER AND WIND DIRECTION
AND SPEED AT THE TOP
C
26 CONTINUE
   WRITE (6,205) TOPSUR,IDIR(JTOP),SPEED(JTOP)
C
CALCULATE SOURCE STRENGTH
C
SIGHCL = 2.276E3 * PHCL * QC1 * AA * (TEMP(1) + 273.15)/
   PRESS(1) * TOPSUR**BB
C
CALCULATE AND WRITE OUT THE MEAN WIND SPEED, ASPD, AND
DIRECTION, ADIR
C
DO 28 I=2,JTOP
   IF(IABS(IDIR(I) - IDIR(I - 1)) .LT. 180) GO TO 28
   DO 27 J=1,JTOP
      IF(IDIR(J) .LT. 180) IDIR(J) = IDIR(J) + 360
      GO TO 31
   27 CONTINUE
   28 CONTINUE
C
31 ASPD = 0.0
ADIR = 0.0
DO 32 I=2,JTOP
I1 = I - 1
DALT = ALT(I) - ALT(I1)
ASPD = ASPD + 0.5 * (SPEED(I) + SPEED(I1)) * DALT
32 ADIR = ADIR + 0.5 * FLOAT(IDIR(I) + IDIR(I1)) * DALT
C
DO 34 I=1,JTOP
34 IF(IDIR(I) .GT. 360) IDIR(I) = IDIR(I) - 360
C
DALT = ALT(JTOP) - ALT(1)
ASPD = ASPD/DALT
ADIR = ADIR/DALT
IF(ADIR .GT. 180.0) GO TO 35
ADIR = ADIR + 180.0
GO TO 36
35 ADIR = ADIR - 180.0
C
36 WRITE (6,206) ASPD, ADIR
C
IS THIS A RESEARCH OR A PRODUCTION RUN?
C
IF(IOPT(2) .EQ. 0) GO TO 45
C
RESEARCH RUN -- READ IN SIGA
C
C** CALL SUBROUTINE RSIGA TO CALCULATE SIGMA VALUE
C
J1 = 1
J2 = 0
J3 = 0
DO 41 JJ=1,31
41 CONTINUE
IF(ABS(ALT(JJ)-304.8).LE.1.0) J3 = JJ
IF(ABS(PRESS(JJ)-1000.).LE.1.0) J2 = JJ
IF(J2.EQ.0 .OR. J3.EQ.0) SIGA = 7.0
IF(J2.EQ.0 .OR. J3.EQ.0) GO TO 42
CALL RSIGA(J1,J2,J3,RSIG)
SIGA = RSIG
42 CALL DREAD(NAMEF,6,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0.,ILINE,64,0,0)
CALL CODE
WRITE(IDATAF,102) SIGA
CALL CHAR(330.,YPOS,0.,DATAF,4,0,0)
CALL INC(2,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
YPOS = YPOS - 32.0

A-92
IF(IX.LE.25) GO TO 45
CALL DREAD(NAMEF,7,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,62,0,0)
NIN = 2
CALL BLANK(IDATAF,10)
CALL IN(0,JTYPE,358.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
CALL CODE
READ (IDATAF,101) ISIGA
IF(NIN .EQ. 1)ISIGA = ISIGA/10
SIGA = FLOAT(ISIGA)
YPOS = YPOS - 32.
WRITE OUT SIGA, THE SIGMA OF THE WIND AZIMUTH ANGLE
45 WRITE (6,208) SIGA
SIGAP = 0.0087266 * SIGA
CALCULATE THE HORIZONTAL AND VERTICAL CLOUD DIMENSIONS,
 i.e. SIGXO AND GSPEED
SIGXO = 0.297674 * ALT(31)
GSPEED = 0.232558 * ALT(31)
CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, CLDHT
CLDHT = ALT(31)
CLDRAD = 2.15 * SIGXO
IV2 = 0
IF(CLDRAD+ALT(31) .GE. ALT(JTOP))IV2 = 1
SIGZO = SIGXO
IF(IV2 .EQ. 1)SIGZO = (ALT(JTOP) - ALT(31) + CLDRAD)/4.3
IF(SIGZO .LT. 0.0) GO TO 47
CLDHT = 0.5 * ALT(JTOP)
SIGZO = 0.64 * CLDHT/2.15
GO TO 49
47 IF(IV2 .EQ. 1)CLDHT = 0.5 * (ALT(JTOP) + ALT(31) - CLDRAD)
49 WRITE (6,209) CLDHT
CALL THE SEGMENT RCONC
CALL WGRAF
CALL RWDIS(NAME,1)
CALL EXEC(9,RCONC)
CALL RWDIS(NAME,0)
END OF RCLOR
END

A-93
SUBROUTINE RYDIS(H,E, JJ)
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
   IDIR(31), I0PT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
   ISMON(2), ISYER, IY2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
   LMON(2), LYEAR, LU, NUM, PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
   SIGHL, RADDEC, RATONC, CLDRAD, R2, R3, SAVEA(30), SAVER(30), SIGA,
   SIGK, SIGS, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, SB, TEMP(31),
   TOPSUR, TWOP, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
   YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL ITIME
INTEGER YES
EQUIVALENCE (C1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
   (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
   (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))
INTEGER ODCB(144), OBUF(669)
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1), ALT(1))
CALL OPEN(ODCB, IERR, NAME, 0)
IF(JJ .EQ. 1) CALL WRITF(ODCB, IERR, OBUF, 669)
IF(JJ .EQ. 0) CALL READF(ODCB, IERR, OBUF, 669)
CALL CLOSE(ODCB, IERR)
RETURN
END
SUBROUTINE DREAD(NAMEF, LNUM, ILINE)
DIMENSION NAMEF(3), IDCB(276), IBUF(40), ILINE(32), IPAR(5)
CALL RMPAR(IPAR)
LU = IPAR(1)
CALL OPEN(IDCBIERR, NAMEF, 0)
LOOP = LNUM - 1
DO 10 I = 1, LOOP
   CALL BLANK(IBUF, LU)
   CALL READF(IDCBIERR, IBUF)
10  CONTINUE
   CALL BLANK(IBUF, LU)
   CALL READF(IDCBIERR, IBUF)
   CALL CODE
   READ(IBUF, 100) (ILINE(I), I = 1, 32)
100  FORMAT(32A2)
   CALL CLOSE(IDCBIERR)
RETURN
END
SUBROUTINE BLANK(IBUF, II)
DIMENSION IBUF(40)
DATA IBLK/2H/
DO 10 I = 1, II
10  IBUF(I) = IBLK
RETURN
A-94
SUBROUTINE LERS(YPOS):
DIMENSION IERS(32)
DATA IERS/32*2H, ' '
IF(YPOS.LE.48) YPOS = 458.0
CALL CHAR(0.,YPOS,0.,IERS,64,0.,0.)
CALL CHAR(0.,YPOS-16.,0.,IERS,64,0.,0.)
RETURN
END

SUBROUTINE RSIGA(J1,J2,J3,RSIG)

C*** THIS SUBROUTINE CALCULATES A SIGMA VALUE GIVEN
C*** ALITUDE, SPEED, TEMP, AND PRESSURE FOR THE
C*** FIRST LEVEL OF DATA, THE 1000FT LEVEL OF DATA
C*** AND THE 1000MB LEVEL OF DATA

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
ISMONC(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
LMON(2), LYEAR, LU, NUM, PI, PI0VR2, PI143, PRESS(31), PTEMP(31),
SIGHCL, RADDEG, RATOMIC, CLDRAD, R2, R3, SAVEC(30), SAVER(30), SIGA,
SIGA0, SIGX, SIGE(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
TOPSUR, T0PI, ASPD, YPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES

DATA C1,C2,C3,C4,C5,C6/-008,-00175, 0008, 50864522, 1132, 1 3.8163/,
DATA C7/-029/

C CALCULATION OF SIGAR
C NEWTONS METHOD FOR SOLUTION OF F(X,B,D) = 0
F(X,B,D) = (1-X**4)/(16.*X**2*(ALOG(0)+C4-2.*ALOG(1.+X))
1 - ALOG(1.+X**2)+2.*ATAN(X)**2) - B
FP(X,D) = (-X**4-1.)/(8.*X**3*(ALOG(0)+C4-2.*ALOG(1.+X))
1 - ALOG(1.+X**2)+2.*ATAN(X)**2) + (1.-X**4)/(2.*(1.+X))
1 *(1.+X**2)*(ALOG(D)+C4-2.*ALOG(1.+X)-ALOG(1.+X**2)+
1 2.*ATAN(X)**3)

C*** READ IST DATA LEVEL
Z1 = ALT(J1)
V1 = SPEED(J1)
T1 = TEMP(J1)
PZ1 = PRESS(J1)

C*** READ 1000MB DATA LEVEL
c
Z2 = ALT(J2)
V2 = SPEED(J2)
T2 = TEMP(J2)
PZ2 = PRESS(J2)

c
C*** READ 1000FT DATA LEVEL
C
Z3 = ALT(J3)
V3 = SPEED(J3)
T3 = TEMP(J3)
PZ3 = PRESS(J3)

c
C ** CONVERT TO PROPER UNITS
C
V1 = V1*.514791
V2 = V2*.514791
V3 = V3*.514791
Z1 = Z1*.3048
Z2 = Z2*.3048
Z3 = Z3*.3048
T1 = T1+273.16
T2 = T2+273.16
T3 = T3+273.16

c
C*** INITIALIZE Z0
C
Z0 = .20
C PZ1 AND PZ3 IN MILLIBARS
C V1, V2 AND V3 IN METER/SEC
C Z1, Z2 AND Z3 IN METERS
C T1, T2 AND T3 IN DEG K
C Z0 IN METERS
E = 22.9183118
Y = V2
T = (T1+T2+T3)/3.
Z = (Z1*Z2*Z3)**.33333
THETA1 = T1*((1000./PZ1)**.288)
THETA2 = T2
THETA3 = T3*((1000./PZ3)**.288)
ZA = (Z1+Z2+Z3)/3.
THETAA = (THETA1 + THETA2 + THETA3)/3.
D = Z/Z0
Z0Z0 = ALOG(D)
DZTHET = ((Z1-ZA)*(THETA1-THETAA)+(Z2-ZA)*(THETA2-THETAA)
1 + (Z3-ZA)*(THETA3-THETAA))/((Z1-ZA)**2 + (Z2-ZA)**2
1 + (Z3-ZA)**2)
B = 9.8*DZTHET*Z**2/(T*V**2)
IF(B) 2,25,6
2 CONTINUE
R = 1.5
U = F(R,B,D)
DO 3 I = 1,50
R1 = R - F(R,B,D)/FP(R,D)
U = F(R1,B,D)
IF(ABS(R1-R).LT.1.E-7) GO TO 21
IF(I.EQ.49) USAV = U
IF(I.NE.50) GO TO 888
IF(USAU.LT.0..AND.U.GT.0..OR.USAU.GT.0..AND.U.LT.0.) GO TO 21
888 CONTINUE
3 R = R1
RSIG = 30.
GO TO 1000
6 AP = Z020 - 1.
Z00L10 = (C6*Z0)/(7.*Z)
A1 = 1.
A2 = 1./((SQRT(B) * 7.*AP)
A3 = -(AP + 1.)/(7.*AP)
RAD = A2**2 - 4.*A1*A3
IF(RAD) 70,80,90
70 CONTINUE
RSIG = 30.
GO TO 1000
80 RE11 = -A2/(2.*A1)
S1 = 1. - 7.*RE11**2
GO TO 26
90 RE1 = (-A2 + SQRT(RAD))/(2.*A1)
RI4 = RE1**2
Z00L4 = Z0*RI4/(Z*(1. -7.*RI4))
IF(B.LT.C3) GO TO 37
IF(B.GE.C3) GO TO 38
21 RI1 = (1.-R1**4)/16.
Z00L1 = Z0*RI1/Z
A = Z020 +C4-2.*ALOG(1.+R1)-ALOG(1.+R1**2)+2.*ATAN(R1)
IF(B.LT.C1) GO TO 22
IF(B.GE.C1.AND.B.LT.C2) GO TO 23
IF(B.GE.C2) GO TO 24
22 RSIG = E*2.7/A
GO TO 1000
23 FB2 = 2.7 + 112.*(-C1 + B)
RSIG = E*FB2/A
GO TO 1000
24 FB3 = 3.4 - 725.5*(-C2 +B)
RSIG = E*FB3/A
GO TO 1000
25 R12 = 0
Z00L2 = 0
RSIG = 48.816/ALOG(D)
GO TO 1000
26 R13 = (S1-1.)/(-7.)


\[
Z00L3 = 20*RI3/(Z*(1. - 7.*RI3))
\]

IF(B.LT.C3) GO TO 27
IF(B.GE.C3) GO TO 28

27 \[
FB3 = 3.4 - 725.5*(-C2 + B)
\]

RSIG = (E*FB3)/( 7.*RI3/( 1. - 7.*RI3) + Z0ZO )
SIGR20=(E*FB3)/(C6+Z0ZO)
IF(RI3.GE.C5) GO TO 110
GO TO 1000

110 CONTINUE
RSIG = SIGR20
GO TO 1000

28 \[
FB4 = 1.55 + 38.04*(B - .0008)
\]

RSIG = (E*FB4)/( 7.*RI4/( 1. - 7.*RI4) + Z0ZO )
SIGR21=(E*FB4)/(C6+Z0ZO)
IF(RI4.GE.C5) GO TO 115
GO TO 1000

115 CONTINUE
RSIG = SIGR21
GO TO 1000

37 \[
FB3 = 3.4 - 725.5*(-C2+B)
\]

RSIG = (E*FB3)/( 7.*RI4/( 1. - 7.*RI4) + Z0ZO )
SIGR20=(E*FB3)/(C6+Z0ZO)
IF(RI4.GE.C5) GO TO 120
GO TO 1000

120 CONTINUE
RSIG = SIGR20
GO TO 1000

38 \[
FB4 = 1.55 + 38.04*(B - .0008)
\]

FB5 = 2.35 + 5.43*(B - C7)
RSIG = (E*FB4)/( 7.*RI4/( 1. - 7.*RI4) + Z0ZO )
SIGR21=(E*FB4)/(C6+Z0ZO)
SIGR22 = (E*FB5)/(C6+Z0ZO)
IF(RI4.GE.C5.AND.B.LT.C7) GO TO 125
IF(RI4.GE.C5.AND.B.GE.C7) GO TO 126
GO TO 1000

125 CONTINUE
RSIG = SIGR21
GO TO 1000

126 CONTINUE
RSIG = SIGR22
GO TO 1000

C

C*** CHECK FOR VALID SIGA VALUE
C

1000 CONTINUE
IF (RSIG.LE.0. .OR. RSIG.GT.30.) RSIG = 30.
RETURN
END
END$
PROGRAM RMETP

COMMON BLOCK

COMMON ALT(31),AL1,CONDMA,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
IDIR(31),10PT(3),ITIME,IDAY,MOUTH(2),IYEAR,ISTIM,ISDAY,
ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
LMOH(2),LYEAR,LU,HUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
SIGCL,RADEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
SIGX0,SIGX,SPED(31),SQRP2I,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
TOPSUR,TWOP1,ASPD,YPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN

LOGICAL LTME
INTEGER YES
EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
(QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
(AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
(HEAT,VPAR(10)),(HEATM,VPAR(11)),(HEATV,VPAR(12)),
(HPCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
(PAL203,VPAR(16)),(PHO,VPAR(17)),(GAMMAK,VPAR(18))

DIMENSION WSCX(31),WSY(31),DTX(31),DTY(31),PTX(31),PTY(31),
1 WDX(31),WDY(31)
DIMENSION ISTP(3),ITTP(3),ISPT(3),ITPT(3),ISWS(3),ITWS(3)
DIMENSION ISWD(3),ITWD(3),XTIC(2),YDTIC(2),ICUR1(21)
DIMENSION ITST(10),TPR(6),IDCB(144)
DIMENSION X(4),Y(4),XTIC(2),XTIC2(2),YTIC1(2),YTIC2(2)
DIMENSION X(2),Y(2),IAAL(8),TSURX(20),BSURX(10)
DIMENSION IALTCH(336),IALT(22),IHARD(16)
DIMENSION IXNUM(13),IYNUM(22),IALT(18)
DIMENSION ITEMP(3),IPRES(3),IDENSD(3)
DIMENSION IDATL(2),ITIML(2)
DIMENSION IDATE(6),AWDIR(31)
DIMENSION ITIME(2)
DIMENSION APTEMP(31)
DIMENSION XL(2),YL(2),IDT(12),IPT(11),IWS(8),IWD(10)
DIMENSION ISURL(30),IALTSP(8)
DIMENSION ISUR(22),ISURL(16),IALTP(8),IALTC(8)
DIMENSION ICYR(4),IST(12),ITOP(2),YWD1(2),YWD2(2),XWD1(18)
DIMENSION IBOT(2)
DIMENSION KWD(2),IWDL1(18),IWDL2(18),IWDL3(18),IWDL4(18)
DIMENSION ITPV(3),NAME(3)
DIMENSION IMET(2),INSTAL(2,2),ISTAB(4)
INTEGER RMTQ(3)
DATA IHARD/2HHA,2HRD,2H C,2HOP,2HY,2HDE,2HSI,2HRE,2HD?,2H
,2HYE,2HS,2H ,2H ,2HNO/
DATA RMTQ/2HHRM,2HET,1HQ/
DATA NAME/036522B,2HEE,1HD/
DATA IWDL1/2H 0,2H ,2H ,2H ,2H 9,2HO,2H ,2H ,2H8,2HO ,
2H ,2H ,2H2,2HO ,2H ,2H ,2H36,2HO /

A-99
DATA LCHAR/1H0/, IALT/2H01, 2H23, 2H45, 2H67, 2H89, 2HAB, 2HCD,
1 2HF, 2HG, 2HIJ, 2HKL, 2HMN, 2HOP, 2HQR, 2HST, 2HUV,
1 2HXY, 2HYZ, 2H+-, 2H*/, 2H0/

C ** THE FOLLOWING DATA STATEMENT CONTAINS OCTAL REPRESENTATION
C ** OF AN ALTERNATE CHARACTER SET AS FOLLOWS: 0-9, A-Z, AND
C ** SPECIAL CHARACTERS +,-, *, /,**</p>

DATA IALTCH/36B, 41B, 41B, 36B, 4*0, 0, 21B, 77B, 1B, 4*0,
1 23B, 45B, 45B, 31B, 4*0, 42B, 41B, 51B, 66B, 4*0,
1 14B, 24B, 77B, 4B, 4*0, 72B, 51B, 46B, 4*0,
1 36B, 45B, 45B, 2B, 4*0, 60B, 43B, 44B, 70B, 4*0,
1 26B, 51B, 51B, 26B, 4*0, 20B, 51B, 51B, 36B, 4*0,
1 37B, 50B, 50B, 37B, 4*0, 77B, 51B, 51B, 26B, 4*0,
1 36B, 41B, 41B, 22B, 4*0, 77B, 41B, 36B, 4*0,
1 77B, 51B, 51B, 41B, 4*0, 77B, 50B, 50B, 40B, 4*0,
1 0, 41B, 77B, 41B, 4*0, 42B, 41B, 76B, 40B, 4*0,
1 77B, 14B, 22B, 41B, 4*0, 77B, 1B, 1B, 4*0,
1 77B, 20B, 10B, 20B, 77B, 3*0, 77B, 30B, 6B, 77B, 4*0,
1 36B, 41B, 41B, 36B, 4*0, 77B, 44B, 44B, 30B, 4*0,
1 34B, 42B, 42B, 35B, 4*0, 77B, 44B, 46B, 31B, 4*0,
1 22B, 51B, 45B, 22B, 4*0, 40B, 40B, 77B, 40B, 40B, 3*0,
1 76B, 1B, 1B, 76B, 4*0, 74B, 2B, 1B, 2B, 74B, 3*0,
1 76B, 1B, 36B, 1B, 76B, 3*0, 61B, 12B, 06B, 12B, 61B, 3*0,
1 60B, 10B, 17B, 10B, 60B, 3*0, 41B, 43B, 45B, 51B, 61B, 3*0,
1 24B, 37B, 24B, 3*0, 5*4B, 3*0, 21B, 12B, 37B, 12B, 21B, 3*0,
1 1B, 2B, 4B, 10B, 20B, 3*0, 0, 36B, 41B, 5*0, 0, 41B, 36B, 5*0/

DATA IALTC1/0, 12B, 12B, 12B, 4*0/
DATA IALTP/0, 1B, 6*0/
DATA IALTCL/0, 12B, 6*0/
DATA IALTP/0, 1B, 6*0/
DATA IMET/2H(M, 1H)/
DATA INSTAB/2HVA, 2HFB, 2HK, ZHC /
DATA ISTAB/2HST, 2HAB, 2H H, 2HT:

C*** CALL VERSION SUBROUTINE TO DETERMINE IF RUNNING ON CRT OR PLASMASCOPE.... IVERS=0 FOR PLASMA IVERS=1 FOR CRT
C
C CALL VERSN(IVERS)
C ** CALL GRAF(1) TO INITIALIZE PLASMASCOPE
CALL GRAF(1)
C ** CALL CLEAR TO CLEAR PLASMASCOPE
CALL CLEAR
C ** CALL ALTERNATE CHARACTER SET
CALL LALT(LCHAR, IALTCH, 10)
CALL LALT(1HA, IALTCH(81), 26)
CALL LALT(1H+, IALTCH(289), 6)
CALL LALT(1H+, IALTCH(1), 1)
CALL LALT(1H, IALTSP, 1)
CALL LALT(1H, IALTCL, 1)
CALL LALT(1H, IALTTP, 1)

A-101
C ** CALL SETOR(XORG, YORG) TO INITIALIZE X, Y ORIGIN
C ** CALL SETSC(XSCAL, YSCAL) TO SET SCALE FACTORS
CALL SETSC(1.0, 1.0)
CALL SETOR(0.0, 0.0)
C ** READ THE COMMON DISC FILE
CALL RDISC(NAME, 0)
C ** LINE(X, Y, NXY, MODE) TO PLOT LINE
C ** X, Y = CO-ORDINATES
C ** NXY = NUMBER OF POINTS TO BE PLOTTED
C ** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
C ** CALL POINT(X, Y, NXY, MODE) SAME AS ABOVE EXCEPT PLOTS POINTS
C ** PRINT DATE
CALL CHAR(20.0, 490.0, IDATE, 4, 2, 1)
XL(1) = 20.0
XL(2) = 48.0
YL(1) = 488.0
YL(2) = 488.0
CALL LINE(XL, YL, 2, 0)
CALL CODE
WRITE(IDATE, 3002) ISDAY, ISMOM(1), ISMOM(2), ISYEAR
3002 FORMAT(12.1X, A2, A1, 1X, I4)
CALL CHAR(60.0, 490.0, IDATE, 11, 2, 1)
C ** PRINT TIME
CALL CHAR(164.0, 490.0, ITIMD, 4, 2, 1)
XL(1) = 164.0
XL(2) = 192.0
CALL LINE(XL, YL, 2, 0)
CALL CODE
WRITE(ITIME, 3001) ISTIM
3001 FORMAT(I4)
CALL CHAR(304.0, 490.0, ITIME, 4, 2, 1)
CALL CHAR(240.0, 490.0, 1, IFLAG(4), 1, 2, 1)
IF(IVERSN .EQ. 0)CALL CHAR(248.0, 490.0, 1, LAUNTD(4), 2, 2, 1)
IF(IVERSN .EQ. 1)CALL CHAR(246.0, 490.0, 1, LAUNTD(4), 2, 2, 1)
IF(IFLAG(3) .EQ. 0)GO TO 2
I = IFLAG(3) - IFLAG(3)/3
CALL CHAR(308.0, 490.0, 0, INSTALL(1, I), 4, 2, 1)
XL(1) = 308.0
XL(2) = 336.0
CALL LINE(XL, YL, 2, 0)
C ** PRINT SURFACE PRESSURE AND DENSITY
2 CALL CHAR(20.0, 475.0, 0, IURL1, 60, 2, 1)
IF(IVERSN .EQ. 0) CALL CHAR(468.0, 478.0, 0, IEXP3, 1, 2, 1)
IF(IVERSN .EQ. 1) CALL CHAR(318.0, 478.0, 0, IEXP3, 1, 2, 1)
XL(1) = 20.0
XL(2) = 76.0
YL(1) = 473.0
YL(2) = 473.0
CALL LINE(XL, YL, 2, 0)
IF (IVERSN .EQ. 0) GO TO 3
CALL CHAR (374.0, 475.0, 0, ISTAB, 8, 2, 1)
CALL CHAR (466.0, 475.0, 0, ISTL(4), 1, 2, 1)
XL(1) = 374.0
XL(2) = 422.0
CALL LINE (XL, YL, 2, 0)
CALL CODE
WRITE (IPRESD, 2007) ALT(31)
CALL CHAR (428.0, 475.0, 0, IPRESD, 6, 2, 1)

C ** PRINT SURFACE -- TOP LAYER HEADER -- BOT LAYER HEADER (IF REQD)
3 IF (IVERSN .NE. 0) GO TO 4
I = 20
IF (IFLAG(2) .EQ. 1) I = 32
CALL CHAR (222.0, 461.0, 0, ISURT1, I, 2, 1)
GO TO 5
4 I = 26
IF (IFLAG(2) .EQ. 1) I = 44
CALL CHAR (222.0, 461.0, 0, ISURT, I, 2, 1)
5 XL(1) = 222.0
XL(2) = 278.0
YL(1) = 459.0
YL(2) = 459.0
CALL LINE (XL, YL, 2, 0)
XL(1) = 302.0
XL(2) = 374.0
CALL LINE (XL, YL, 2, 0)
IF (IFLAG(2) .NE. 1) GO TO 8
XL(1) = 398.0
XL(2) = 470.0
CALL LINE (XL, YL, 2, 0)
C ** PRINT DRY TEMPERATURE
8 CALL CHAR (30., 450., 0, IDT, 24, 2, 1)
C ** PRINT POTENTIAL TEMPERATURE
CALL CHAR (30., 440., 0, IPT, 22, 2, 1)
C ** PRINT WIND SPEED
CALL CHAR (30., 430., 0, IWSD, 31, 2, 1)
C ** PRINT WIND DIRECTION
CALL CHAR (30., 420., 0, IWD, 20, 2, 1)
C ** DRAW X AXIS
CALL LINE (X, Y, 2, 0)
C ** DRAW Y AXIS
CALL LINE (X(3), Y(3), 2, 0)
C ** DO LOOP TO ADD TIC MARKS FOR X AXIS
XTIC = 70.
XTIC2(1) = 88.
XTIC2(2) = 92.
XNUM1 = 62.
DO 10 I = 1, 13
XTIC = XTIC + 30.
XTIC1(1) = XTIC
XTIC1(2) = XTIC
CALL LINE(XTIC1,XTIC2,2,0)
XTIC1(1) = XTIC1(1) + 15.
XTIC1(2) = XTIC1(2) + 15.
IF(I.EQ.13) GO TO 13
CALL LINE(XTIC1,XTIC2,2,0)
13 CONTINUE
XNUM1 = XNUM1 + 30.
IF(I.EQ.1) CALL CHAR(84.,80.,0,I1NUMUS,1,2,1)
CALL CHAR(XNUM1,80.,0,I1NUM(I),2,2,1)
10 CONTINUE
C ** DRAW TIC MARKS FOR WIND DIRECTION SCALE
XWD2(1) = 300.
XWD2(2) = 460.
YWD2(1) = 70.
YWD2(2) = 70.
YWD1(1) = 68.
YWD1(2) = 72.
CALL LINE(XWD2,YWD2,2,0)
CALL CHAR(310.,50.,0,IWD,20,2,1)
C ** PRINT LABELS FOR X-AXIS
CALL CHAR(100.,70.,0,ISTL,24,2,1)
C ** DO LOOP TO ADD TIC MARKS TO Y-AXIS
YTIC = 58.
XTIC2(1) = 98.
XTIC2(2) = 102.
DO 20 I = 1,11
YTIC = YTIC + 32.
XTIC2(1) = YTIC
YTIC2(2) = YTIC
N = (I-1)*2 + 1
CALL CHAR(64.,YTIC2,0,IYNUM(N),4,2,1)
CALL LINE(XTIC2,YTIC2,2,0)
20 CONTINUE
C ** PRINT LABEL FOR Y-AXIS
YX = 360.
DO 30 I = 1,8
YX = YX - 20.
CALL CHAR(30.,YX,0,IALTL(I),2,2,1)
30 CONTINUE
CALL CHAR(30.,YX-20.,0,IMET,3,2,1)
C ** THIS PRINTS SURFACE PRESSURE AND DENSITY VALUES
A = PRESS(1)
CALL CODE
WRITE(IPRESD,2007) A
2007 FORMAT(F6.1)
IF(IVERSN.EQ.0)CALL CHAR(196.,475.,0,IPRESD,6,2,1)
IF(IVERSN.EQ.1)CALL CHAR(133.,475.,0,IPRESD,6,2,1)
A = SUREDEN
CALL CODE
** C ** PRINT DRY TEMPERATURES

A = TEMP(1)
CALL CODE
WRITE(ISPT,2007) A
CALL CHAR(230.,450.,0,ISPT,6,2,1).

** C ** PRINT POTENTIAL TEMPERATURES

A = PTEMP(1) - 273.15
CALL CODE
WRITE(ISPT,2007) A
CALL CHAR(230.,440.,0,ISPT,6,2,1)

DO 133 JJ=1,NUM
IF(ALT(JJ).GE.4000.) GO TO 3131
WSY(JJ) = ALT(JJ)*.08 + 90.
DTY(JJ) = ALT(JJ)*.08 + 90.
PTY(JJ) = ALT(JJ)*.08 + 90.
WDY(JJ) = ALT(JJ)*.08 + 90.
AWDIR(JJ) = IDIR(JJ).
APTEMPP(JJ) = PTEMP(JJ) - 273.15
133 CONTINUE
JJ = NUM + 1
3131 ILP = JJ - 1

** C** CALL SUBROUTINE TO ROTATE WIND DIRECTION FOR PLOTTING

CALL WINDS(AWDIR,ILP,ISC)
DO 123 IK=1,9
N = (IK-1)*2 + 1
CALL LINE(XWD1(N),YWD1,2,0)
XBBW = XWD1(N) - 8.
YBBW = 60.
IF(ISC.EQ.1) CALL CHAR(XBBW,YBBW,0,IWD1(N),4,2,1)
IF(ISC.EQ.2) CALL CHAR(XBBW,YBBW,0,IWD2(N),4,2,1)
IF(ISC.EQ.3) CALL CHAR(XBBW,YBBW,0,IWD3(N),4,2,1)
IF(ISC.EQ.4) CALL CHAR(XBBW,YBBW,0,IWD4(N),4,2,1)
123 CONTINUE
DO 134 KK=1,ILP
WSX(KK) = (SPEED(KK))*6. + 160.
DTX(KK) = (TEMP(KK))*6. + 160.
PTX(KK) = (APTEMPP(KK))*6. + 160.
IF(TEMP(KK) .LT.-10.) DTX(KK) = 100.
IF(TEMP(KK) .GT. 50.) DTX(KK) = 460.
IF(APTEMPP(KK) .LT.-10.) PTX(KK) = 100.
IF(APTEMPP(KK) .GT. 50.) PTX(KK) = 460.
WDX(KK) = ABS(AWDIR(KK))*444444 + 300.
134 CONTINUE

** C ** PRINT WIND SPEEDS

A = SPEED(1)
CALL CODE
WRITE(ISWS,2007) A
CALL CHAR(230.,430.,0,ISWS,6,2,1)
C ** PRINT WIND DIRECTIONS
A = IDIR(1)
CALL CODE
WRITE(ISWD,2007) A
CALL CHAR(230.,420.,0,ISWD,6,2,1)
C ** THIS PORTION DRAWS THE WIND SPEED LINE
CALL DLINE(WSX,WSY,ILP,0,8,4)
XHT = WSY(ILP) + 3.
CALL CHAR(WSX(ILP),XHT,0,ICRVT(1),2,2,1)
C ** THIS PORTION DRAWS THE DRY TEMPERATURE LINE
CALL LINE(DTX,DTY,ILP,0)
XHT = DTY(ILP) - 5.0
CALL CHAR(DTX(ILP)+4.0,XHT,0,ICRVT(2),2,2,1)
C ** THIS PORTION DRAWS THE POTENTIAL TEMPERATURE LINE
CALL DLINE(PTX,PTY,ILP,0,4,4)
XHT = PTY(ILP) + 3.
CALL CHAR(PTX(ILP),XHT,0,ICRVT(3),2,2,1)
C ** THIS PORTION DRAWS THE WIND DIRECTION LINE
I1 = 1
DO 777 I2=2,ILP
IF(AUDIR(I2),GE.0.) GO TO 777
HUMP = I1 - I2
CALL DLINE(WDX(I1),WDY(I1),HUMP,0,4,8)
I1 = I1
777 CONTINUE
HUMP = ILP - I1 + 1
CALL DLINE(WDX(I1),WDY(I1),HUMP,0,4,8)
XHT = WDY(ILP) - 5.0
CALL CHAR(WDX(ILP)+4.0,XHT,0,ICRVT(4),2,2,1)
C ** THIS PORTION DRAWS TIC MARKS AT VALID DATA POINT OF Y AXIS
DO 330 K=1,ILP
YDTIC(1) = ALT(K)*.08 + 90.
YDTIC(2) = YDTIC(1)
CALL LINE(XDTIC,YDTIC,2,0)
330 CONTINUE
C
DRAW THE CLOUD

YCLOUD = ALT(31) * 0.08 + 90.0
CALL CLOUD(250.0,YCLOUD)
C
WRITE OUT THE TOP OF THE SURFACE LAYER LINE AND ALLOW IT TO BE MOVED UP AND DOWN

CALL MOVEM(JTOP,ILP,2,ITOP,318.0,TSURX,10)
TOPSUR = ALT(JTOP)
C
A-106
IF REQUESTED, WRITE OUT THE BOTTOM OF THE SURFACE LAYER
LINE AND ALLOW IT TO BE MOVED UP AND DOWN

IF(IFLAG(2) .NE. 1) GO TO 444
CALL MOVE(JBOT, ILP, 1, IBOT, 414.0, BSURX, 5)
ZB = ALT(JBOT)

C*** CHECK FOR CRT OR PLASMA VERSION

444 IF(IVERSION .EQ. 1) GO TO 446
CALL CHAR(24.,16.,0,IHARD(1),18,3,0)
CALL CHAR(180.,16.,0,IHARD(10),0,0,0)
CALL CHAR(232.,16.,0,IHARD(14),6,0,0)
CALL INC(1, JTYPE, 0.,0.,0.,0.,31,0,31,IX, IY)
IF(IX.GT.15) GO TO 446
CALL RWDIS(NAME, 1)
CALL EXEC(9, RMETQ)
CALL RWDIS(NAME, 0)

446 CONTINUE

C ** CALL RWDIS TO PASS CHANGES IN COMMON DIS FILE C

CALL RWDIS(NAME, 1)

C ** CALL NGRAF TO RE-INITIATE PLASMASCOPE
CALL CLEAR
CALL NGRAF
STOP
END
SUBROUTINE WINDS(WD, NWD, ISC)
DIMENSION WD(1), ENDPT(4), NUMUP(4)
EQUIVALENCE (J, LEAST)
DATA ENDPT/0.0, 90.0, 180.0, 270.0/
DO 2 I=1,4
2 NUMUP(I) = 0
WD2 = WD(I)
DO 8 I=2, NWD
WD1 = WD2
WD2 = WD(I)
6 D O 6 J=1,4
C1 = WD1 - ENDPT(J)
IF(C1 .LT. 0.0)C1 = C1 + 360.0
C2 = WD2 - ENDPT(J)
IF(C2 .LT. 0.0)C2 = C2 + 360.0
IF(ABS(C1-C2) .LE. 180.0) GO TO 6
NUMUP(J) = NUMUP(J) + 1
6 CONTINUE
8 CONTINUE
ISC = 1
LEAST = NUMUP(1)
DO 12 I=2, 4
IF(NUMUP(I) .GE. LEAST) GO TO 12
A-107
ISCC = I
LEAST = NUMUP(I)
12 CONTINUE
DO 17 I=1,NWD
WD(I) = WD(I) - ENDP(I)
IF(WD(I) .LT. 0.0) WD(I) = WD(I) + 360.0
17 CONTINUE
WD2 = WD(I)
DO 22 I=2,NWD
WD1 = WD2
WD2 = WD(I)
IF(ABS(WD1-WD2) .LE. 180.0)GO TO 22
WD(I) = - WD(I)
22 CONTINUE
RETURN
END
SUBROUTINE CLOUD(XP,YP)
C
COMMON ALT(31),A,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
IDIR(31),IOPT(3),ITIME,IDAY,MOUTH(2),IYEAR,ISTIM,ISDAY,
ISMON(2),ISYEAR,IY2,JTOP,JSUBT,LAUNTD(10),LTIME,LTIM,LDAY,
LMON(2),LYEAR,LU,NUM,P1,P10YR2,P143,PRESS(31),PTEMP(31),
SIGHC,RADDEG,ROTMC,CLDRAD,R2R3,SAVER(30),SAVER2(30),SIGA,
SIC,SGX,SPEED(31),SQR2PI,SURDEN,SIG20,SIGAP,SB,TEMP(31),
TOPSUR,TWOP1,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
YP0S,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
(QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
(AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
(HEATM,VPAR(10)),(HEATM,VPAR(11)),(HEATM,VPAR(12)),
(PHCL,VPAR(13)),(PC0,VPAR(14)),(PC02,VPAR(15)),
(PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
DIMENSION X(181),Y(181)
RADIUS = GAMMAX * ALT(31) * 0.08
DO 7 I=1,181
X(I) = RADIUS * COS(0.01745329252 * FLOAT(2 * I)) + XP
7 Y(I) = RADIUS * SIN(0.01745329252 * FLOAT(2 * I)) + YP
CALL LINE(X,Y,181,0)
RADIUS = 5.0
X(1) = XP + RADIUS
X(2) = XP
X(3) = XP - RADIUS
X(4) = XP
X(5) = X(1)
Y(1) = YP
Y(2) = YP + RADIUS
Y(3) = YP
Y(4) = YP - RADIUS
Y(5) = Y(1)
CALL LINE(X, Y, 5, 0)
X(2) = XP - RADIUS
Y(2) = YP
CALL LINE(X, Y, 2, 0)
X(3) = XP
Y(3) = YP + RADIUS
CALL LINE(X(3), Y(3), 2, 0)
RETURN
END

SUBROUTINE MOVEM(JND, MAXJND, MINJND, LAB, XLABEL, XLINE, NLINE)

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
   IDIR(31), IPOC(3), ITIME, IDAY, MONTH(2), IYEAR, MSTIM, ISDAY,
   ISMONC(2), ISYEAR, IY2, JTOP, JBOT, LAUNTD(10), LTIE, LTIM, LDAY,
   LMAG(3), LYEAR, LU, NUM, PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
   SIC, TCRD, TATMC, CLDRAE, R2, R3, SAVEA(30), SAVR(30), SICA,
   SICK0, SICX, SPEED(31), SQ2PI, SURDEN, SIG20, SIGCAP, SB, TEMP(31),
   TOPSUR, TWPDA, VPAR(10), CRTIME(31), DIST, YES, Y1, NUMRUN,
   YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIE
INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   (HEAT, VPAR(10)), (HEATM, VPAR(11)), (HEATB, VPAR(12)),
   (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
   (PAL03, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

2000 FORMAT (F6.1)
2001 FORMAT ("I3",0)

INTEGER QUES(13), ANS1, ANS2(2), ANS3(4), BLANKS(26)
DIMENSION LAB(1), XLINE(1), YLINE(2), JNDALT(3), JNDVAR(3, 4)
EQUIVALENCE (JNDR1, JNDVAR(1, 1)), (JNDR2, JNDVAR(1, 2)),
   (JNDR3, JNDVAR(1, 3)), (JNDR4, JNDVAR(1, 4))

DATA QUES/2HMO, 2HVE, 2H, 2H, 0.2HF, 2HSU, 2HRF, 2HAC, 2HE, 2HLA,
   2HVE, 2HR:
DATA ANS1/2HUP/, ANS2/2HDO, 2HWN/, ANS3/2HCO, 2HNT, 2HIN, 2HUE/
DATA BLANKS/26*2H /
NEWJND = 0
1 YLINE(1) = ALT(JND) * 0.08 + 90.0
YLINE(2) = YLINE(1)
DO 4 I = 1, NLINE
   J = 2 * I - 1
4 CALL LINE(XLINE(J), YLINE, 2, 0)
Y = YLINE(1) + 2.0
CALL CHAR(460.0, Y, 0, LAB, 4, 2, 1)
Y = Y - 10.0
CALL CODE
WRITE (JNDALT,2000) ALT(JND)
CALL CHAR(460.0,Y,0,JNDALT,6,2,1)
CALL CODE
WRITE (JNDVR1,2000) TEMP(JND)
YLABEL = PTEMP(JND) - 273.15
CALL CODE
WRITE (JNDVR2,2000) YLABEL
CALL CODE
WRITE (JNDVR3,2000) SPEED(JND)
CALL CODE
WRITE (JNDVR4,2001) IDIR(JND)
YLABEL = 450.0
DO 6 I=1,4
CALL CHAR(XLABEL,YLABEL,0,JNDVAR(1,I),6,2,1)
6  YLABEL = YLABEL - 10.0
IF(NEWJND.EQ. JND)GO TO 11
QUES(3) = LAB(1)
QUES(4) = LAB(2)
CALL CHARC(0.0,1.0,-1,QUES,26,3,0)
CALL CHARC(29.0,1.0,-1,ANS1,2,0,0)
CALL CHARC(36.0,1.0,-1,ANS2,4,0,0)
CALL CHARC(43.0,1.0,-1,ANS3,8,3,0)
11  CALL IN(I,J,0.0,0.0,0,0,0,0,31,0,31,I,J)
IF(I.LE. 20)GO TO 15
CALL CHAR(0.0,1.0,-1,BLANKS,51,0,0)
RETURN
15 IF(I.GE. 17)GO TO 18
NEWJND = MINO(JND + 1,MAXJND)
GO TO 22
18  NEWJND = MAXO(JND - 1,MINJND)
22  IF(NEWJND .EQ. JND)GO TO 11
DO 24 I=1,NLINE
J = 2 * I - 1
24  CALL LINE(XLINE(J),YLINE,2,1)
CALL CHARC(460.0,Y,0,JNDALT,6,1,1)
Y = Y + 10.0
CALL CHARC(460.0,Y,0,LAB,4,1,1)
YLABEL = 450.0
DO 26 I=1,4
CALL CHARC(XLABEL,YLABEL,0,JNDVAR(1,I),6,1,1)
26  YLABEL = YLABEL - 10.0
JND = NEWJND
GO TO 1
END
SUBROUTINE RWDIS(NAME, JJ)

C
C          COMMON BLOCK
C

A-110
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
  IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTEM, ISDAY,
  ISMON(2), IYEAR, IV2, JT0P, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
  LMON(2), LYEAR, LU, NUM, PI, PI4VR2, PI43, PRESS(31), PTEMP(31),
  SIGMA, SIGX0, SIGX, SPEED(31), SQR2P1, SURDEN, SIGZ0, SIGAP, SB, TEMP(31),
  TOPSUR, TWPRI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
  YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETn

LOGICAL LTIM
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
  (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
  (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
  (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
  (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
  (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

INTEGER OCDC(144), OBUF(669)
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1), ALT(1))
CALL OPEN(OCDC, IERR, NAME, 0)
IF(JJ.EQ.1)CALL WRITF(OCDC, IERR, OBUF, 669)
IF(JJ.EQ.0)CALL READF(OCDC, IERR, OBUF, 669)
CALL CLOSE(OCDC, IERR)
RETURN
END
PROGRAM RMEQ

COMMON BLOCK

COMMON ALT(31), AL1, CONHMAX, CONCPK, DEGRAD, ADIR, DOSP, E1, CLDHT,
     IDIR(31), IOPTC(3), ITIME, IDAY, NMONTH(2), IYEAR, ISTIM, ISDAY,
     ISMOC(2), IYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
     LMON(2), LYEAR, LU, NUM, PI, PI0FR2, PI43, P3SS(31), PTEMP(31),
     SIGHC, RADDEC, RATOMC, CCLRAD, R2, R3, SAVEA(30), SAVER(30), SIGA,
     SICNO, SICNO, SPEED(31), SQR2PI, SURDEN, SIGO, SIGAP, S8, TEMP(31),
     TOPSUR, TWOP, TVPAP, VPAR(18), CTRTIME(31), DIST, YES, YI, NUMRUN,
     YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
     (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
     (AAV, VPAR(7)), (BBV, VPAR(8)), (CCV, VPAR(9)),
     (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
     (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC0V, VPAR(15)),
     (PAL02, VPAR(16)), (PNO, VPAR(17)), (GAMMAV, VPAR(18))

DIMENSION WSX(31), WSY(31), DTX(31), DTY(31), PTX(31), PTY(31),
     WDX(31), WDY(31)
DIMENSION ISTE(3), ITP(3), ISPT(3), IPRTC(3), IWS(3), ITWS(3)
DIMENSION ISUD(3), ITWD(3), XDTIC(2), YDTIC(2), ICUR1(21)
DIMENSION ITEST(10), TPRC(6), IDCB(144)
DIMENSION X(4), Y(4), X2TIC(2), Y2TIC(2), YDIC(12), YDIC2(2)
DIMENSION KSC(2), YSC(2), IALT(8), TSURX(20), BSURX(10)
DIMENSION IALTCH(336), IALT(22)
DIMENSION IXNUM(13), IYNUM(22), IALT(8)
DIMENSION ITEMPD(3), IPRESO(3), IDENSO(3)
DIMENSION IDATL(2), ITIML(2)
DIMENSION IDATE(6), AWDIR(31)
DIMENSION ITIME(2)
DIMENSION ATEMP(31)
DIMENSION XLC(2), YLC(2), IDT(12), IPT(11), IWS(8), IW(10)
DIMENSION ISURL(30), IALTSP(8)
DIMENSION ISURT(22), IALT(8), IALTCL(8)
DIMENSION ICRT(4), ISTL(12), ITP(2), YWD1(2), YWD2(2), YWD3(18)
DIMENSION IBOOT(2)
DIMENSION XWD2(2), IWD1(18), IWD2(18), IWD3(18), IWD4(18)
DIMENSION IPTV(3), NAME(3)
DIMENSION IMET(2), INSTAB(2, 2), ISTAT(4)
DATA NAME/0365522B, 2HEE, 1HD/
DATA IWD1/2H 0, 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H , 2H /

A-112
DATA IALTCH/36B,41B,41B,36B,4*0,0,21B,77B,1B,4*0/, 0,1
1 23B,45B,45B,31B,4*0,42B,41B,51B,66B,4*0/, 2,3
1 14B,24B,77B,4B,4*0,72B,51B,51B,46B,4*0/, 4,5
1 36B,45B,45B,2B,4*0,60B,43B,44B,70B,4*0/, 6,7
1 26B,51B,51B,26B,4*0,20B,51B,51B,36B,4*0/, 8,9
1 37B,50B,50B,37B,4*0,77B,51B,51B,26B,4*0/, A,B
1 36B,41B,41B,22B,4*0,77B,41B,41B,36B,4*0/, C,D
1 77B,51B,51B,41B,4*0,77B,50B,50B,40B,4*0/, E,F
1 36B,41B,45B,26B,4*0,77B,10B,10B,77B,4*0/, G,H
1 0,41B,77B,41B,4*0,42B,41B,76B,40B,4*0/, I,J
1 77B,14B, 22B,41B,4*0,77B,1B,1B,1B,4*0/, K,L
1 77B,20B,10B,20B,77B,3*0,77B,30B,6B,77B,4*0/, M,N
1 36B,41B,41B,36B,4*0,77B,44B,44B,30B,4*0/, O,P
1 34B,42B,42B,35B,4*0,77B,44B,46B,31B,4*0/, Q,R
1 22B,51B,45B,22B,4*0,40B,40B,77B,40B,40B,3*0/, S,T
1 76B,1B,1B,76B,4*0,74B,2B,1B,2B,74B,3*0/, U,V
1 76B,1B,36B,1B,76B,3*0,61B,12B,04B,12B,61B,3*0/, W,X
1 60B,108,17B,10B,60B,3*0,41B,43B,45B,51B,61B,3*0/, Y,Z
1 2*4B,37B,2*4B,3*0,5*4B,3*0,21B,12B,37B,12B,21B,3*0/, +,-
1 1B,2B,4B,10B,20B,3*0,0,36B,41B,5*0,0,41B,36B,5*0/ ;, ( =.SP

DATA IALTCl/0,12B,12B,12B,4*0/
DATA IALTPI/0,1B,6*0/
DATA IALTCL/0,12B,6*0/
DATA IALTSP/8*0/
DATA IMET/2H(M,1H)/
DATA INSTA/2HYA,2HFB,2HKs,2HC /
DATA ISTAB/2HST,2HAB,2H H,2HT;/
C ** CALL GRAPH(1) TO INITIALIZE PLASMASCOPE
   CALL GRAPH(1)
C ** CALL CLEAR TO CLEAR PLASMASCOPE
C CALL CLEAR
C ** CALL ALTERNATE CHARACTER SET
   CALL LALT(LCHAR,IALTCH,10)
   CALL LALT(IHA,IALTCH(81),26)
   CALL LALT(IH+,IALTCH(89),6)
   CALL LALT(IH=,IALTCl(1),1)
   CALL LALT(IH",IALTSP,1)
   CALL LALT(IH:,IALTCL,1)
   CALL LALT(IH,.IALTLP,1)
C ** CALL SETOR(XORG,YORG) TO INITIALIZE X,Y ORIGIN
C ** CALL SETSC(XSCAL,YSCAL) TO SET SCALE FACTORS
   CALL SETSC(1.,1.)
   CALL SETOR(0.,0.)
C ** READ THE COMMON DISC FILE
C CALL RWDIS(NAME,0)
C ** LlNEX,Y,NXY,MODE) TO PLOT LINE
C ** X, Y = CO-ORDINATES
C ** NXY = NUMBER OF POINTS TO BE PLOTTED
C ** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
C ** CALL POINT(X,Y,HXY,MODE) SAME AS ABOVE EXCEPT PLOTS POINTS
C ** PRINT DATE
CALL CHAR(20.,490.,0,IDATL,4,2,1)
XL(1) = 20.
XL(2) = 48.
YL(1) = 488.
YL(2) = 488.
CALL LINE(XL,YL,2,0)
CALL CODE
WRITE(IDATE,3002) ISDAY,ISMON(1),ISMON(2),ISYEAR
3002 FORMAT(12,1X,A2,A1,1X,I4)
CALL CHAR(60.,490.,0,IDATE,11,2,1)
C ** PRINT TIME
CALL CHAR(164.,490.,0,ITIML,4,2,1)
XL(1) = 164.
XL(2) = 192.
CALL LINE(XL,YL,2,0)
CALL CODE
WRITE(ITIME,3001) ISTIM
3001 FORMAT(I4)
CALL CHAR(204.,490.,0,ITIMM,4,2,1)
CALL CHAR(240.0,490.0,0,IFLAG(4),1,2,1)
CALL CHAR(246.0,490.0,0,LAUNTD(4),2,2,1)
IF(IFLAG(3) .EQ. 0) GO TO 2
I = IFLAG(3) - IFLAG(3)/3
CALL CHAR(308.0,490.0,0,INSTL(1),4,2,1)
XL(1) = 308.0
XL(2) = 336.0
CALL LINE(XL,YL,2,0)
C ** PRINT SURFACE PRESSURE AND DENSITY
2 CALL CHAR(20.,475.,0,ISURL1,60,2,1)
CALL CHAR(318.,478.,0,IEXP3,1,2,1)
XL(1) = 20.
XL(2) = 76.
YL(1) = 473.
YL(2) = 473.
CALL LINE(XL,YL,2,0)
CALL CHAR(374.0,475.0,0,ISTAB,8,2,1)
CALL CHAR(466.0,475.0,0,ISTL(4),1,2,1)
XL(1) = 374.0
XL(2) = 422.0
CALL LINE(XL,YL,2,0)
CALL CODE
WRITE(IPRESD,2007) ALT(31)
CALL CHAR(428.0,475.0,0,IPRESD,6,2,1)
C ** PRINT SURFACE -- TOP LAYRT HEADER -- BOT LAYER HEADER (IF REQD)
I = 26
IF(IFLAG(2) .EQ. 1) I = 44
CALL CHAR(222.0,461.0,0,ISURT,1,2,1)
XL(1) = 222.
XL(2) = 278.
YL(1) = 459.
YL(2) = 459.
CALL LINE(XL,YL,2,0)
XL(1) = 302.
XL(2) = 374.
CALL LINE(XL,YL,2,0)
IF(IFLAG(2) .NE. 1) GO TO 8
XL(1) = 398.
XL(2) = 470.
CALL LINE(XL,YL,2,0)
C ** PRINT DRY TEMPERATURE
8 CALL CHAR(30.,450.,0,IDT,24,2,1)
C ** PRINT POTENTIAL TEMPERATURE
CALL CHAR(30.,440.,0,IPT,22,2,1)
C ** PRINT WIND SPEED
CALL CHAR(30.,430.,0,IWS,16,2,1)
C ** PRINT WIND DIRECTION
CALL CHAR(30.,420.,0,IWD,20,2,1)
C ** DRAW X AXIS
CALL LINE(X,Y,2,0)
C ** DRAW Y AXIS
CALL LINE(X(3),Y(3),2,0)
C ** DO LOOP TO ADD TIC MARKS FOR X AXIS
XTIC = 70.
XTIC2(1) = 88.
XTIC2(2) = 92.
XNUM1 = 62.
DO 10 I = 1,13
XTIC = XTIC + 30.
XTIC1(1) = XTIC
XTIC1(2) = XTIC
CALL LINE(XTIC1,XTIC2,2,0)
XTIC1(1) = XTIC1(1) + 15.
XTIC1(2) = XTIC1(2) + 15.
IF(I.EQ.13) GO TO 13
CALL LINE(XTIC1,XTIC2,2,0)
13 CONTINUE
XNUM1 = XNUM1 + 30.
IF(I.EQ.1) CALL CHAR(84.,80.,0,IMINUS,1,2,1)
CALL CHAR(XNUM1,80.,0,IXNUM(1),2,2,1)
10 CONTINUE
C ** DRAW TIC MARKS FOR WIND DIRECTION SCALE
XWD2(1) = 300.
XWD2(2) = 460.
YWD2(1) = 70.
YWD2(2) = 70.
YWD1(1) = 68.
YWD1(2) = 72.
CALL LINE(XWD2,YWD2,2,0)
CALL CHAR(310.,50.,0,1WD,20,2,1)

C ** PRINT LABELS FOR X-AXIS
CALL CHAR(100.,70.,0,1STL,24,2,1)

C ** DO LOOP TO ADD TIC MARKS TO Y-AXIS
YTIC = 58.
XTIC2(1) = 98.
XTIC2(2) = 102.
DO 20 I = 1,11
YTIC = YTIC + 32.
YTIC2(1) = YTIC
YTIC2(2) = YTIC
N = (I-1)*2 + 1
CALL CHAR(64.,YTIC2,0,IYNUM(N),4,2,1)
CALL LINE(XTIC2,YTIC2,2,0)

20 CONTINUE

C ** PRINT LABEL FOR Y-AXIS
YX = 360.
DO 30 I = 1,8
YX = YX - 20.
CALL CHAR(30.,YX,0,IALT(I),2,2,1)

30 CONTINUE

CALL CHAR(30.,YX-20.,0,IMET,3,2,1)

C ** THIS PRINTS SURFACE PRESSURE AND DENSITY VALUES
A = PRESS(1)
CALL CODE
WRITE(IPRESD,2007) A

2007 FORMAT(F6.1)
CALL CHAR(133.,475.,0,IPRESD,6,2,1)
A = SURED
CALL CODE
WRITE(IDENSD,2007) A
CALL CHAR(260.,475.,0,IDENSD,6,2,1)

C ** PRINT DRY TEMPERATURES
A = TEMP(1)
CALL CODE
WRITE(ISTP,2007) A
CALL CHAR(230.,450.,0,ISTP,6,2,1)

C ** PRINT POTENTIAL TEMPERATURES
A = PTEMP(1) - 273.15
CALL CODE
WRITE(ISPT,2007) A
CALL CHAR(230.,440.,0,ISPT,6,2,1)

DO 133 JJ=1,NUM
IF(ALT(JJ).GE.4000.) GO TO 3131
WSY(JJ) = (ALT(JJ))*0.08+ 90.
DTY(JJ) = (ALT(JJ))*0.08+ 90.
PTY(JJ) = (ALT(JJ))*0.08+ 90.
WDY(JJ) = (ALT(JJ))*0.08+ 90.
AWDIR(JJ) = IDIR(JJ)
APTEMP(JJ) = PTEMP(JJ) - 273.15

A-117
CONTINUE
JJ = NUM + 1
ILP = JJ - 1

CALL SUBROUTINE TO ROTATE WIND DIRECTION FOR PLOTTING

CALL WINDS(AWDIR, ILP, ISC)
DO 123 IK = 1, 9
N = (IK - 1) * 2 + 1
CALL LINE(XWD1(N), YWD1, 2, 0)
XBWD = XWD1(N) - 8.
YBWD = 60.
IF(ISC.EQ.1) CALL CHAR(XBWD, YBWD, 0, IWDL1(N), 4, 2, 1)
IF(ISC.EQ.2) CALL CHAR(XBWD, YBWD, 0, IWDL2(N), 4, 2, 1)
IF(ISC.EQ.3) CALL CHAR(XBWD, YBWD, 0, IWDL3(N), 4, 2, 1)
IF(ISC.EQ.4) CALL CHAR(XBWD, YBWD, 0, IWDL4(N), 4, 2, 1)

CONTINUE
DO 134 KK = 1, ILP
WSX(KK) = (SPEED(KK)) * 6. + 160.
DTX(KK) = (TEMP(KK)) * 6. + 160.
PTX(KK) = (APTEMP(KK)) * 6. + 160.
IF(TEMP(KK) .LT. -10.) DTX(KK) = 100.
IF(TEMP(KK) .GT. 50.) DTX(KK) = 460.
IF(APTEMP(KK) .LT. -10.) PTX(KK) = 100.
IF(APTEMP(KK) .GT. 50.) PTX(KK) = 460.
WDX(KK) = ABS(AWDIR(KK)) * .44444 + 300.

PRINT WIND SPEEDS
A = SPEED(1)
CALL CODE
WRITE(ISWS, 2007) A
CALL CHAR(230., 430., 0., ISWS, 6, 2, 1)

PRINT WIND DIRECTIONS
A = IDIR(1)
CALL CODE
WRITE(ISWD, 2007) A
CALL CHAR(230., 420., 0., ISWD, 6, 2, 1)

THIS PORTION DRAWS THE WIND SPEED LINE
CALL DLINE(WSX, WSY, ILP, 0, 8, 4)
XHT = WSY(ILP) + 3.
CALL CHAR(WSX(ILP), XHT, 0, ICRVT(1), 2, 2, 1)

THIS PORTION DRAWS THE DRY TEMPERATURE LINE
CALL LINE(DTX, DTY, ILP, 0)
XHT = DTY(ILP) - 5.0
CALL CHAR(DTX(ILP) + 4.0, XHT, 0, ICRVT(2), 2, 2, 1)

THIS PORTION DRAWS THE POTENTIAL TEMPERATURE LINE
CALL DLINE(PTX, PTY, ILP, 0, 4, 4)
XHT = PTY(ILP) + 3.
CALL CHAR(PTX(ILP), XHT, 0, ICRVT(3), 2, 2, 1)

THIS PORTION DRAWS THE WIND DIRECTION LINE

A-118
I1 = 1
DO 777 I=2,ILP
IF(AWDIR(I) .GE. 0.) GO TO 777
NUMP = I - I1
CALL DLINE(WDX(I1),WDX(I1),NUMP,0,4,8)
I1 = I
777 CONTINUE
NUMP = ILP - I1 + 1
CALL DLINE(WDX(I1),WDX(I1),NUMP,0,4,8)
XHT = WDX(ILP) - 5.0
CALL CHAR(WDX(ILP)+4.0,XHT,0,ICRYT(4),2,2,1)
C ** THIS PORTION DRAWS TIC MARKS AT VALID DATA POINT OF Y AXIS
DO 330 K=1,ILP
YDTIC(1) = ALT(K)*.08 + 90
YDTIC(2) = YDTIC(1)
CALL LINE(XDTIC,YDTIC,2,0)
330 CONTINUE
C
C DRAW THE CLOUD
C
YCLOUD = ALT(31)*.08 + 90.0
CALL CLOUD(250.0,YCLOUD)
C
WRITE OUT THE TOP OF THE SURFACE LAYER LINE
C
CALL MOVEM(JTOP,ILP,2,ITOP,318.0,TSURX,10)
C
IF REQUESTED, WRITE OUT THE BOTTOM OF THE SURFACE LAYER LINE
C
IF(IFLAG(2) .NE. 1)GO TO 444
CALL MOVEM(JBOT,ILP,1,BOT,414.0,BSURX,5)
C
CALL NGRAF TO REINITIALIZE PLASMASCOPE
C
CALL CLEAR
CONTINUE
444 CONTINUE
CALL NGRAF
STOP
END
SUBROUTINE WINDS(WD,NWD,ISC)
DIMENSION WD(1),ENDPT(4),NUMP(4)
EQUIVALENCE (J,LEAST)
DATA ENDPT/0.0,90.0,180.0,270.0/
DO 2 I=1,4
NUMP(I) = 0
2 WD2 = WD(I)
DO 8 I=2,NWD
WD1 = WD2
WD2 = WD(I)
8 J=1,4
\begin{verbatim}
C1 = WD1 - ENDPT(J)
IF(C1 .LT. 0.0)C1 = C1 + 360.0
C2 = WD2 - ENDPT(J)
IF(C2 .LT. 0.0)C2 = C2 + 360.0
IF(ABS(C1-C2) .LE. 180.0)GOTO 6
NUMUP(J) = NUMUP(J) + 1
6 CONTINUE
8 CONTINUE
ISC = 1
LEAST = NUMUP(1).
DO 12 I=2,4
IF(NUMUP(I) .GE. LEAST)GOTO 12
ISC = I
LEAST = NUMUP(I)
12 CONTINUE
DO 17 I=1,NWD
WD(I) = WD(I) - ENDPT(ISC)
IF(WD(I) .LT. 0.0)WD(I) = WD(I) + 360.0
17 CONTINUE
WD2 = WD(1)
DO 22 I=2,NWD
WD1 = WD2
WD2 = WD(I)
IF(ABS(WD1-WD2) .LE. 180.0)GOTO 22
WD(I) = - WD(I)
22 CONTINUE
RETURN
END

SUBROUTINE CLOUD(XP,YP)
COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSP,PI1,CLDHT,
   IDIR(31),IOP(3),LTIME,IDAY,MONTH(2),LTIME,LTIM,LDAY,
   LMON(2),LYEAR,LE,NP,F,PI2,PI43,PRESS(31),PTEMP(31),
   SIGH,RADEG,RAFPK,CLDRAD,II3,SAVER(30),SAVER(30),SIGA,
   SIGXO,SIGX,SPEED(31),SOR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
   TOPSUR,TWUPI,ASPD,VPAR(18),CTIME(31),DIST,YES,Y1,NUMRUN,
   YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
   (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
   (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
   (HEAT,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
   (PHCL,VPAR(13)),(PC0,VPAR(14)),(PC02,VPAR(15)),
   (P203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
2000 FORMAT (F6.1)
DIMENSION X(181),Y(181)
\end{verbatim}
RADIUS = GAMMAX * ALT(31) * 0.08
DO 7 I=1,181
7 Y(I) = RADIUS * SIN(0.01745329252 * FLOAT(2 * I)) + YP
CALL LINE(X,Y,181,0)
RADIUS = 5.0
X(1) = XP + RADIUS
X(2) = XP
X(3) = XP - RADIUS
X(4) = XP
X(5) = X(1)
Y(1) = YP
Y(2) = YP + RADIUS
Y(3) = YP
Y(4) = YP - RADIUS
Y(5) = Y(1)
CALL LINE(X,Y,5,0)
X(2) = XP - RADIUS
Y(2) = YP
CALL LINE(X,Y,2,0)
X(3) = XP
Y(3) = YP + RADIUS
CALL LINE(X(3),Y(3),2,0)
RETURN
END
SUBROUTINE MOVEM (JND, MAXJND, MINJND, LAB, XLABEL, XLINE, NLINE)

COMMON BLOCK

COMMON ALT(31), AL1, COMMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
  IDIR(31), I0PT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
  ISMON(2), ISYEYAR, IV2, JTOP, JBOF, LAUNTD(10), LTIME, LTIM, LDAY,
  LMOM(2), LYEAR, LU, NUM, PI, PI0VR2, PI43, PRESS(31), PTEMP(31),
  SIGHCL, RADDG, RATOMC, CLRAD, R2, R3, SAVEC(30), SAVER(30), SIGA,
  SIGO, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZO, SIGAP, S8, TEMP(31),
  TOPSUR, TOPOI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
  YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
  (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
  (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
  (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
  (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
  (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

2000 FORMAT (F6.1)
2001 FORMAT ("'13',0")
INTEGER QUES(13), ANS1, ANS2(2), ANS3(4), BLANKS(26)
DIMENSION LAB(1), XLINE(1), YLINE(2), JNDALT(3), JNDVAR(3, 4)
EQUIVALENCE (JNDYR1, JNDVAR(1, 1)), (JNDYR2, JNDVAR(1, 2))
(JNDVR3, JNDVAR(1, 3)), (JNDVR4, JNDVAR(1, 4))
DATA QUES/2HNO, 2HVE, 2H , 2H 0, 2HF, 2HSU, 2HRF, 2HC, 2HE, 2HLA, 2HYE, 2HR/
DATA ANS1/2HUP/, ANS2/2HDO, 2HWN/, ANS3/2HC0, 2HMT, 2HN, 2HUE/
DATA BLANKS/26*2H /
NEWJND = 0
1 YLINE(1) = ALT(JND) * 0.08 + 90.0
YLINE(2) = YLINE(1)
DO 4 I=1, NLIN
J = 2 * I - 1
4 CALL LINE(XLINE(J), YLINE, 2, 0)
Y = YLINE(1) + 2.0
CALL CHAR(460.0, Y, 0, LAB, 4, 2, 1)
Y = Y - 10.0
CALL CODE
WRITE (JNDALT, 2000) ALT(JND)
CALL CHAR(460.0, Y, 0, JNDALT, 6, 2, 1)
CALL CODE
WRITE (JNDVR1, 2000) TEMP(JND)
YLABEL = PTEMP(JND) - 273.15
CALL CODE
WRITE (JNDVR2, 2000) YLABEL
CALL CODE
WRITE (JNDVR3, 2000) SPEED(JND)
CALL CODE
WRITE (JNDVR4, 2001) IDIR(JND)
YLABEL = 450.0
DO 6 I=1, 4
CALL CHAR(XLABEL, YLABEL, 0, JNDVAR(1, 1), 6, 2, 1)
6 YLABEL = YLABEL - 10.0
RETURN
END
SUBROUTINE RWDIS(NAME, JJ)
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
. IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
. ISMONC(2), ISYEAR, IV2, ITOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
. LMON(2), LYEAR, LU, NUM, PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
. SICHCL, RADDEG, RATOMC, CLDRAD, R2, R3, SAVEA(30), SAEVE3(30), SIGA,
. SICKO, SIGX, SPEED (31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
. TOPSUR, TWDPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, YI, NUMRUN,
. YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
. (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
. (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
. (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
. (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
. (PAL203, VPAR(16)), (PHD, VPAR(17)), (GAMMAX, VPAR(18))
INTEGER ODCB(144), OBUF(669)
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1), ALT(1))
CALL OPEN(ODCB, IERR, NAME, 0)
IF(JJ.EQ.1) CALL WRITF(ODCB, IERR, OBUF, 669)
IF(JJ.EQ.0) CALL READF(ODCB, IERR, OBUF, 669)
CALL CLOSE(ODCB, IERR)
RETURN
END
END$
**CONCENTRATION AND DOSAGE PROGRAM -- A PROGRAM OF THE REED SERIES OF PROGRAMS**

**COMMON BLOCK**

```plaintext
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
      IDIR(31), IPOT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
      ISMON(2), ISYEAR, IV2, JTOP, JBOT, LAUHD(10), LTIME, LTIM, LDAY,
      LMON(2), LYEAR, LU, NUM_PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
      SIGHCL, RADDEG, RATOMC, CLDRAD, R2, R3, SAVSA(30), SAVER(30), SIGA,
      SIGX0, SIGX, SPEED(31), SQR2PI, SURDEN, SIG20, SIGAP, SB, TEMPS(31),
      TOPSUR, TWDPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
      YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETUN
LOGICAL ITIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
      (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
      (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
      (HEATN, VPAR(10)), (HEATOM, VPAR(11)), (HEATP, VPAR(12)),
      (PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
      (PALL203, VPAR(16)), (PN0, VPAR(17)), (GAMMAX, VPAR(18))
```

**OUTPUT FORMAT STATEMENTS**

```plaintext
200 FORMAT ("1"12X"CLOUD CONCENTRATIONS AND DOSAGES"/
           "DISTANCE"4X"CONCENTRATION"5X"DOSAGE"6X"
           "TIME AFTER LAUNCH(SEC)"
           "(METERS)"8X"(PPM)"8X"(PPM SEC)"8X"START"3X"FINISH")
201 FORMAT (1XF7.1, 8XF7.3, 8XF7.3, 9XF5.1, 3XF5.1)
202 FORMAT ("0***POINT OF MAXIMUM CONCENTRATION***"/
           "RANGE(M): "F8.1/
           "DIRECTION(DEG): "F5.1/
           "HEIGHT(M): "F6.1/
           "MAXIMUM CONCENTRATION(PPM): "F6.3)
203 FORMAT ("0***CONCENTRATIONS AND DOSAGES WITH 10 DEGREE "
           "UNCERTAINTIES***")
204 FORMAT ("0"5X"RANGE(M): "F7.1/
           "AZIMUTH(DEG): "F5.1/
           "MATERIAL "5X"CONCENTRATION(PPM)"11X"DOSAGE(PPM)")
205 FORMAT (415,12)
206 FORMA (7X3A2, 6XF8.3) +/- "F8.3, 4XF8.3" +/- "F8.3)
```

A-124
C TYPE AND DIMENSION STATEMENTS

LOGICAL IGRAF
INTEGER RISOP(3)
DIMENSION FACT(3),CMNPL(3),DMNPL(3),MATS(3,5),NAME(3),
       NAMEF(3),ILINE(32),IDATAF(10),IERS(32),
       DISTV(81),DOSV(81),CONCV(81)

DATA STATEMENTS
DATA IERS/32*2/
DATA HAHE/035228,2HEE,1EX/,NAMEF/2H?R,2HCO,2HNC/

DATA FACT/0.0,-0.174533,0.174533/
DATA MATS/2H,2HCH,2H,2H,2H C,2H0,2H,2H2,2H A,2H2,2H2,
       2H,2H M,2H0/
DATA RISOP/2HRI,2HSO,1HP/

CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING
PLASMASCOPE)

CALL GRAF(1)

READ COMMON DISK FILE #REDD

CALL RWDIS(NAME,0)

IF THIS IS A RESEARCH RUN, DETERMINE IF PLOTTING IS DESIRED

IF(IOPT(2) .EQ. 0)GO TO 55

CALL DREAD(NAMEF,2,ILINE)
CALL IERS(YPOS)
CALL CHARR(0.,YPOS,0,ILINE,42,3,0)
CALL CHARR(384.,YPOS,0,ILINE(25),8,3,0)
CALL CHARR(464.,YPOS,0,ILINE(30),6,0,0)
CALL IN(1,JTYPE,0.,0.,0,0,0,31,0,31,IX,1Y)
CALL CHARR(0.,YPOS,0,ILINE,64,0,0)

IF(IX .LE. 25)CALL CHARR(464.,YPOS,0,IERC,6,0,0)
IF(IX .GT. 25)CALL CHARR(384.,YPOS,0,IERC,8,0,0)
YPOS = YPOS - 32.

IF(IX .LE. 25)IGRAF = .TRUE.
IF(IX .GT. 25)IGRAF = .FALSE.

DO LOOP FOR CONCENTRATION AND DOSAGE CALCULATIONS

DIST - RANGE FROM STABILIZATION
DOSPK - DOSAGE
DOSMAX - MAXIMUM DOSAGE
CONCPK - CONCENTRATION
CONMAX - MAXIMUM CONCENTRATION

55 NUMV = 0
CONMAX = 0.0
DOSMAX = 0.0
ACTVOL = PI43 * CLDRAD * CLDRAD * CLDRAD
TOTVOL = ACTVOL
IF(lV2 .EQ. 1) ACTVOL = PI * (ALT(JTOP) + CLDRAD - ALT(31))**2 *
(2.0 * CLDRAD - ALT(JTOP) + ALT(31))/3.0
SIGHCL = SIGHCL * ACTVOL/TOTVOL
WRITE (6,200)
DO 59 I=0,20000,250
NUMV = NUMV + 1
DIST = I
DISTY(NUMV) = DIST
CALL DFEXP(JTOP,1000.0)
DOSP = SIGHCL * D1/(2PI2 * R2 * ASPD * SQRT(0.5 * R3))
DOSV(NUMV) = DOSP
CONCPK = DOSP * ASPD/(SQR2PI * SIGX)
CONCY(NUMV) = CONCPK
DOSMAX = AMAX1(DOSP,DOSMAX)
IF(lCONCPK .LE. CONMAX)GO TO 58
RATOC = DIST
CONMAX = CONCPK
SXMAX = SIGX
SYMAX = SIGY
58 IF(AMOD(DIST,1000.0) .NE. 0.0)GO TO 59
ARG1 = CRITME(31) + (DIST - AL1)/ASPD
ARG2 = CRITME(31) + (DIST + AL1)/ASPD
WRITE (6,201) DIST,CONCPK,DOSP,ARG1,ARG2
59 CONTINUE
IF REQUESTED, PLOT THE CENTERLINE DOSAGE AND CONCENTRATION VALUES
ARG1 = ALOGT(DOSMAX)
IEXP = ARG1
IF(ARG1 .LT. 0.0)IEXP = IEXP - 1
IEXP = - IEXP
ARG1 = ALOGT(CONMAX)
IEXPC = ARG1
IF(ARG1 .LT. 0.0) IEXPC = IEXPC - 1
IEXPC = - IEXPC
IF(.NOT. IGRAF) GO TO 61
CALL CPLOT(DISTV, DOSV, CONC1, NUMV, IEXPD, IEXPC)

CALCULATE AND WRITE OUT THE POINT OF MAXIMUM CONCENTRATION

61 ARG1 = DEGRAD * ADIR
DIST = RATOMC * COS(ARG1)
Y1 = RATOMC * SIN(ARG1)

DO 62 I=2, JTOP
IF(CLDHT .LE. ALT(I)) GO TO 63
62 CONTINUE
I = JTOP

63 IM1 = I - 1
RANGSR = SAVING(IM1) + (CLDHT - ALT(IM1)) * 
(SAVING(I) - SAVING(IM1))/(ALT(I) - ALT(IM1))

ARG1 = SAVING(I) - SAVING(IM1)
IF(ABS(ARG1) .LT. 180.0) GO TO 66
IF(ARG1 .GT. 0.0) GO TO 65
SAVEAI = SAVING(I) + 360.0
GO TO 66
65 SAVING(IM1) = SAVING(IM1) + 360.0

66 AZCS = SAVING(IM1) + (CLDHT - ALT(IM1)) * 
(SAVING(I) - SAVING(IM1)) / 
(ALT(I) - ALT(IM1))
IF(AZCS .GE. 360.0) AZCS = AZCS - 360.0

ARG1 = DEGRAD * AZCS
X2 = RANGSR * COS(ARG1)
Y2 = RANGSR * SIN(ARG1)
X = DIST + X2
Y = Y1 + Y2

RNGE = SQRT(X * X + Y * Y)
DIR = RADDEG * ATAN2(Y, X)
IF(DIR .LT. 0.0) DIR = DIR + 360.0
WRITE (6, 202) RNGE, DIR, ZB, COMMAX

IF THIS IS A PRODUCTION RUN, SKIP THE OFF CENTER CONCENTRATION
SECTION AND THE CALL OF PROGRAM RISOP -- IF PLOTTING WAS NOT
REQUESTED, JUST SKIP THE OFF CENTER CONCENTRATION SECTION

IF(IGRAF) GO TO 68
IF(IOPT(2) .EQ. 0) GO TO 88
GO TO 81
OFF CENTER CONCENTRATIONS SECTION

68 CALL LABEL(IEXPD, IEXPQ)

ARE OFF CENTER CONCENTRATIONS DESIRED?

CALL DREAD(NAMEF, 3, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0., ILINE, 30, 3, 0)
CALL CHAR(384., YPOS, 0., ILINE(25), 8, 0, 0)
CALL CHAR(464., YPOS, 0., ILINE(30), 6, 3, 0)
CALL IN(1, JTYPE, 0., 0., 0., 0., 0., 0., 31, 0, 31, IX, IY)
CALL CHAR(0., YPOS, 0., ILINE, 64, 0, 0)
IF(IX .LE. 25) CALL CHAR(464., YPOS, 0., IERS, 6, 0, 0)
IF(IX .GT. 25) CALL CHAR(384., YPOS, 0., IERS, 8, 0, 0)
YPOS = YPOS - 32.
IF(IX .GT. 25) GO TO 81

OFF CENTER CONCENTRATIONS ARE DESIRED

WRITE (6, 203)
CALL ORGIN(ISET, 3, IYSET)

ARG1 = 0.0
IF(ADIR .GT. 180.0) ARG1 = 360.0
BETAF = DEGRAD * (180.0 + ARG1 - ADIR)

ARG1 = 0.0
IF(AZCS .GT. 180.0) ARG1 = 360.0
BETAS = DEGRAD * (180.0 + ARG1 - AZCS)
XP = RANGSR * COS(BETAS)
YP = RANGSR * SIN(BETAS)

ITER = 0

LOOP ON OFF CENTER CONCENTRATION REQUESTS

CALL DREAD(NAMEF, 5, ILINE)
CALL LERS(YPOS)
CALL CHAR(0., YPOS, 0., ILINE, 64, 0, 0)
YPOS = YPOS - 16.
71 ITER = ITER + 1

READ IN AND WRITE OUT THE RANGE AND AZIMUTH FOR THE
OFF CENTER CONCENTRATION CALCULATION -- ENTERING A RANGE OF 0
TERMINATES THE PROCEDURE

IF(YPOS .LT. 48.) YPOS = 458.
CALL DREAD(NAMEF, 6, ILINE)
CALL LERS(YPOS)
CALL CHAR(0, YPOS, 0, ILINE, 64, 0, 0)
NIN = 7
CALL BLANK(IDATAF, 10)
CALL IN(0, JTYPE, 112, YPOS, 0, IDATAF, NIN, 0, 31, 0, 31, IX, IY)
CALL CODE
READ (IDATAF, *) RP
IF (RP .LE. 0.0) GO TO 78
NIN = 7
CALL BLANK(IDATAF, 10)
CALL IN(0, JTYPE, 272, YPOS, 0, IDATAF, NIN, 0, 31, 0, 31, IX, IY)
CALL CODE
READ (IDATAF, *) AZP
YPOS = YPOS - 16.
IF (YPOS .LT. 48.0) YPOS = 458.0
WRITE (6, 204) RP, AZP

ARG1 = 0.0
IF (AZP .GT. 180.0) ARG1 = 360.0
AP = DEGRAD * (180.0 + ARG1 - AZP)
XS = RP * COS(AP)
YS = RP * SIN(AP)

ON THE PLOTTER, WRITE OUT AN ASTERISK AND THE ITERATION
NUMBER AT THE LOCATION WHERE THE OFF CENTER CONCENTRATION
CALCULATION IS DESIRED

IX = IXSET + 0.2631 * XS
IY = IYSET + 0.3545 * YS
WRITE (12) -1, 1, IX, IY
CALL SYMBL(100, 125, 1H*)
IX = IX + 75
WRITE (12) -1, 1, IX, IY
WRITE (12, 205) 100, 0, 0, 125, ITER

CALCULATE THE CONCENTRATIONS AND DOSAGES AT THIS POINT PLUS
10 DEGREES UNCERTAINTIES ON EITHER SIDE

XHAT = XS - XP
YHAT = YS - YP

DO 74 I = 1, 3
ARG1 = BETAF - FACT(I)
Y = - XHAT * SIN(ARG1) + YHAT * COS(ARG1)
DIST = XHAT * COS(ARG1) + YHAT * SIN(ARG1)
CALL DFEXP(JTOP, 1000.0)
DOS = SIGHCL * E1 * EXP(- Y * Y/(2.0 * R2 * R2))/
     (TWOPI * R2 * ASPD * SQRT(0.5 * R3))
CONC = DOS * ASPD/(SQR2PI * SIGX)
THIS SUBROUTINE PLOTS THE DOSAGE AND CONCENTRATION CENTERLINE CURVES

COMMON BLOCK

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
        IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTDAY, 
        ISMON(2), ISYEAR, IV2, JTOP, JBOT, LAUHTD(10), ITIME, LTIM, LDAY, 
        LMON(2), LYEAR, LU2, NUM, PI, PI43, PRESS(31), PI43, PRESS(31), 
        SIGHCL, RADDEC, RATOC, CLDRAD, R2, R3, SAVEA(30), SAVCR(30), SIGA, 
        SIGX0, SIGX, SPEED(31), SQR2P1, SURDEN, SIGZ0, SIGAP, S8, TEMP(31), 
        TOPSUR, TWOP, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN, 
        YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETURN

LOGICAL LTME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)), 
        (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
        (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)), 
        (HEATHN, VPAR(10)), (HEATHN, VPAR(11)), (HEAT, VPAR(12)),
        (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
        (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))

DIMENSION STATEHEHT

DIMENSION DISTY(1), DOSY(1), CONCVY(1)

CALCULATE PLOTTING FACTORS

FDIST = 9295.0/30000.0

A-132
FDOS = 8231.0 * 10.0**(IEXPD - 1)
FCONC = 8231.0 * 10.0**(IEXPC - 1)

PLOT THE DOSAGE CENTERLINE CURVE

DO 7 I=1,NUMV
  IX = DISTV(I) * FDIST + 725.0
  IY = DOSV(I) * FDOS + 1040.0
  WRITE (12) -1,1,IX,IY
  7 CALL SYMBL(100,100,25400B)

PLOT THE CONCENTRATION CENTERLINE CURVE

DO 16 I=1,NUMV
  J = 1/I
  J = 1 - 2 * J
  IX = DISTV(I) * FDIST + 725.0
  IY = CONCV(I) * FCONC + 1040.0
  16 WRITE (12) J,1,IX,IY

RETURN TO RCONC

RETURN

END OF CPLOT

END SUBROUTINE LABEL(IEXPD,IEXPC)

******************************************************************************

THIS SUBROUTINE LABELS THE CONCENTRATION AND DOSAGE CENTERLINE PLOTS

******************************************************************************

COMMON BLOCK

COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSP,K,E1,CLDHT,
        IDIR(31),IPT(3),ITIME,IDAY,MONT(2),IYEAR,ISTIM,ISDAY,
        ISMONC2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
        LMON(2),LYEAR,LU,NUM,P1,POPVR2,P143,PRESS(31),PTEMP(31),
        SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVE(30),SIGA,
        SIGX0,SIGX,SPED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
        TOPSUR,TUOP1,ASD,VPAR(18),CRTIME(31),DIST,YES,YI,NUMRUN,
        YP0S,IFLAG(5),ZB,ZZ,REFLC,IRETRN

LOGICAL LTME

INTEGER YES

EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
(QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
(AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
(HEATH, VPAR(10)), (HEATH, VPAR(11)), (HEATA, VPAR(12)),
(PHCL, VPAR(13)), (PC0, VPAR(14)), (PC02, VPAR(15)),
(PAL203, VPAR(16)), (PNO, VPAR(17)), (GANMAX, VPAR(18))

OUTPUT FORMAT STATEMENTS

200 FORMAT (4I5, 12)
201 FORMAT (4I5, F5.0)
202 FORMAT (4I5, F5.2)
203 FORMAT (4I5, 14, 1X, A1, A2, 2XI2, 1XA2, A1, 1XI4)
204 FORMAT (4I5, 14, "C*A2, 2XI2, 1XA2, A1, 1XI4")
205 FORMAT (4I5, 14, 1XR1, A2, 2XI2, 1XA2, A1, 1XI4)

LABEL THE PLOT

I = - IEXPC
WRITE (12) -1, 1, 300, 5000
WRITE (12, 200) 0, 150, -100, 0, I
I = - IEXPD
WRITE (12) -1, 1, 300, 6500
WRITE (12, 200) 0, 150, -100, 0, I
WRITE (12) -1, 1, 3700, 8950
WRITE (12, 201) 125, 0, 0, 125, CLDHT
WRITE (12) -1, 1, 3700, 8745
WRITE (12, 201) 125, 0, 0, 125, CRTIME(31)
WRITE (12) -1, 1, 3700, 8540
WRITE (12, 202) 125, 0, 0, 125, CONMAX
WRITE (12) -1, 1, 3700, 8335
WRITE (12, 201) 125, 0, 0, 125, ALT(JTOP)
WRITE (12) -1, 1, 3700, 8130
WRITE (12, 201) 125, 0, 0, 125, ZB
WRITE (12) -1, 1, 3700, 7925
WRITE (12, 201) 125, 0, 0, 125, ZB
IF(IOPT(1) .EQ. 1) GO TO 4
WRITE (12) -1, 1, 5625, 8980
WRITE (12) 1, 1, 6125, 8980
GO TO 7
4 WRITE (12) -1, 1, 5025, 8980
WRITE (12) 1, 1, 5525, 8980
WRITE (12) -1, 1, 5725, 8950
WRITE (12, 203) 125, 0, 0, 125, ISTIM, IFLAG(4), LAUNTD4, ISDAY, ISMON,
ISYEAR
7 WRITE (12) -1, 1, 5725, 8695
WRITE (12, 204) 125, 0, 0, 125, ITIME, LAUNTD4, IDAY, MONTH, IYEAR
WRITE (12) -1, 1, 5725, 8490
IF(LTIME) WRITE (12, 205) 125, 0, 0, 125, LTIM, LAUNTD3, LAUNTD4, LDAY,
LMON, LYEAR

A-134
RETURN TO RCONC
RETURN
END OF LABEL

END

SUBROUTINE DFEXP(J, CONC)

*******************************************************************************

THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS

J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER
CONC - CONCENTRATION TO BE TESTED

*******************************************************************************

COMMON BLOCK

COMMON ALT(31), AL1, CONMAK, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
   IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTEM, ISDAY,
   ISMOR(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
   LMON(2), LYEAR, LU, NUM, PI, P10V2, P143, PRES(31), PTEMP(31),
   SIGHCL, RADDEG, RATOMC, CLDRAD, R2, R3, SVEA(30), SVEA(30), SIGA,
   SIGXO, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
   TOPSUR, TWOP, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
   YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN

LOGICAL LTIME
INTEGER YES

EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEAT, VPAR(12)),
   (PHCL, VPAR(13)), (PCD, VPAR(14)), (PCO2, VPAR(15)),
   (PALO3, VPAR(16)), (PNO, VPAR(17)), (CGMMAK, VPAR(18))

CALCULATE SIGMA Z

SIGZ = DIST * SIGAP + SIGZ0/1.28
R3 = 2.0 * SIGZ * SIGZ

CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION

TWOI = 2.0
ZT = ALT(J)
TEMP2 = CLDHT - ZZ
TEMP3 = CLDHT - 2.0 * ZZ + ZZ

A-135
$E1 = \exp(-TEMP1 \times TEMP2/R3) + \exp(-TEMP3 \times TEMP3/R3)$

4 $TEMP1 = \text{W01} \times (ZT - ZB)$
$TEXPSM = EL$
$TEXP = (TEMP1 - TEMP2)**2/R3$

IF($TEXP \leq 120.0$)$EL = EL + \exp(-TEXP)$
$TEXP = (TEMP1 + TEMP2)**2/R3$

IF($TEXP \leq 120.0$)$EL = EL + \exp(-TEXP)$
$TEXP = (TEMP1 - TEMP3)**2/R3$

IF($TEXP \leq 120.0$)$EL = EL + \exp(-TEXP)$
$TEXP = (TEMP1 + TEMP3)**2/R3$

IF($E1 \neq TEXPSM$)$GO TO 7$
$W01 = W01 + 2.0$
$GO TO 4$

7 $E1 = \text{REFLEC} \times E1$

CALCULATE SIGMA Y

$S8 = \text{DIST} \times \text{SIGAP} + \text{SIGX0}$
$R2 = \sqrt{S8 \times S8 + (0.0040589 \times \text{FLOAT(IDIR(J) - IDIR(1))})^2}$

CALCULATE CLOUD LENGTH

$TEMPl = \text{SPEED}(J) - \text{SPEED}(1)$
$AL1 = 0.28 \times TEMPl \times DIST/ASPD$

IF($TEMPl \geq 0.0$)$GO TO 11$

IF($PTEMP(J)-PTEMP(1) \lt 0.0$)$AL1 = 0.0$

CALCULATE SIGMA X

11 $SIGX = \sqrt{(AL1/4.3)**2 + SIGX0 \times SIGX0}$

IF($\text{CONC}=1000.0$), DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN TO THE CALLING PROGRAM

IF($\text{CONC} \neq 1000.0$)$RETURN$

CALCULATE CROSS WIND DISTANCE

$Y1 = -2.0 \times R2 \times R2 \times \text{ALOG}(15.7496 \times \text{CONC} \times SIGX \times R2 \times SIGZ/(\text{SIGHCL} \times E1))$

$Y1 = \sqrt{\text{AMAX1}(Y1,0.0)}$

RETURN TO THE CALLING PROGRAM

RETURN
END OF DFEXP

END

SUBROUTINE ORGIN(IX0, IYO)

C*****************************************************************************************

CTHIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
FOR THE COMPLEX AND MAP SELECTED

C******************************************************************************************

C COMMON BLOCK

COMMON ALT(31), AL1, CONMAX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLDHT,
. IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTRIM, ISDAY,
. ISMOL(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIM, LTIM, LDAY,
. LMON(2), LYEAR, LU, NUM, PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
. SIGHC, RADDG, RATOMC, CLDRAD, R2, R3, SAVR(30), SAVR(30), SIGA,
. SIGX0, SIGK, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, S8, TEMP(31),
. TOPSU, TVPOI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
. YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIM
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
. (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
. (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
. (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATN, VPAR(12)),
. (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
. (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))
DIMENSION ILINE(32), IDATAF(10), IERS(32), IMAPL(48), NAMEF(3)

C INPUT FORMAT STATEMENT

100 FORMAT (I2, 1XA1)

C OUTPUT FORMAT STATEMENT

C TYPE AND DIMENSION STATEMENTS

LOGICAL NOTIST
DIMENSION IX(8), IY(8)

C DATA STATEMENTS

DATA IERS/32*2H/
DATA NAMEF/2H?R, 2HIS, 2HP/
DATA IMAPL/2H40, 2H, S, 2HEA, 2H M, 2HAP, 2H

A-137
DATA NOT1ST/.FALSE./, LCHAR/IHL/
DATA IX/5450, 5411, 4830, 4825, 8750, 8730, 4100, 4100/
DATA IY/2630, 8243, 2465, 8050, 2990, 8600, 1700, 7300/

IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
COORDINATES, I, AGAIN

IF(NOT1ST)GO TO 7

THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
AND THE DESIRED MAP, i.e. SEA OR LAND

NOT1ST = .TRUE.
CALL DREAD(NAMEF,7,ILINE)
CALL LERS(YPOS)
CALL CHARO.,YPOS,0,ILINE,64.,0,0)
YPOS = YPOS - 16.
IF(YPOS.LT.0.) YPOS = 458.
IF(IOPT(3).EQ.1) CALL DREAD(NAMEF,8,ILINE)
IF(IOPT(3).EQ.2) CALL DREAD(NAMEF,9,ILINE)
IF(IOPT(3).EQ.0) CALL DREAD(NAMEF,10,ILINE)
CALL LERS(YPOS)
CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
CALL IN(1,JTYPE,0.,0.,0.,0.,0.,0.,0.,0.,0.,31,0.,31,IXC,IYC)
CALL CHARO.,YPOS,0,IERs,64.,0,0)
CALL CHAR(200.,YPOS+16.,0,iers,25.,0,0)
IF(IJC.LT.6.AND.IOPT(3).EQ.1) I=1
IF(IJC.GT.5.AND.IJC.LT.12.AND.IOPT(3).EQ.1) I=2
IF(IJC.GT.11.AND.IJC.LT.18.AND.IOPT(3).EQ.1) I=3
IF(IJC.GT.17.AND.IOPT(3).EQ.1) I=4
IF(IJC.LT.6.AND.IOPT(3).EQ.2) I=5
IF(IJC.GE.6.AND.IOPT(3).EQ.2) I=6
IF(IJC.LT.6.AND.IOPT(3).EQ.0) I=7
IF(IJC.GE.6.AND.IOPT(3).EQ.0) I=8
IMP = (I - 1)*6 + 1
CALL CHAR(208.,YPOS+16.,0,IMAPL(IMP),12,0,0)
YPOS = YPOS - 16.
IF(YPOS.LT.48.) YPOS = 458.

SET THE COORDINATES BASED ON THE INDEX I
7 IX0 = IX(I)
    IY0 = IY(I)

C
RETURN TO THE CALLING PROGRAM
C
RETURN

C
END OF ORGIN
C
END

SUBROUTINE SYMIX(IX, IY, ISYMB)
IX = IX/2
IY = IY/2
WRITE(12) -1, -1, IX, IY
WRITE(12, 100) IX, IY, ISYMB
100 FORMAT(415, A1, I H_)
IY = -IY
WRITE(12) -1, -1, IX, IY
RETURN
END

SUBROUTINE DREAD(NAMEE, LNUM, ILINE)
DIMENSION NAMEE(3), IDC5B(276), IBUF(40), ILINE(32), IPAR(5)
CALL RMPAR(IPAR)
LU = IPAR(1)
CALL OPEN(IDC5B, IERR, NAMEE, 0)
LOOP = LNUM - 1
DO 10 I=1, LOOP
CALL BLANK(IBUF, 40)
CALL READF(IDC5B, IERR, IBUF)
10 CONTINUE
CALL BLANK(IBUF, 40)
CALL READF(IDC5B, IERR, IBUF)
CALL CODE
READ(IBUF, 100) (ILINE(I), I=1, 32)
100 FORMAT(32A2)
CALL CLOSE(IDC5B, IERR)
RETURN
END

SUBROUTINE BLANK(IBUF, II)
DIMENSION IBUF(40)
DATA IBLK/2H /
DO 10 I=1, II
10 IBUF(I) = IBLK
RETURN
END

SUBROUTINE IERS(YPOS)
DIMENSION IERS(32)
DATA IERS/32*2H /
IF(YPOS.LE.48) YPOS = 458.0
CALL CHAI(0., YPOS, 0, IERS, 64, 0, 0)
CALL CHAR(0, YPOS-16, 0, IERS, 64, 0, 0)
RETURN
END
END$
FTN4, L

PROGRAM RISOP

*------------------------------------------------------------------------*
* ISOPLETH PLOTTING PROGRAM -- A PROGRAM IN THE REED SERIES            *
* OF PROGRAMS                                                          *
*------------------------------------------------------------------------*

COMMON BLOCK

COMMON ALT(31), AL1, CONMX, CONCPK, DEGRAD, ADIR, DOSPK, E1, CLOHT,
   IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, ISTIM, ISDAY,
   ISNOW(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
   LMON(2), LYEAR, LU, NUM, PI, PIOVR2, PI43, PRESS(31), PTEMP(31),
   SIGHCL, RADDEG, RATOMC, CLRAD, R2, R3, SAVA(30), SAVR(30), SIGA,
   SIGK, SIGX, SPEED(31), SQR2PI, SURDEN, SIGZ0, SIGAP, SB, TEMP(31),
   TOPSUR, TwpI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
   YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
   (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
   (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
   (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
   (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
   (PAL203, VPAR(16)), (PNO, VPAR(17)), (GAMMA, VPAR(18))

OUTPUT FORMAT STATEMENTS

200 FORMAT ("120X"CLOUD LOCATION AND DIMENSIONS",
  " TIME FROM CLOUD STABILIZATION"5X"RANGE"5X"AZIMUTH",
  8X"DIAMETERS" (METERS)",
  11X"MINUTES"14X"METERS"4X"DEG"6X"CROSS WIND",
  4X"ALONG WIND")
201 FORMAT (12XF6.2, 16XF8.1, 4XF5.1, 7XF7.1, 7XF7.1)
202 FORMAT (4I5, I4" C"A2, 2XI2, 1XA2, A1, 1XI4)
203 FORMAT (4I5, A1)
204 FORMAT (4I5, F5.2")
205 FORMAT (4I5", "F5.2")
206 FORMAT (F6.3)
207 FORMAT (I1)
208 FORMAT (4I5, I4, 1XR1, A2, 2XI2, 1XA2, A1, 1XI4)

TYPE AND DIMENSION STATEMENTS

LOGICAL DFALTC
DIMENSION CONC(10), NAME(3), NAMEF(3), ILINE(32), IDATAF(10),
A-141
CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING PLASMASCOPE)

CALL GRAF(1)

READ COMMON DISK FILE

CALL RWDIS(NAME,0)

DETERMINE THE ORIGIN ON THE MAP FOR THIS PLOT AND MOVE THE PEn THERE

CALL ORGIN(IX0,IYO)
WRITE (12) -1,1,IX0,IYO

DETERMINE THE INDEX IN THE ALTITUDE DATA ARRAY THAT HAS THAT ALTITUDE JUST LOWER THAN THE EFFECTIVE CLOUD HEIGHT, CLDHT

DO 4 I=2,JTOP
IF(CLDHT .GT. ALT(I)) GO TO 4
ICLDHT = I - 1
GO TO 5
4 CONTINUE
ICLDHT = JTOP

DRAW THE LINE DEPICTING CLOUD MOVEMENT ALONG THE GROUND AS FAR AS THE CLOUD STABILIZATION POINT

5 X = 0.0
   Y = 0.0
   DO 9 I=2,ICLDHT
      IM1 = I - 1
      RANGE = 0.5 * (CRTIME(I) - CRTIME(IM1)) * (SPEED(I) + SPEED(IM1))
      DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1))
      IF(IABS(IDIR(I) - IDIR(IM1)) .GT. 180) DIR = DIR - 180.0
      IF(DIR .LT. 0.0) DIR = DIR + 360.0
      DIR = DEGRAD * (360.0 - DIR)
      X = X + RANGE * COS(DIR)
      Y = Y + RANGE * SIN(DIR)
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IYO
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999) GO TO 11
9 WRITE (12) 1,1,IX,IY

MAKE THE CALCULATIONS NECESSARY TO WRITE OUT THE CLOUD LOCATION AND DIMENSIONS
C

11 ALT1 = 0.5 * (CLDHT + ALT(ICLDHT))
   ICLDP1 = ICLDHT + 1
   ARG1 = ALT(ICLDP1) - ALT(ICLDHT)
   ARG2 = (CLDHT - ALT(ICLDHT))/ARG1
   SPCEXT = SPEED(ICLDHT) + (SPEED(ICLDP1) - SPEED(ICLDHT)) * ARG2
   RANGE = SPCEXT * (CRTIME(ICLDP1) - CRTIME(ICLDHT)) * ARG2
   IF(IABS(IDIR(ICLDP1) - IDIR(ICLDHT)) .LT. 180) GO TO 14
   IF(IDIR(ICLDP1) .LT. 180) IDIR(ICLDHT) = IDIR(ICLDHT) + 360
   IF(IDIR(ICLDHT) .LT. 180) IDIR(ICLDHT) = IDIR(ICLDHT) + 360
   DIR = FLOAT(IDIR(ICLDHT)) + (ALT1 - ALT(ICLDHT)) * 
       FLOAT(IDIR(ICLDP1) - IDIR(ICLDHT))/ARG1
   IF(DIR .GT. 360.0) DIR = DIR - 360.0
   IF(DIR .GT. 180.0) GO TO 17
   DIR = DIR + 180.0
   GO TO 18.
17 DIR = DIR - 180.0
18 DIR = 180.0 - DIR
   ARG1 = DEGRAD * DIR
   X = X + RANGE * COS(ARG1)
   Y = Y + RANGE * SIN(ARG1)
   R = SQRT(X * X + Y * Y)
   DELR = 300.0 * ASPD
C
   DACRS = 4.30 * SIGX0
   DALNG = 4.30 * SIGX0
C
   ARG1 = 180.0
   IF(DIR .GT. 180.0) ARG1 = 540.0
   AZ = ARG1 - DIR
C
   ARG1 = 180.0
   IF(AZ .GT. 180.0) ARG1 = 540.0
   DAZ = ARG1 - AZ
   ARG1 = DEGRAD * DAZ
   DELX = DELR * COS(ARG1)
   DELY = DELR * SIN(ARG1)
C
   DELU = ABS(SPEED(ICLDHT) - SPEED(I))
C
   DELTH = IDIR(JTOP) - IDIR(I)
C
   TIM = 0.0
   R1 = 0.0
   XC = X
   YC = Y
   TXL = 0.28 * DELU/ASPD
   SIGX02 = SIGX0 * SIGX0
   S82 = S8 * S8
   WRITE (6,200)
DO 22 I=1,13
WRITE (6,201) TIM,R,AZ,DACRS,DALNG
TIM = TIM + 5.0
R1 = R1 + DELR
XL = R1 * TXL
SIGX = SQRT((XL/4.30)**2 + SIGX0**2)
DACRS = 4.30 * SIGX
SIGY = SQRT(S82 + (0.0040589 - 3.0 * DELTH * R1**2)
DALNG = 4.30 * SIGY
XC = XC + DELX
YC = YC + DELY
R = SQRT(XC * XC + YC * YC)
22  AZ = 180.0 - RADDEG * ATAN2(YC,XC)

C LABEL THE CLOUD STABILIZATION POINT WITH A +
IX = INT(0.2631 * X) + IX0
IY = INT(0.3545 * Y) + IY0
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 77
IXX = IX
IYY = IY
WRITE (12) 1,1,IX,IY
CALL SYMBLC(150,150,1H+)

C LABEL THE POINT OF MAXIMUM CONCENTRATION WITH A @

DIR = DEGRAD * (180.0 - ADIR)
CDIR = COS(DIR)
SDIR = SIN(DIR)
IX1 = INT(0.2631 * (X + RATOMC * CDIR)) + IX0
IY1 = INT(0.3545 * (Y + RATOMC * SDIR)) + IY0
WRITE (12) -1,1,IX1,IY1
CALL SYMBLC(150,150,1H0)

C DRAW THE LINE OF CLOUD MOVEMENT ALONG THE GROUND FROM
THE CLOUD STABILIZATION POINT ON

WRITE (12) -1,1,IXX,IY
RANGE = 1000.0
27  X = X + RANGE * CDIR
Y = Y + RANGE * SDIR
IX = INT(0.2631 * X) + IX0
IY = INT(0.3545 * Y) + IY0
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 29
WRITE (12) 1,1,IX,IY
GO TO 27
29  WRITE (12) -1,1,IXX,IY

C ARE DEFAULT CONCENTRATION VALUES GOING TO BE USED
C FOR THE PLOTS

IF(YPOS.LT.48.) YPOS = 458.
CALL DREAD(NAMEF,2,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
YPOS = YPOS - 16.
CALL DREAD(NAMEF,3,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
YPOS = YPOS - 32.

C YES -- SET UP THE DEFAULT VALUES

CONC(1) = 0.1 * CONHAX
CONC(2) = 0.5 * CONHAX
CONC(3) = 0.75 * CONHAX
CONC(4) = -1.0
CALL CODE
WRITE (IDATAF,206) CONC(1)
CALL CHAR(440.,YPOS+48.,0,IDATAF,5,0,0)
CALL CODE
WRITE (IDATAF,206) CONC(2)
CALL CHAR(120.,YPOS+32.,0,IDATAF,5,0,0)
CALL CODE
WRITE (IDATAF,206) CONC(3)
CALL CHAR(256.,YPOS+32.,0,IDATAF,5,0,0)
CALL DREAD(NAMEF,4,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,46,3,0)
CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
CALL INC1,TYPE,0.,0.,0.,0.,0.,0.,0,31,0,31,IX,IY)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
IF(IX .LE. 25) CALL CHAR(464.,YPOS,0,IERS,6,0,0)
IF(IX .GT. 25) CALL CHAR(384.,YPOS,0,IERS,8,0,0)
YPOS = YPOS - 32.
IF(YPOS .LT. 64.0) YPOS = 458.0
DFALTC = .FALSE.
IF(IX .LT. 26) DFALTC = .TRUE.

C DO LOOP OVER THE 10 POSSIBLE CONCENTRATION VALUES FOR THE PLOTS

IF(DFALTC) GO TO 35
CALL DREAD(NAMEF,5,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
YPOS = YPOS - 32.
IF(YPOS .LE. 64) YPOS=458.
35 DO 59 I=1,10
IF DEFAULT CONCENTRATION VALUES ARE NOT BEING USED,
READ IN THE VALUE FOR THIS PLOT

IF(DFALT.C)GO TO 37
CALL DREAD(NAMEF,6,ILINE)
CALL LERS(YPOS)
CALL CHAR(0.,YPOS,0,ILINE,17,3,0)
CALL CODE
WRITE (IDX,207) I
CALL CHAR(111.,YPOS,0,IDX,1,3,0)
NIN = 9
CALL BLANK(IDATAF,10)
CALL IN(0.,JTYPE,144.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
CALL CODE
READ (IDATAF,* ) CONC(I)
CALL CHAR(0.,YPOS,0,ILINE,17,0,0)
CALL CHAR(111.,YPOS,0,IDX,1,0,0)
YPOS = YPOS - 16.
IF(YPOS .GT. 48.0)YPOS = 458.0
37 IF(CONC(I) .LT. 0.0)GO TO 61

ITERATE TO FIND THE LOCATION OF THIS CONCENTRATION
ON THE PLOT

DIST = 0.0
DINC = 1000.0

41 CALL DFEXP(JTOP,CONC(I))
IF(Y1 .GT. 0.0)GO TO 42
DIST = DIST + DINC
GO TO 41

42 IF(DINC .LE. 100.0)GO TO 43
DIST = DIST - 900.0
DINC = 100.0
GO TO 41

43 IF(DINC .LE. 10.0)GO TO 44
DIST = DIST - 90.0
DINC = 10.0
GO TO 41

PLOT OUT THE CONCENTRATION LINE ON BOTH SIDES

44 DIST = DIST - 10.0
IX = INT(0.2631*DIST * CDIR) + IXX
IY = INT(0.3545 * DIST * SDIR) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
NUMA = 1

A-146
IXA(NUMA) = IX
IYA(NUMA) = IY
NUMB = 1
IXB(NUMB) = IX
IYB(NUMB) = IY
C
DIST = DIST + 10.0
IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
NUMA = 2
IXA(NUMA) = IX
IYA(NUMA) = IY
C
IX = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
NUMB = 2
IXB(NUMB) = IX
IYB(NUMB) = IY
C
46 DIST = DIST + 500.0
CALL DFEXP(JTOP,CONC(I))
IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
NUMA = NUMA + 1
IXA(NUMA) = IX
IYA(NUMA) = IY
C
IF(Y1 .GT. 0.0)GO TO 52
NUMB = NUMB + 1
IXB(NUMB) = IX
IYB(NUMB) = IY
GO TO 54.
C
52 IX = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
IY = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
NUMB = NUMB + 1
IXB(NUMB) = IX
IYB(NUMB) = IY
GO TO 46.
C
54 WRITE (12) -1,1,IXA(1),IYA(1)
DO 56 J=2,NUMA
56 WRITE (12) 1,1,IXA(J),IYA(J)
IF(NUMB .EQ. 1)GO TO 59
WRITE (12) -1,1,IXB(1),IYB(1)
DO 57 J=2,NUMB
A-147
57 WRITE (12) 1,1,IXB(J),IYB(J)
C
59 CONTINUE
C
ON THE PLOT, CROSS OUT EITHER THE WORD FORECAST OR SOUNDING
C
61 IF(IOPT(1) .NE. 0)GO TO 62
WRITE (12) 1,1,707,604
WRITE (12) 1,1,1174,604
GO TO 64
C
62 WRITE (12) 1,1,1269,604
WRITE (12) 1,1,1760,604
C
PRINT OUT THE PREDICTION TIME ON THE PLOT
C
64 WRITE (12) 1,1,1869,319
WRITE (12,202) 100,0,0,150,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
C
IF THE LAUNCH TIME WAS ENTERED, PRINT IT OUT ON THE PLOT
C
IF(.NOT. LTIME)GO TO 67
WRITE (12) 1,1,1869,112
WRITE (12,208) 100,0,0,150,LTIM,LAUNTD(3),LAUNTD(4),LDAY,
LMON,LYEAR
C
ON THE PLOT, PRINT OUT THE CHARACTERS + AND @ FOR THE LEGEND
C
67 WRITE (12) 1,1,1041,1342
WRITE (12,203) 150,0,0.150,1H+
WRITE (12) 1,1,1041,1104
WRITE (12,203) 150,0,0.150,1H0
C
FOR THE LEGEND ON THE PLOT, PRINT OUT THE CONCENTRATION VALUES
C FOR WHICH CONTOURS WERE DRAWN
C
WRITE (12) 1,1,1066,9587
DO 75 I=1,10
IF(CONC(I) .LT. 0.0)GO TO 77
IF(I .NE. 1)GO TO 72
WRITE (12,204) 125,0,0,150,CONC(I)
GO TO 75
72 WRITE (12,205) 125,0,0,150,CONC(I)
75 CONTINUE
C
WRITE OUT COMMON DISK FILE
C
77 CALL RWDIS(NAME,1)
C
CALL NGRAF TO RETURN SCOPE TO NORMAL MODE OF OPERATION
A-148
CALL MGRAF
RETURN TO THE MAIN PROGRAM REED
STOP
END OF RISOP

SUBROUTINE RYDISCNAHE, JJ)
COMMON ALT(31), AL1, CONMAX, CONCPK, DEGARAD, ADIR, DOSPK, E1, CLDHT,
     IDIR(31), IOPT(3), ITIME, IDAY, MONTH(2), IYEAR, IMON, ISDAY,
     ISMON(2), ISYEAR, IV2, JTOP, JBOT, LAUNTD(10), LTIME, LTIM, LDAY,
     LMON(2), LYEAR, LU, NUM, PI, PI0VR2, PI43, PRESS(31), PTEMP(31),
     SIGCL, RADDEG, RATOMIC, CLRAD, R2, R3, SAVCA(30), SAVCA(30), SIGA,
     SIGX0, SIGX, SPEED(31), SQR2PI, SURED, SIG20, SIGAP, S8, TEMP(31),
     TOPSUR, TWOPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
     YPOS, IFLAG(5), ZB, ZZ, REFLAC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)，(QC2, VPAR(2)，(QC3, VPAR(3)
     (QT1, VPAR(4)，(QT2, VPAR(5)，(QT3, VPAR(6)
     (AA, VPAR(7)，(BB, VPAR(8)，(CC, VPAR(9)
     (HEATN, VPAR(10)，(HEATM, VPAR(11)，(HEATA, VPAR(12)
     (PHCL, VPAR(13)，(PCO, VPAR(14)，(PCO2, VPAR(15)
     (PAL203, VPAR(16)，(PNO, VPAR(17)，(GAMMAX, VPAR(18)
INTEGER ODCB(144), OBUF(669)
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1), ALT(1)
CALL OPEN(ODCB, IERR, NAME, 0)
IF(JJ.EQ.1)CALL WRITE(ODCB, IERR, OBUF, 669)  
IF(JJ.EQ.0)CALL READF(ODCB, IERR, OBUF, 669)
CALL CLOSE(ODCB, IERR)
RETURN
END
SUBROUTINE DFEXP(J, CONC)
*************************************************************************
C THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS
C J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER
C CONC - CONCENTRATION TO BE TESTED
*************************************************************************
COMMON BLOCK

A-149
CALCULATE SIGMA Z

SIGZ = DIST * SIGAP + SIGZ0/1.28
R3 = 2.0 * SIGZ * SIGZ

CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION

TWOI = 2.0
ZT = ALT(J)
TEMP2 = CLDHT - ZZ
TEMP3 = CLDHT - 2.0 * ZB + ZZ
E1 = EXP(- TEMP2 * TEMP2/R3) + EXP(- TEMP3 * TEMP3/R3)

4 TEMP1 = TWOI * (ZT - ZB)
TEXPSM = E1
TEXP = (TEMP1 - TEMP2)**2/R3
IF(TEMP .LE. 120.0)E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 + TEMP2)**2/R3
IF(TEMP .LE. 120.0)E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 - TEMP3)**2/R3
IF(TEMP .LE. 120.0)E1 = E1 + EXP(- TEXP)
TEXP = (TEMP1 + TEMP3)**2/R3
IF(TEMP .LE. 120.0)E1 = E1 + EXP(- TEXP)
IF(E1 .EQ. TEXPSM)GO TO 7
TWOI = TWOI + 2.0
GO TO 4

7 E1 = REFLEC * E1

CALCULATE SIGMA Y

S8 = DIST * SIGAP + SIGX0
R2 = SQRT(S8 * S8 + (0.0040589 * FLOAT(IDIR(J) - IDIR(1)) * DIST)**2)

CALCULATE CLOUD LENGTH

TEMP1 = SPEED(J) - SPEED(1)
AL1 = 0.28 * TEMP1 * DIST/ASPD
IF(TEMP1 .GE. 0.0)GO TO 11
IF(PTEMP(J)-PTEMP(1) .GT. 0.0)AL1 = 0.0
AL1 = ABS(AL1)

CALCULATE SIGMA X

11 SIGX = SQRT((AL1/4.3)**2 + SIGX0 * SIGX0)

IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN TO THE CALLING PROGRAM

IF(CONC .EQ. 1000.0)RETURN

CALCULATE CROSS WIND DISTANCE

Y1 = - 2.0 * R2 * R2 * ALOG(15.7496 * CONC * SIGX * R2 * SIGZ/(SIGHCL * E1))
Y1 = SQRT(AMAX1(Y1,0.0))

RETURN TO THE CALLING PROGRAM

RETURN

END OF DFEXP

END SUBROUTINE ORGIN(IX0,IY0)

*************************************************************************

COMMON BLOCK

COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSP,K,E1,CLDHT,
IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),YEAR,TIM,ISDAY,
ISMON(2),ISYEAR,IY2,JTOP,JBOT,LGMTD(10),LTIME,LMON,LDAY,
LMON(2),LYEAR,LU,NUM,PI,PIVR2,P143,PRESS(31),PTEMP(31),
SIGHCL,RADDEG,RATOMIC,CLDRAD,R2,R3,SAYEA(30),SAVER(30),SIGA,
SICXO, SIGX, SPEED(31), SQRT2PI, SURDEN, SIGZ0, SIGCAP, S8, TEMP(31),
TOPSUR, TWPPI, ASPD, VPAR(18), CRTIME(31), DIST, YES, Y1, NUMRUN,
YPOS, IFLAG(5), ZB, ZZ, REFLEC, IRETRN
LOGICAL LTIME
INTEGER YES
EQUIVALENCE (QC1, VPAR(1)), (QC2, VPAR(2)), (QC3, VPAR(3)),
    (QT1, VPAR(4)), (QT2, VPAR(5)), (QT3, VPAR(6)),
    (AA, VPAR(7)), (BB, VPAR(8)), (CC, VPAR(9)),
    (HEATN, VPAR(10)), (HEATM, VPAR(11)), (HEATA, VPAR(12)),
    (PHCL, VPAR(13)), (PCO, VPAR(14)), (PCO2, VPAR(15)),
    (P1203, VPAR(16)), (PNO, VPAR(17)), (GAMMAX, VPAR(18))
DIMENSION ILINE(32), IDATAF(10), IERS(32), IMAPL(48), NAMEF(3)

C

INPUT FORMAT STATEMENT

100 FORMAT (I2,1XA1)

OUTPUT FORMAT STATEMENT

TYPE AND DIMENSION STATEMENTS

LOGICAL NOT1ST
DIMENSION IX(8), IY(8)

DATA STATEMENTS

DATA IERS/32*2H /
DATA NAMEF/2H?R, 2HIS, 2HOP/
DATA IMAPL/2H40, 2H, 2HEA, 2H M, 2HAP, 2H ,
    1  2H40, 2H, L, 2HAN, 2HD , 2HMA, 2HP ,
    1  2H41, 2H, S, 2HEA, 2H M, 2HAP, 2H ,
    1  2H41, 2H, L, 2HAN, 2HD , 2HMA, 2HP ,
    1  2H17, 2H, S, 2HEA, 2H M, 2HAP, 2H ,
    1  2H17, 2H, L, 2HAN, 2HD , 2HMA, 2HP ,
    1  2H39, 2H, S, 2HEA, 2H M, 2HAP, 2H ,
    1  2H39, 2H, L, 2HAN, 2HD , 2HMA, 2HP /
DATA NOT1ST/.FALSE./, LCHAR/1HL/
DATA IX/5450, 5411, 4830, 4825, 8750, 8730, 4100, 4100/
DATA IY/2630, 8243, 2465, 8050, 2990, 8600, 1700, 7300/

IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
COORDINATES, I, AGAIN

IF(NOT1ST)GO TO 7

THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
AND THE DESIRED MAP, i.e. SEA OR LAND

A-152
EXTERNAL NOTIST
CALL DREAD(NAMEF,7,ILINE)
CALL LERS(YPOS)
CALL CHAR(0,YPOS,0,ILINE,64,0,0)
YPOS = YPOS - 16.
IF(YPOS .LT. 48.) YPOS = 458.
IF(IOPT(3) .EQ. 0) CALL DREAD(NAMEF,8,ILINE)
IF(IOPT(3) .EQ. 2) CALL DREAD(NAMEF,9,ILINE)
IF(IOPT(3) .EQ. 0) CALL DREAD(NAMEF,10,ILINE)
CALL LERS(YPOS)
CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IXC,IYC)
CALL CHAR(0.,YPOS,0,IERES,64,0,0)
CALL CHAR(200.,YPOS+16.,0,IEC,25,0,0)
IF(IXC.LT.6.AND.IOPT(3) .EQ. 1) I=1
IF(IXC.GT.5.AND.IHC.LT.12.AND.IOPT(3) .EQ. 1) I=2
IF(IXC.GT.11.AND.IHC.LT.18.AND.IOPT(3) .EQ. 1) I=3
IF(IXC.GT.17.AND.IOPT(3) .EQ. 1) I=4
IF(IXC.LT.6.AND.IOPT(3) .EQ. 2) I=5
IF(IXC.GE.6.AND.IOPT(3) .EQ. 3) I=6
IF(IXC.LT.6.AND.IOPT(3) .EQ. 0) I=7
IF(IXC.GE.6.AND.IOPT(3) .EQ. 0) I=8
IMP = (I - 1)*6 + 1
CALL CHAR(200.,YPOS+16.,0,IMAPL(IMP),12,0,0)
YPOS = YPOS - 16.
IF(YPOS .LT. 48.) YPOS = 458.

SET THE COORDINATES BASED ON THE INDEX I

7 IX0 = IX(I)
IYO = IY(I)
RETURN TO THE CALLING PROGRAM
RETURN

END OF ORIGIN

END

SUBROUTINE SYMBL(IWIDE,IHI,ISYM)
IX=-IWide/2
IY=-IHi/2
WRITE(12) -1,-1,IX,IY
WRITE(12,100) IWIDE,0,0,IHI,ISYM
100 FORMAT(415,A1,1H_)
IY=-IY
WRITE(12) -1,-1,IX,IY
RETURN
END
SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
DIMENSION NAMEF(3), IDCBF(276), IBUF(40), IILINE(32), IPAR(5)
CALL RMPAR(IPAR)
LU = IPAR(1)
CALL OPEN(IDCBF, IERR, NAMEF, 0)
LOOP = LNUM - 1
DO 10 I=1,LOOP
CALL BLANK(IBUF, 40)
CALL READF(IDCBF, IERR, IBUF)
10 CONTINUE
CALL BLANK(IBUF, 40)
CALL READF(IDCBF, IERR, IBUF)
CALL CODE
READ(IBUF, 100) (ILINE(I), I=1,32)
100 FORMAT(32A2)
CALL CLOSE(IDCBF, IERR)
RETURN
END
SUBROUTINE BLANK(IBUF,II)
DIMENSION IBUF(40)
DATA IBLK/2H. /
DO 10 I=1,II
10 IBUF(I) = IBLK
RETURN
END
SUBROUTINE LERS(YPOS)
DIMENSION IERS(32)
DATA IERS/32*2H /
IF(YPOS.LE.48) YPOS = 4S8.0
CALL CHAR(0., YPOS, 0, IERS, 64, 0, 0)
CALL CHAR(0., YPOS-16., 0, IERS, 64, 0, 0)
RETURN
END
END$
Program MIXH
A-156

```
PROGRAM MIXH
DIMENSION IPAR(5), Z(20), TV(20), IDC(256), IBUF(40), FD(20), P(20)
DIMENSION ITIME(3), DUM(20), ITEST(40), T(20), YP(20)
DIMENSION ITIME(3), IDATE(6)
DATA C1, C2, C3/-0.0005, -0.005, 100./
DATA NAME/2H&M, 2HIX, 2HD1/
DTV(I) = TV(I+1) - TV(I)
DZ(I) = Z(I+1) - Z(I)
GT(I) = DTV(I) / DZ(I)
DS(I) = Z(I+1) - Z(I)
CALL RMPAR(IPAR)
C ** OPEN & MIXD1 DATA FILE
CALL OPEN(IDC, IERR, NAME, 0)
LU = IPAR(1)
C ** INITIALIZE FLAGS TO ZERO
IFL = 0
IFLBS = 0
IFLT = 0
C ** THIS IS TO INPUT THE TIME AND DATE
CALL READ(IDC, IERR, IBUF)
CALL CODE
READ(IBUF, 201) IDATE, ITIME
201 FORMAT(6A2, 2X, 3A2)
DO 444 I = 1, 20
CALL READ(IDC, IERR, IBUF)
CALL CODE
READ(IBUF, *) Z(I), T(I), P(I), FD(I)
C ** CONVERT Z(I) TO METERS
Z(I) = Z(I) * .3048
C ** CONVERT FD(I) TO DECIMAL
FD(I) = FD(I) / 100.
YP(I) = 6.11 * FD(I) * 10. ** (-7.5 * T(I) / (T(I) + 237.3))
TV(I) = (T(I) + 273.16) * (1. + 376932 * YP(I) / P(I)) - 273.16
444 CONTINUE
C ** Z(I) IS ALTITUDE IN METERS
C ** TV(I) IS VIRTUAL TEMPERATURE IN DEG C
C ** P(I) IS PRESSURE IN MILLIBARS
C ** FD(I) IS RELATIVE HUMIDITY
C ** WRITE INPUT VARIABLES
WRITE(6, 6999)
6999 FORMAT(1H1, " ALTIMET " 5X, " TEMPERATURE" 3X, " PRESSURE " 5X
L "RELATIVE HUMIDITY")
WRITE(6, 7000) (Z(I), TV(I), P(I), FD(I), I = 1, 20)
7000 FORMAT(1H1, 4(F10.3, 5X))
C ** SPECIFICATION OF HEIGHT OF GROUND BASED INVERSION
I = 1
IF(GTV(I), LT, C1) GO TO 2
DO 11 I = 2, 19
IF(GTV(I), LT, C1) GO TO 12
```
CONTINUE
WRITE(6,6000)
6000 FORMAT(//,1HO,"INVALID DATA")
GO TO 4
12 IF(DS(I).GE.C3) GO TO 100
GO TO 2
2 WRITE(6,1000)
1000 FORMAT(//,1HO,"NO SURFACE BASED INVERSION")
IFLG1 = 1
GO TO 4
100 WRITE(6,2000) Z(I)
GI = Z(I)
GO TO 4
2000 FORMAT(//.1H,"TOP OF SURFACE BASED INVERSION = ",F7.2)
C ** SPECIFICATION OF THE BASE OF THE FIRST STABLE LAYER
4 DO 10 I = 2,19
IF(GTV(I).GE.C1) GO TO 60
10 CONTINUE
6 WRITE(6,3000)
IFLBS = 1
CALL CLOSE(IDCB,IERR)
GO TO 9000
3000 FORMAT(//.1H,"NO STABLE LAYERS")
200 WRITE(6,4000) Z(I)
BS = Z(I)
GO TO 30
4000 FORMAT(//.1H,"BASE OF FIRST STABLE LAYER = ",F7.2)
60 J = I + 1
DO 61 I = J,19
IF(GTV(I).GE.C1.AND.DS(I).GE.C3) GO TO 200
61 CONTINUE
GO TO 6
C ** SPECIFICATION OF THE TOP OF THE FIRST STABLE LAYER
30 J = I + 1
DO 210 I = J,19
IF(GTV(I).LE.C2) GO TO 300
210 CONTINUE
GO TO 400
300 WRITE(6,5000) Z(I)
TS = Z(I)
5000 FORMAT(//.1H,"TOP OF FIRST STABLE LAYER = ",F7.2)
CALL CLOSE(IDCB,IERR)
GO TO 9000
400 WRITE(6,6000)
6001 FORMAT(//.1H,"TOP OF STABLE LAYER AT ALTITUDE EXCEEDING THE"
11X"MAXIMUM ALTITUDE OF AVAILABLE DATA")
CALL CLOSE(IDCB,IERR)
IFLTS = 1
C *** WRITE OUT DATE-TIME FOR GI,BS,TS
9000 CONTINUE

A-157
WRITE(6,9001)
9001 FORMAT(1H1," DATE AND TIME "6X"GRND INV"6X"BASE LAYER"6X
1"TOP LAYER")
WRITE(6,9002) IDATE,ITIME
9002 FORMAT(1X,6A2,2X,3A2)
IF(IFLGI.EQ.1) WRITE(6,9003)
9003 FORMAT(1H*,22X,"NONE")
IF(IFLGI.EQ.0) WRITE(6,9004) GI
9004 FORMAT(1H*,22X,F5.1)
IF(IFLBS.EQ.1) WRITE(6,9005)
9005 FORMAT(1H*,36X,"NONE")
IF(IFLBS.EQ.0) WRITE(6,9006) BS
9006 FORMAT(1H*,36X,F5.1)
IF(IFLTS.EQ.1) WRITE(6,9007)
9007 FORMAT(1H*,53X,"NONE")
IF(IFLTS.EQ.0) WRITE(6,9008) TS
9008 FORMAT(1H*,53X,F7.2)
END
END$
Program JWSPL
PROGRAM JWSPL
COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAY(8),ISAYF
1,NEWD
DIMENSION IAX1(11),XY1(3),XY2(3),IYTIT(21),ISDATE(8)
DIMENSION WS1(800),WS2(800)
DIMENSION IYLAB(8),IKLAB(12),IXLAB3(2),NITIME(4)
DIMENSION IXLAB2(3),OKPLT(3),IXDATE(9),IXYTIT(21)
DIMENSION ILEG1(7),ILEG2(14),ILEG3(14)
DIMENSION NUMFB(2),IPAR(5),NAME(3),NAME1(3),NAME2(3),NAME3(3)
DATA ILEG1/12,2HNA,2HSA,2H M,2HSF,2HC /
DATA XY1/0.,0.3,0.6,0.
DATA XY2/0.,0.3,0.6/
DATA ILEG2/2G,PHSP,2HAC,2HE,2HIE,2HNC,2HES,12H
DATA ILEG3/26,2HSP,2HAC,2HE,2HIE,2HNC,2HES,12H
DATA IXLAB1/21,2HAC,2HE,2HIE,2HNC,2HES,12H
DATA IXLAB2/2,2H-1/
DATA IXLAB3/1,2H/
DATA IXYTIT/46,2H,2H0,2HIN,2HT,2H02,2HGU,2H J,2HIM,2HSP,
12H,
DATA IXDATE/16/
DATA NAME1/2H J,2HKS,2HC1/
DATA NAME2/2H J,2HKS,2HC2/
DATA NAME3/2H J,2HPT,2HM /
C ** INITIALIZE LU DEVICE
LU = 7
C ** OPEN DATE AND TIME FILE
C CALL OPEN(IDCBE,IERR,NAME,0)
C ** INITIALIZE PLOTTER
CALL PTLU(12)
NEWD = 0
IF1 = 0
INAME = 0
ISAYF = 0
IRDY = 0
CALL CLEER
WRITE(LU,405)
405 FORMAT(/"**HASA/MSFC JIMSPHERE WIND PROFILE PLOTTING"
11X"PROGRAM**")
WRITE(LU,214)
214 FORMAT(/"Jimsphere Wind Profile Data Desired?"5X"WIND SPEED"
11X"WIND DIRECTION")

A-160
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.10) GO TO 215
WRITE(LU,216)
216 FORMAT(/"Use Program JWDPL For Wind Direction Plots")
WRITE(LU,217)
217 FORMAT(/,"*** JWSPL *** TERMINATED ***")
STOP
215 CONTINUE
WRITE(LU,102)
102 FORMAT(/"Jimsphere Wind Profile Data Desired?"5X"CAPE KENNEDY"
110X"POINT MUGU")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.GT.10) GO TO 103
ICK = 1
WRITE(LU,103)
103 FORMAT(/,.5X,"Cape Kennedy Data Desired?"10X"1964-1966"
19X"1967-1970")
104 FORMAT(/,.10X"Point Mugu Data For: 1965-1970")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LE.9) WRITE(LU,105)
IF(IX.LE.9) INAME = '1'
IF(IX.GT.9) WRITE(LU,106)
IF(IX.GT.9) INAME = '2'
IF(INAME.EQ.2) GO TO 172
DO 141 K=1,3
NAME(K) = NAME1(K)
141 CONTINUE
GO TO 173
172 CONTINUE
DO 142 K = 1,3
NAME(K) = NAME2(K)
142 CONTINUE
173 CONTINUE
105 FORMAT(/,.10X"Cape Kennedy Data For: 1964-1966")
106 FORMAT(/,.10X"Cape Kennedy Data For: 1967-1970")
GO TO 72
71 CONTINUE
DO 171 J=1,3
NAME(J) = NAME3(J)
171 CONTINUE
WRITE(LU,104)
72 CONTINUE
WRITE(LU,108)
C108 FORMAT(/,"Jimsphere Wind Speed Data Being Processed")
WRITE(LU, 400)
400 FORMAT(//"TURN ON PLOTTER...POSITION PAPER...TOUCH PANEL WHEN" 
11"READY")
234 CONTINUE
XX1 = 56.
XX1 = 34.
XX1 = 30.
CALL TOUCH(0, 31, 0, 31, IX, IY)
IX = IX/2
IY = IY/2
IF(IX.GT.15) GO TO 234
C THIS IS WHERE THE DISC FILE IS OPENED
CALL OPEN(IDCB, IERR, NAME, 0)
CALL CLEAR
WRITE(LU, 907)
907 FORMAT(//"****PLOTTING HAS BEEN INITIALIZED****")
941 CONTINUE
69 IFLAG = 0
CALL LLEFT
CALL SFAC(15., 10.)
CALL PLOT(1., 1.5, -3)
C ** WRITE NASA LEGEND
CALL PLOT(0., 0., 3)
CALL PLOT(-.5, -.95, 3)
CALL SYMB(-.5, -.95, 1, ILEG1, 0., 1)
CALL PLOT(-.5, -1.1, 3)
CALL SYMB(-.5, -1.1, .08, ILEG2, 0., 1)
CALL PLOT(-.5, -1.25, 3)
CALL SYMB(-.5, -1.25, .08, ILEG3, 0., 1)
C ** THIS PORTION DRAWS Y-AXIS
CALL PLOT(0., 0., 3)
CALL PLOT(0., 0., 2)
CALL PLOT(0., 5., 2)
DO 30 I=1, 10
A = I/2.
CALL PLOT(0., A, 3)
CALL PLOT(0., A, 2)
B = I*2
CALL NUMB(-.3, A, 1, B, 0., -1)
30 CONTINUE
CALL SYMB(-.45, 1.9, 10, IYLAB, 90., 1)
C ** THIS PORTION WRITES HEADERS AND LEGEND
CALL SYMB(3.5, -1.1, 10, IXLAB, 0., 1)
CALL SYMB(5.6, -1.0, 1, IXLAB2, 0., 1)
CALL SYMB(5.8, -1.1, 10, IXLAB3, 0., 1)
IF(INANE.EQ.0) CALL SYMB(2.3, 6.0, 12, IXTIT, 0., 1)
IF(INANE.GT.0) CALL SYMB(2.3, 6.0, 12, IYTIT, 0., 1)
C ** THIS PORTION READS THE FIRST WS1 DATA ARRAY
IF(IF1.EQ.0) CALL RWS2(WS1, IFLAG)
IF(NEWD.EQ.1) GO TO 941
IF1 = 1
IF(ISAVF.NE.1) GO TO 129
DO 128 J=1,8
IDATE(J) = IDSAV(J)
128 CONTINUE
IF(ISAVF.EQ.1) ISAVF = 0
129 CONTINUE
IFLAG = 0
DO 571 KL=1,8
KP = KL + 1
IXDATE(KP) = IDATE(KL)
571 CONTINUE
CALL SYMBC4, + 50,70,12, IXDATE,0..1)
XX = 0.
XY11 = XY1(1)
XY22 = XY2(1)
XYFLG = 1
C ** THIS PORTION DRAWS THE X AXIS **********
95 CALL PLOT(0.,0.,3)
CALL PLOT(X,XY11,-3)
XX=0.
DO 456 I=1,799
IF(WS1(I).GE.100.) GO TO 456
XX = AMAX1(XX,WS1(I))
456 CONTINUE
IX = XX/10 + 2
IF(IX.GT.6) IX = 6
XI = .5 + (IX - 2)* .5
CALL PLOT(XI,0.,2)
DO 35 I=1,IX
A = (I-1)/2.
D = (I-1)*10.
CALL PLOT(A,0.,3)
CALL PLOT(A,.05,2)
B = A -.05
CALL NUMB(B,-.15,.1,D,.0,.1)
35 CONTINUE
B = 0.
JC = 0
CALL PLOT(0.,XY22,-3)
IF(WS1(1).GE.100.) GO TO 642
A=WS1(1)/20.
B= B + .00625
CALL PLOT(A,B,3)
642 CONTINUE
DO 36 I=2,799
B = B + .00625
IF(WS1(I).GE.100.) GO TO 643
A = WS1(I)/20.
JC = JC + 1
36 CONTINUE
IF(JC.EQ.1) CALL PLOT(A,B,3)
CALL PLOT(A,B,2)

CC = B
643 CONTINUE
36 CONTINUE
C = A - .25
D = CC+ .05
HTIME(2) = ITIME(1)
HTIME(3) = ITIME(2)
HTIME(4) = ITIME(3)
CALL SYMB(C,D,.08,HTIME,0.,1)
CALL RWS2(WS2,IFLAG)
IF(NEWD.EQ.1) GO TO 941
IF(IFLAG.EQ.0) GO TO 70
DO 96 I=1,799
WS1(I) = WS2(I)
96 CONTINUE
GO TO 69
70 X = 0.
DO 300 I=1,799
IF(WS1(I).GE.100.0R.WS2(I).GE.100.) DIFF = 0.
IF(WS1(I).GE.100.0R.WS2(I).GE.100.) GO TO 300
DIFF = WS1(I) - WS2(I)
X = AMAX1(X,DIFF)
300 CONTINUE
X = (X/20.)
IF(X.LE..5) X = 0.5
IF(X.GT..5.AND.X.LE.1.) X = 1.0
IF(X.GT.1..5.AND.X.LE.1.5) X = 1.5
IF(X.GT.1.5.AND.X.LE.2.) X = 2.0
IF(X.GT.2..AND.X.LE.2.5) X = 2.5
X = X + 0.5
IF(XYFLG.EQ.1) XY11 = XY1(2)
IF(XYFLG.EQ.1) XY22 = XY2(2)
IF(XYFLG.EQ.2) XY22 = XY2(3)
IF(XYFLG.EQ.2) XY11 = XY1(3)
IF(XYFLG.EQ.3) XY11 = XY1(1)
IF(XYFLG.EQ.3) XY22 = XY2(1)
IF(XYFLG.EQ.3) XYFLG = 0
IF(XYFLG.EQ.3) CALL PLOT(0.,6.,-3)
XYFLG = XYFLG + 1
DO 80 I=1,799
WS1(I) = WS2(I)
80 CONTINUE
GO TO 95
999 CALL URITE
CALL CLOSE(IDC8,IERR)
STOP
END
SUBROUTINE CLEER
INTEGER RSFF
DATA RSFF/017014B/
CALL EXEC(2,1078,RSFF,-2)
RETURN
END
SUBROUTINE RWSS2(WS2,IFLAG)
COMMON IDATE(N),ITIME(N),IDCB(144),IBUF(40),LU,IDSAY(N),ISAYF
1,NEWD
DIMENSION WS2(800)
DATA IBLK/2H
ICF = 0
IF(NEWD.EQ.1) GO TO 942
IK = 1
DO 51 K = 1,100
KK = K*8
READ(8,*)(WS2(IJ),IJ=IK,KK)
IK = IK + 8
51 CONTINUE
CALL READF(IDCBI,IERR,IBUF)
CALL CODE
READ(IBUF,300)(IDATE(NN),NN=1,8),(ITIME(NK),NK=1,3)
300 FORMAT(8A2,3X,3A2)
IF(IDATE(1).EQ.IBLK) GO TO 20
DO 99 J=1,8
IDSAY(J) = IDATE(J)
99 CONTINUE
IF(ICF.EQ.1) GO TO 953
IF(IRDY.EQ.0) GO TO 45
CALL CLEER
WRITE(LU,580)
580 FORMAT(/"DO YOU WISH TO TERMINATE PROGRAM?"/"YES"0X"NO"
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.10) WRITE(LU,349)
IF(IX.LT.10) STOP
349 FORMAT(/"****PROGRAM JIPL HAS BEEN TERMINATED****")
WRITE(LU,101)
101 FORMAT(/"CHANGE PLOT PAPER............TOUCH PANEL TO CONTINUE")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.15) IFLAG = 1
45 CONTINUE
IRDY = 1.
WRITE(LU,100) (IDATE(NK),NK=1,8)
100 FORMAT(/"New Date is: "8A2)
RETURN
WRITE(LU,940)
940 FORMAT(/,"Different Date Desired?"10X"YES"10X"NO")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.GT.6) GO TO 942
WRITE(LU,951)
951 FORMAT(/"ENTER NUMBER OF CURVES SKIPPED? _")
READ(LU,*) ICURS
NFB = ICURS - 1
IFB = NFB
NFB = NFB*100
CALL PTAPE(8,0,NFB)
CALL POINT(IDCB,IERR,IFB,0)
ICF = 1
GO TO 15
942 CONTINUE
943 CONTINUE
NEWD = 0
ISAYF = 1
20 CONTINUE
WRITE(LU,301)(ITIME(NK),NK=1,3)
301 FORMAT(/"Time of Curve is: "3A2,5X"Plot Desired?"10X"YES"10X"NO")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.GT.10) WRITE(LU,223)
223 FORMAT(10X"Curve NOT PLOTTED")
IF(IX.GT.10) GO TO 15
IF(IX.LE.10) WRITE(LU,222)
222 FORMAT(/"Curve Desired...Will It Fit On Paper?"5X"YES"10X"NO")
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.10) GO TO 23
CALL CLEER
WRITE(LU,101)
CALL TOUCH(0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.15) ISAYF = 1
IF(IX.LT.15) IFLAG = 1
23 CONTINUE
WRITE(LU,414)
414 FORMAT(5X,"Curve Being Plotted")
CALL FILTR(WS2)
RETURN
END
SUBROUTINE TOUCH(IXL,IYX,IYL,IYH,IX,IY)
INTEGER ENQ
DIMENSION I(2)
EQUIVALENCE (IA,I(1)),(IB,I(2))
DATA ENQ/002400B/
4 CALL EXEC(2,107B,ENQ,-1)
CALL EXEC(1,107B,I,-4)
IX = IAND(I(1),37B)
IY = IAND(ISHIF(I(2),8),37B)
RETURN
END
SUBROUTINE FILTR(WS1)
DIMENSION WS1(1)
DO 1000 IC1=1,798
IC2 = IC1 + 1
IC3 = IC1 + 2
IF(WS1(IC1) .GE. 100.0)GO TO 1000
DIF1 = WS1(IC1) - WS1(IC2)
DIF2 = WS1(IC1) - WS1(IC3)
DIF3 = WS1(IC2) - WS1(IC3)
IF(ABS(DIF1).GT.1.0 .AND. ABS(DIF3).GT.1.0)WS1(IC2) = WS1(IC1)
IF(((ABS(DIF1).GT.1.0) .AND. (ABS(DIF2).GT.1.0)) .AND.
   ((DIF1.GT.0.0) .AND. (DIF3.LT.0.0)).OR.
   ((DIF1.LT.0.0) .AND. (DIF3.GT.0.0))))WS1(IC2) = WS1(IC1)
1000 CONTINUE
RETURN
END
END$
Program JWDPL
**FTM4.L**

**PROGRAM JUDPL**

COMMON IDATE(8), ITIME(3), IDCBO(144), IBUFFER(40), LU, IDSAV(8), ISAVF

1 NEWD

DIMENSION IAX1(11), XY1(3), XY2(3), IXYTIT(21), ISDAT(8)

DIMENSION WS1(800), WS2(800), DEC1(12), DEC2(12), DEC3(12), DEC4(12)

DIMENSION IYLAB(8), IXLAB(12), IXLAB3(2), NTIME(4)

DIMENSION IXLAB6(3), OKPLT(3), IXDATE(9), IXYTIT(21)

DIMENSION SKP(8)

DIMENSION ILEG1(7), ILEG2(14), ILEG3(14)

DIMENSION NUMFB(2), IPAR(5), NAME(3), NAME1(3), NAME2(3), NAME3(3)


DATA IXY1/0., -3., -3., 0., 3., 3., 6.

DATA IXY2/0., -3., -3., 0., 3., 3., 6.

DATA ILEG2/26, 2HSP, 2HAC, 2HE, 2HSC, 2HIE, 2HNC, 2HES,

12H L, 2HAB, 2HOR, 2HAT, 2HOR, 2HY /

DATA IAX1/2, 2H, 2H 4, 2H 6, 2H 8, 2H10, 2H12, 2H14, 2H16, 2H18, 2H20 /

DATA ILEG3/26, 2HAE, 2HRO, 2HSP, 2HAC, 2HE, 2HEN, 2HVI, 2HRO,

12HNM, 2HEN, 2HT, 2HDI, 2HY /

DATA IXLAB/14, 2HAL, 2HTI, 2HTU, 2HDE, 2H, (2HKM, 2H) /

DATA NTIME/5 /

DATA IXLAB/21, 2HWI, 2HND, 2H D, 2HIR, 2HEC, 2HTI, 2HON, 2H (2HDE,

1 2HGS, 2H) /

DATA IXLAB2/2, 2H-1 /

DATA IXLAB3/1, 2HT) /

DATA IXYTIT/40, 2H , 2HPO, 2HIN, 2HT, 2HBU, 2HGU, 2H J, 2HIM, 2HSP,

1 2HHE, 2HRE, 2H W, 2HIN, 2HD, 2HPR, 2HOF, 2HIL, 2HE, 2HDA, 2HTA/

DATA IXDATE/16 /

DATA NAME1/2H&J, 2HKS, 2HC1 /

DATA NAME2/2H&J, 2HKS, 2HC2 /

DATA NAME3/2H&J, 2HPT, 2HM /

C ** INITIALIZE LU DEVICES**

CALL RMPAR(IPAR)

LU = IPAR(1)

C CALL EXEC(22, 1)

LU = 7

C ** OPEN DATE AND TIME FILE**

C CALL OPEN(IDCBO, IERR, NAME, 0)

C ** INITIALIZE PLOTTING**

CALL PLTLU(12)
NEWD = 0
IF1 = 0
INAME = 0
ISAVF = 0
IRDY = 0
CALL CLEAR
C405 FORMAT(///"****HASA/MSFC JIMSPHERE WIND PROFILE PLOTTING"
C 11X"PROGRAM****")
C WRITE(LU,214)
C214 FORMAT(///"Jimsphere Wind Profile Data Desired?"5X"WIND SPEED"
C 10X"WIND DIRECTION")
CALL TOUCH(0.15,0.15,IX,IY)
IF(IX.GT.9) GO TO 215
WRITE(LU,216)
C16 FORMAT(///"Use Program JWDPL For Wind Direction Plots")
WRITE(LU,217)
C17 FORMAT(///"**** JWSPL *** TERMINATED ****")
STOP
215 CONTINUE
WRITE(LU,102)
102 FORMAT(///"Jimsphere Wind Profile Data Desired?"5X"CAPE KENNEDY"
10X"POINT MUGU")
CALL TOUCH(0.15,0.15,IX,IY)
IF(IX.GT.10) GO TO 71
ICK = 1
WRITE(LU,103)
103 FORMAT(///"Cape Kennedy Data Desired?"10X"1964-1966"
19X"1967-1970")
104 FORMAT(///"Point Mugu Data For: 1965-1970")
CALL TOUCH(0.15,0.15,IX,IY)
IF(IX.LE.9) WRITE(LU,105)
IF(IX.LE.9) INAME = 1
IF(IX.GT.9) WRITE(LU,106)
IF(IX.GT.9) INAME = 2
IF(INAME.EQ.2) GO TO 172
DO 141 K=1,3
NAMECK = NAME1(K)
141 CONTINUE
GO TO 173
172 CONTINUE
DO 142 K = 1,3
NAMECK = NAME2(K)
142 CONTINUE
173 CONTINUE
105 FORMAT(///"Cape Kennedy Data For: 1964-1966")
106 FORMAT(///"Cape Kennedy Data For: 1967-1970")
GO TO 72
71 CONTINUE
DO 171 J=1,3
NAME(J) = NAME3(J)
171 CONTINUE
   WRITE(LU,104)
72 CONTINUE
C WRITE(LU,100)
C100 FORMAT(//,"Jimosphere Wind Speed Data Being Processed")
   WRITE(LU,400)
400 FORMAT(//,"TURN ON PLOTTER...POSITION PAPER...TOUCH PANEL WHEN""
   11X*READY")
234 CALL TOUCH(0,15,0,15,IX,IY)
   IF(IX.GT.15) GO TO 234
C THIS IS WHERE THE DISC FILE IS OPENED
   CALL OPEN(IDCB,IERR,NAME,0)
   CALL CLEAR
C WRITE(LU,907)
C07 FORMAT(/****PLOTTING HAS BEEN INITIALIZED****")
941 CONTINUE
69 IFLAG = 0
   CALL LLEFT
   CALL SFACT(15.,10.)
   CALL PLOT(1,1.5,-3)
C ** WRITE HASA LEGEND
   CALL PLOT(0,0,3)
   CALL PLOT(-.5,-.95,3)
   CALL SYMB(-.5,-.95,1,ILEG1,0,1)
   CALL PLOT(-.5,-1.1,3)
   CALL SYMB(-.5,-1.1,08,ILEG2,0,1)
   CALL PLOT(-.5,-1.25,3)
   CALL SYMB(-.5,-1.25,08,ILEG3,0,1)
C ** THIS PORTION DRAWS Y-AXIS
   CALL PLOT(0,0,3)
   CALL PLOT(0,0,2)
   CALL PLOT(0,.5,2)
   DO 30 I=1,10
      A = I/2.
   CALL PLOT(0,A,3)
   CALL PLOT(.05,A,2)
50 B = I*2
   CALL HUMB(-.3,A,.1,.8,.0,-1)
30 CONTINUE
   CALL SYMB(-.45,1.9,.10,IYLAB,.90,1)
C ** THIS PORTION WRITES HEADERS AND LEGEND
   CALL SYMB(3.5,-1.1,.10,IYLAB,0,1)
   IF(INAME.EQ.0) CALL SYMB(2.3,.6,.0,.12,IXTIT,0,1)
   IF(INAME.GT.0) CALL SYMB(2.3,.6,.0,.12,IXTIT,0,1)
C ** THIS PORTION READS THE FIRST WS1 DATA ARRAY
   IF(IF1.EQ.0) CALL RWS2(WS1,IFLAG,IQDS)
   IF(NEWD.EQ.1) GO TO 941
   IF1 = 1
   IF(ISAVF.NE.1) GO TO 129
DO 128 J=1,8
   IDATE(J) = IDSAY(J)
128 CONTINUE
   IF(ISAYF.EQ.1) ISAYF = 0
129 CONTINUE
   IFLAG = 0
   DO 571 KL=1,8
      KP = KL + 1
      IDATE(KP) = IDATE(KL)
571 CONTINUE
   CALL SYMB(4.,5.70,12,IXDATE,0.,1)
   XX = 0.
   XY11 = XY1(I)
   XY22 = XY2(I)
   XYFLG = 1
   C ** THIS PORTION DRAWS THE X AXIS **********
95 CALL PLOT(0.,0.,3)
   CALL PLOT(X,XY11,-3)
   XX = 0.
   DO 456 I=1,790
      IF(WS1(I).GE.1000.) GO TO 456.
      XX = AMAX1(XX,WS1(I))
456 CONTINUE
   XX=(XX+SKP(IQDS))/180.0
   XX=AINT(2.0*(XX+0.4999999))/2.0
   CALL PLOT(XX,0.,2)
   NXX = INT(2.0 * XX) + 1
   DO 35 I=1,NXX
      A = (I-1)/2.
      D=DEG1(I)
      IF(IQDS.EQ.1) D = DEG2(I)
      IF(IQDS.EQ.2) D = DEG3(I)
      IF(IQDS.EQ.3) D = DEG4(I)
      IF(IQDS.EQ.4) D = DEG5(I)
      IF(IQDS.EQ.5) D = DEG6(I)
      IF(IQDS.EQ.6) D = DEG7(I)
      IF(IQDS.EQ.7) D = DEG8(I)
      IF(IQDS.EQ.8) D = DEG9(I)
      CALL PLOT(A,O,3)
      CALL PLOT(A,.05,2)
      B = A -.05
      CALL NUMB(B,-.15,.1,D,0.,-1)
35 CONTINUE
   B=0.
   CALL PLOT(0.,XY22,-3)
   IF(WS1(I).GE.1000.) GO TO 642
   A=(WS1(I)+SKP(IQDS))/180.
   B = B + .00625
   CALL PLOT(A,B,3)
642 CONTINUE
N = 1
JFLG = -1
DO 36 I=2,790
B = B + .00625
IF(WS1(I).GE.1000.) GO TO 647
A = (WS1(I)+SKP(IQDS))/100
XC = ABS(WS1(I)-WS1(N))
IF(JFLG.NE.0) CALL PLOT(A,B,3)
JFLG=0
CALL PLOT(A,B,2)
CC= B
N = I
643 CONTINUE
36 CONTINUE
C = A -.25
D = CC+ .05
NTIME(2) = ITIME(1)
NTIME(3) = ITIME(2)
NTIME(4) = ITIME(3)
CALL SYMB(C,D,.08,NTIME,0,1)
CALL RWS2(WS2,IFLAG,IQDS)
IF(NEWD.EQ.1) GO TO 941
IF(IFLAG.EQ.0) GO TO 70
DO 96 I=1,790
WS1(I) = WS2(I)
96 CONTINUE
GO TO 69
70 X = XX
IF(XYFLG.EQ.1) XY11 = XY1(2)
IF(XYFLG.EQ.1) XY22 = XY2(2)
IF(XYFLG.EQ.2) XY22 = XY2(3)
IF(XYFLG.EQ.2) XY11 = XY1(3)
IF(XYFLG.EQ.3) XY11 = XY1(1)
IF(XYFLG.EQ.3) XY22 = XY2(1)
IF(XYFLG.EQ.3) XYFLG = 0
IF(XYFLG.EQ.3) CALL PLOT(0,6,-3)
XYFLG = XYFLG + 1
DO 80 I=1,790
WS1(I) = WS2(I)
80 CONTINUE
GO TO 95
999 CALL URITE
C CALL EXEC(22,0)
CALL CLOSE(IDCBI,IERR)
END
SUBROUTINE RWS2(WS2,IFLAG,IQDS,DSHF)
COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAY(8),ISAYF
1,NEWD
DIMENSION WS2(800)
DATA IBLK/2H /
ICF = 0
IF(NEWD.EQ.1) GO TO 942
15 IK = 1
DO 51 K = 1,100
KK = K*8
READ(B,*)(WS2(IJ),IJ=IK,KK)
IK = IK + 8
51 CONTINUE
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,300)(IDATE(NH),NH=1,8),(ITIME(NK),NK=1,3)
300 FORMAT(8A2,3X,3A2)
IF(IDATE(1).EQ.IBLK) GO TO 20
DO 89 J=1,8
IDSAV(J) = IDATE(J)
89 CONTINUE
IF(ICF.EQ.1) GO TO 953
IF(IRDY.EQ.0) GO TO 45
CALL CLEAR
WRITE(LU,580)
580 FORMAT(/,"DO YOU WISH TO TERMINATE PROGRAM?"10X"YES"10X"NO")
CALL TOUCH(0,15,0,15,IX,IY)
C IF(IX.LT.10) WRITE(LU,349)
IF(IX.LT.10) STOP
C49 FORMAT(/"****PROGRAM JIMPL HAS BEEN TERMINATED****")
WRITE(LU,101)
101 FORMAT(/"CHANGE PLOT PAPER.................TOUCH PANEL TO CONTINUE")
CALL TOUCH(0,15,0,15,IX,IY)
IF(IX.LT.15) IFLAG = 1
45 CONTINUE
IRDY = 1
WRITE(LU,100)(IDATE(NK),NK=1,8)
100 FORMAT(/"New Date is: "8A2)
C RETURN
WRITE(LU,940)
940 FORMAT(/,"Different Date Desired?"10X"YES"10X"NO")
CALL TOUCH(0,15,0,15,IX,IY)
IF(IX.GT.6) GO TO 942
WRITE(LU,951)
951 FORMAT(/"ENTER NUMBER OF CURVES SKIPPED? _")
READ(LU,*) ICURS
NFB = ICURS - 1
IFB = NFB
NFB = NFB*100
CALL PTAPE(0,0,NFB)
CALL POSNT(IDCB,IERR,IFB,0)
ICF = 1
GO TO 15
953 CONTINUE
CONTINUE
NEWD = 0
ISAYF = 1
CONTINUE
WRITE(LU,301)(ITIME(NK),NK=1,3)
301 FORMAT(2A2,A1,5X"Plot Desired?"10X"YES" 10X"NO")
CALL TOUCH(0,15,0,15,IX,IY)
CONTINUE
IF(IX.GT.10) WRITE(LU,223)
C23 FORMAT(10X"CURVE NOT PLOTTED")
IF(IX.GT.10) GO TO 15
IF(IX.LE.10) WRITE(LU,222)
222 FORMAT(5X,"Curve Desired... Will It Fit On Paper?"5X"YES"10X"NO")
CALL TOUCH(0,15,0,15,IX,IY)
IF(IX.LT.10) GO TO 23
CALL CLEAR
WRITE(LU,101)
CALL TOUCH(0,15,0,15,IX,IY)
IF(IX.LT.10) ISAYF = 1
IF(IX.LT.10) IFLAG = 1
CONTINUE
WRITE(LU,414)
C14 FORMAT(5X,"Curve Being Plotted")
CALL FILTR(WS2,IQDS,DSHF)
RETURN
END
SUBROUTINE TOUCH(IXL,IXH,IYL,IYH,IX,IY)
INTEGER ENQ
DIMENSION I(2)
EQUIV(I,A(I(1)),I(B(I(2))))
CALL EXEC(2,107B,ENQ,-1)
CALL EXEC(1,107B,I,-4)
IX = IAND(ISHIFT(1B,-9),1B)
IX = IOR(IAND(ISHIFT(1A,-3),2B),IX)
IX = IOR(IAND(ISHIFT(1A,-1),4B),IX)
IX = IOR(IAND(ISHIFT(1A,1),10B),IX)
IF(IX.LT.IXL .OR. IX.GT.IXH)GO TO 4
IY = IAND(ISHIFT(1B,-12),1B)
IY = IOR(IAND(ISHIFT(1B,-10),2B),IY)
IY = IOR(IAND(ISHIFT(1B,-8),4B),IY)
IY = IOR(IAND(ISHIFT(1B,-6),10B),IY)
IF(IY.LT.IYL .OR. IY.GT.IYH)GO TO 4
RETURN
END
SUBROUTINE CLEAR
INTEGER RSFF
DATA RSFF/017014B/
CALL EXEC(2,107B,RSFF,-2)
RETURN
SUBROUTINE FILTR(WS1, IQDS, DSHF)
DIMENSION WS1(I)
N = 1
DO 100 I=2,790
  IF(WS1(N).GT.100.) GO TO 50
  IF(WS1(I).GT.100.) GO TO 100
  IF(ABS(WS1(N)-WS1(I)).LT.180.) GO TO 50
  WS1(I) = WS1(I) + 360.0
  GO TO 50
40  WS1(I) = WS1(I) - 360.0
N = I
100 CONTINUE
RHIN = 10000.
DO 101 I=2,790
  RHIN = AHIH1(RHIN,WS1(I))
101 CONTINUE
IF(RHIN.GT.-270. .AND. RHIN.LE.-180.) IQDS = 1
IF(RHIN.GT.-180. .AND. RHIN.LE.-90.) IQDS = 2
IF(RHIN.GT.-90. .AND. RHIN.LE.0.) IQDS = 3
IF(RHIN.GT.0. .AND. RHIN.LE.90.) IQDS = 4
IF(RHIN.GT.90. .AND. RHIN.LE.180.) IQDS = 5
IF(RHIN.GT.180. .AND. RHIN.LE.270.) IQDS = 6
IF(RHIN.GT.270. .AND. RHIN.LE.1000.) IQDS = 7
IF(RHIN.GT.-360. .AND. RHIN.LE.-270.) IQDS = 8
IF(IQDS.EQ.1) DSHF = 270.
IF(IQDS.EQ.2) DSHF = 180.
IF(IQDS.EQ.3) DSHF = 90.
IF(IQDS.EQ.4) DSHF = 0.
IF(IQDS.EQ.5) DSHF = -90.
IF(IQDS.EQ.6) DSHF = -180.
IF(IQDS.EQ.7) DSHF = -270.
IF(IQDS.EQ.8) DSHF = 360.
RETURN
END
SUBROUTINE IQDCP(IQD, A)
IF(IQD.EQ.2.AND.A.GT.5) A = A - .5
IF(IQD.EQ.2.AND.A.LE.5) A = A + 1.5
IF(IQD .EQ.3.AND.A.GT.1.) A = A - 1.
IF(IQD .EQ.4.AND.A.LE.1.5) A = A + .5
IF(IQD .EQ.4.AND.A.GT.1.5) A = A - 1.5
RETURN
END
END
Program JIMPS
PROGRAM JIMPS
DIMENSION X(4),Y(4),XX(10),YY(800),IHEAD(20),WS1(800)
DIMENSION X Tic(12),X Tic1(2), Y Tic(42),Y Tic1(2),IN X NUM(6)
DIMENSION IP AR(5)
DIMENSION ITIME(18),IERS(32),JWSPL(3),I LINE(32)
DIMENSION IXLEG(9),JIMFI(3)
DIMENSION IDATE(8),YL AB(10),Y ALT(8),X NUM(12)
INTEGER YLAB,Y ALT
REAL LL
DATA IERS/32*200408/
DATA JWSPL/2HJW,2HSP,2HL/
DATA JIMFI/2HJi,2HM F,2H1/
DATA ITIME/2H13,2H06,2Hz ,2H14,2H31,2Hz ,2H16,2H00,2Hz ,
12H17,2H31,2Hz ,2H19,2H00,2Hz ,2H22,2H00,2Hz /
DATA X/64.,164.,64.,64./
DATA Y Tic1/62.,166./
1272.,272.,296.,296.,320.,320.,344.,344.,368.,368./
DATA IX NUM/2H 0.2H10,2H20,2H30,2H40,2H50/
DATA IXLEG/2HSC,2HAl,2Har,2H W,2Him,2Hd ,2HSp,2Hee,2Hd /
1156.,156./
DATA X Tic/64.,64.,84.,84.,104.,104.,124.,124.,144.,144.,
1164.,164./
DATA X Tic1/126.,130./
DATA IHEAD/2HCa,2Hpe,2H K,2Hen,2Hne,2Hdy,2H J,2Him,2Hsp,2Hhe,
12Hre,2H W,2Him,2Hd ,2HPr,2Hof,2Hil,2He ,2HDa,2Hta/
DATA Y ALT/2HA,2HI ,2Ht ,2Hi ,2Ht ,2Hw ,2Hd ,2He /
DATA Y/128.,128.,128.,368./
DATA XX/50.,75.,150.,175.,200.,250.,275.,300.,350.,360./
DATA YY/50.,100.,150.,175.,200.,300.,325.,350.,360.,368./
DATA IDATE/2H ,2H D,2Hec,2H 2,2H99 ,2H 1,2H96,2H4 /
DATA YL AB/2H 2,2H 4,2H 6,2H 8,2H10,2H12,2H14,2H16,2H18,2H30/
C ** INITIALIZE LU DEVICE
CALL RMPAR(IP AR)
LU = IP AR(5)
C ** THIS PROGRAM IS TO TEST %PLIB LIBRARY OF SUBROUTINES
C ** WRITTEN FOR THE PLASMA SCOPE
C ** CALL GRAF TO INITIATE PLASMA SCOPE FOR GRAPHING
C ** CALL GRAF(0)
C ** CALL CLEAR TO CLEAR PLASMA SCOPE
1 CALL CLEAR
C ** CALL SETOR(XORG,Y ORG) TO INITIALIZE X,Y ORIGIN
IF G = 0
CALL SETOR(0.,0.)
C ** CALL SETSC(XSCAL,YSCAL) TO SET SCALE FACTORS
CALL SETSC(1.,1.)
C ** CALL LINE(X,Y,NXY,MODE) TO PLOT POINTS
C ** X AND Y = THE X,Y CO-ORDINATES
** NXY = THE NUMBER OF POINTS TO BE PLOTTED
** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
** CALL POINT(X,Y,NXY,MODE) SAME AS ABOVE EXCEPT PLOTS A LINE

XSHF = -64.
71 CALL LINE(X,Y,2,0)
    DO 70 J=1,6
        I = (J-1)*2 + 1
        CALL LINE(XTIC(I),XTIC1,2,0)
        CALL CHAR(XNUM(I),110.,0,IXNUM(J),2,0,0)
    70 CONTINUE
    IF(IFG.GT.0) GO TO 52
    CALL LINE(X(3),Y(3),2,0)
    YLB = Y(1) - 8.
    YNM = Y(1)
    XLB = X(1) - 24.
    DO 10 I=1,10
        YLB = YLB + 24.
        YNM = YNM + 24.
        J = (I-1)*2 + 1
        CALL CHAR(XLB,YLB,J,YLB,J,2,0)
        CALL LINE(YTIC(J),YTIC(J),2,0)
    10 CONTINUE

** PRINT SCALAR WIND LEGEND
CALL CHAR(175.,40.,0,IXLEG,18,0,0)

** PRINT ALTITUDE LEGEND
YAL = 360.
    DO 20 I =1,8
        YAL = YAL - 24.
        CALL CHAR(YA,YAL,0,YALT(I),2,0,0)
    20 CONTINUE

** PRINT HEADER
CALL CHAR(100.,470.,0,IHEAD,40,0,0)
CALL CHAR(200.,445.,0,IDATE,16,0,0)

** RESET ORIGIN TO PLOT LINE
52 CONTINUE
    CALL SETOR(XSHF,-128.)
    CALL SETSC(3.2,1.)
    CALL SETSC(1.,1.)
    CALL LINE(XX,YY,10,0)
    LL = 0.
    DO 40 L=1,1000
        LL = LL + .3
        YY(L) = LL
    40 CONTINUE
    IK = 1
    DO 30 K=1,100
        KK = K*8
        READ(8,*) (WS1(IJ),IJ=IK,KK)
        IK = IK + 8
    30 CONTINUE
CALL FILTR(US1)
N = 0
DO 50 L=1,797,4
IF(US1(L).GT.100.) GO TO 50
N = N + 1
US1(N) = US1(L)*2.
YY(N) = YY(L)
C CALL POINT(US1(L),YY(L),1,0)
50 CONTINUE
CALL LINE(US1,YY,N,0)
NJ = (IFG*3) + 1
R = YY(N) + 4.
CALL CHAR(US1(N),R,0,ITIME(NJ),6,0,0)
C IF(IFG.EQ.0) CALL SETOR(-124.,-110.)
C IF(IFG.EQ.1) CALL SETOR(-184.,-92.)
C X(1) = X(1) + 60.
C X(2) = X(2) + 60.
C Y(1) = Y(1) - 22.
C Y(2) = Y(2) - 22.
C IF(IFG.EQ.0) CALL SETOR(-60.,22.)
C IF(IFG.EQ.1) CALL SETOR(-120.,44.)
C IF(IFG.EQ.2) CALL SETOR(-180.,0.)
C IF(IFG.EQ.3) CALL SETOR(-240.,22.)
C IF(IFG.EQ.4) CALL SETOR(-300.,44.)
C IF(IFG.GE.5) GO TO 51
IFG = IFG + 1.
XSHF = XSHF - 60.
GO TO 71
51 CONTINUE
C CALL LINE(US1,YY,800,0)
C ** CALL TOUCH TO SEE IF USER DESIRES TO CONTINUE OR TERMINATE
C CALL NGRAF
C CALL GRAF(0)
345 CALL DREAD(JIMF1,2,ILINE)
CALL CHAR(8.0,16.0,ILINE,64,0,0)
CALL IN(1,JTYPE,0.,0.,0,9,0,0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.14) GO TO 1
CALL CHAR(8.,16.0,IER5,64,0,0)
CALL DREAD(JIMF1,3,ILINE)
CALL CHAR(8.,16.0,ILINE,64,0,0)
CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
IX = IX/2
IY = IY/2
IF(IX.LT.8) GO TO 344
REWIND 8
CALL NGRAF
CALL EXEC(9,USPL)
CALL GRAF(0)

A-180
GO TO 345
C ** CALL MGRAF TO RE-ESTABLISH PLASMA SCOPE FOR TOUCH MODE
344 CALL CLEAR
    CALL MGRAF
C ** REWIND TAPE
    REWIND 8
STOP
END
SUBROUTINE FILTR(WS1)
DIMENSION WS1(1)
DO 1000 IC1=1,798
    IC2 = IC1 + 1
    IC3 = IC1 + 2
    IF(WS1(IC1) .GE. 100.0)GO TO 1000
    DIF1 = WS1(IC1) - WS1(IC2)
    DIF2 = WS1(IC1) - WS1(IC3)
    DIF3 = WS1(IC2) - WS1(IC3)
    IF(ABS(DIF1).GT.1.0 .AND. ABS(DIF3).GT.1.0)WS1(IC2) = WS1(IC1)
    IF((ABS(DIF1).GT.1.0) .AND. (ABS(DIF2).GT.1.0) .AND.
        (((DIF1.GT.0.0) .AND. (DIF3.LT.0.0)) .OR.
        ((DIF1.LT.0.0) .AND. (DIF3.GT.0.0))))WS1(IC2) = WS1(IC1)
1000 CONTINUE
RETURN
C END
C SUBROUTINE CLEAR
C INTEGER RSFF
C DATA RSFF/017014B/
C CALLEXEC(2,107B,RSFF,-2)
C RETURN
END
SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32)
CALL OPEN(IDCBI,IER,NAMEF,0)
LOOP = LNUM - 1
DO 10 I=1,LOOP
    CALL BLANK(IBUF,40)
    CALL READF(IDCBI,IER,IBUF)
10 CONTINUE
CALL BLANK(IBUF,40)
CALL READF(IDCBI,IER,IBUF)
CALL CODE
READ(IBUF,100) (ILINE(I),I=1,32)
100 FORMAT(32A2)
CALL CLOSE(IDCBI,IER)
RETURN
END
SUBROUTINE BLANK(IBUF,II)
DIMENSION IBUF(40)
DATA IBLK/2H /
DO 10 I=1,II
10 CONTINUE
10  IBUF(I) = IBLK
RETURN
END
END$
Program SKEW T (Version I)
PROGRAM SKEWT
DIMENSION IALT(31,3), IDIR(31,3), SPEED(31,3), TEMP(31,3), PRESS(31,3)
DIMENSION SURDN(6), V(31), ISWR(20), DPTEMP(31,3), PTEMP(31,3)
DIMENSION Q(30), A(23), C(23), B(23), E(15), F(4,4), G(4,4), K(23)
DIMENSION W(31,5), X(40,5), LHEAD(40,80), ALT(31,3)

CALL DATE
P9=3.14159
M=0
ISWR(1)=0
ISWR(2)=0
ISWR(3)=0
ISWR(4)=0
ISWR(5)=0
ISWR(6)=0
ISWR(7)=0
ISWR(8)=0
ISWR(9)=0
ISWR(10)=0
ISWR(11)=0
ISWR(12)=0
ISWR(15)=1

IUNIT=5
IF(ISWR(1) .EQ. 0) IUNIT=1

N=1
IF(ISWR(8) .EQ. 1) GO TO 140
WRITE(6,9010)

C : DEFINITION OF TERMS:
C TEMP(IALT,N)--TEMPERATURE; PRESS(IALT,N)--PRESSURE; DPTEMP(IALT,N)--HUMIDITY

C **LOAD DATA**
ITIMES=5
CALL IOHED(LHEAD, ITIMES)

READ(8,*) N
READ(8,9860) LTIM, LDAY, LMON, LYEAR
ITIMES=19
CALL IOHED(LHEAD, ITIMES)

WRITE(6,9015)
READ(1,9850) IFNO
CALL PTAPE(8, IFNO, 0)

C
T2=9999
ITIMES=3
CALL IOHED(LHEAD, ITIMES)

8000 READ(8,9865) ISTIM, ISDAY, ISMON, ISM, ISYEAR
IF(ISDAY.EQ.0) GO TO 8000
WRITE(12) -1,1,1000,9000
WRITE(12,8040)175,0,0,200, ISTIM, ISDAY, ISMON, ISM, ISYEAR

A-184
8040  FORMAT(415, 20HRAWINSONDE SOUNDING: , I5, 1HZ, 2X, I2, 1X, 2A2, I4)  
     ITIMES=2  
     CALL IOHED(ICHEN, ITIMES)  
     READ (8, 9870) IALT(I, N), IDIR(I, N), SPEED(I, N), TEMP(I, N),  
     1DPTEMP(I, N), PRESS(I, N), SURDEN(N)  
     DO 120 I = 2, 30  
115  READ (8, 9875) IALT(I, N), IDIR(I, N), SPEED(I, N), TEMP(I, N),  
     1DPTEMP(I, N), PRESS(I, N)  
     CALL EXEC(I3, 8, IEQT)  
     IEQT = IAND(IEQT, 2008)  
     IF (IEQT .GT. 0) GO TO 140  
     IF (IALT(I, N) .LT. 10) GO TO 115  
     IF (IALT(I, N) .GT. 10000) GO TO 115  
     JARAY = I  
120  CONTINUE  
     C  
     C  
140  IF (IUNIT .EQ. 2 OR. IUNIT .EQ. 3) WRITE(6, 4020)  
4020  FORMAT ("NEED JUMP")  
     C : N---DATA SET NUMBER; LMON, LDAY, LYEAR, LTIM---LAUNCH DATE/TIME  
     C :  
     C : ISTIM--SOUNDING TIME; T2--PREDICTION TIME  
     C : CONVERTING SOUNDING TIME FROM ZULU TO EDT - AM, PM  
     ISTIM = ISTIM - 400  
     IF (ISTIM .GT. 0) GO TO 250  
     ISTIM = 2400 - ABS(ISTIM)  
     ISDAY = ISDAY - 1  
250  IF (ISTIM .GE. 1300) GO TO 260  
     IF (ISTIM .GE. 1200 AND. ISTIM .LT. 1300) GO TO 270  
     GO TO 280  
260  ISTIM = ISTIM - 1200  
270  CONTINUE  
280  CONTINUE  
     C : SURDEN(N) = SURFACE DENSITY  
     C  
     C : CONVERT DATA TO METRIC, SORT DATA BY IALT, CAL POT TEMP=PTEMP(IALT, N)  
     DO 590 I = 1, JARAY  
     C :  
     C : ............ENGLISH TO METRIC.............  
     ALT(I, N) = IALT(I, N)  
     ALT(I, N) = .3048 * ALT(I, N)  
     SPEED(I, N) = .515 * SPEED(I, N)  
     C :  
     C : ............SORT............................  
809  L = I  
509  IF (L .EQ. 1) GO TO 590  
     IF (ABS(ALT(L, N)) .GT. ALT(L-1, N)) GO TO 590  
     ALT(31, N) = ALT(L-1, N)  
     IDIR(31, N) = IDIR(L-1, N)  
     SPEED(31, N) = SPEED(L-1, N)  
     TEMP(31, N) = TEMP(L-1, N)  
     PRESS(31, N) = PRESS(L-1, N)
DPTEMP(31,N) = DPTEMP(L-1,N)
ALT(L-1,N) = ALT(L,N)
IDIR(L-1,N) = IDIR(L,N)
SPEED(L-1,N) = SPEED(L,N)
TEMP(L-1,N) = TEMP(L,N)
PRESS(L-1,N) = PRESS(L,N)
DPTEMP(L-1,N) = DPTEMP(L,N)
ALT(L,N) = ALT(31,N)
IDIR(L,N) = IDIR(31,N)
SPEED(L,N) = SPEED(31,N)
TEMP(L,N) = TEMP(31,N)
PRESS(L,N) = PRESS(31,N)
DPTEMP(L,N) = DPTEMP(31,N)
L=L-1

570 GOTO 510
590 CONTINUE
C CALCULATE POTENTIAL TEMPERATURE (DEG K) PTEMP(ALT,N)
DO 690 I=1,JARAY
C ALT(I,N) = ABS(ALT(I,N))
PTEMP(I,N) = (TEMP(I,N) + 273.15) * ((1000/PRESS(I,N)) ** .288)
690 CONTINUE
C PRINT METEOROLOGICAL DATA

725 J=J9
C IF (ISWR(12).EQ.0) WRITE (6,*) "CTR PRINT"
C IF (ISWR(12).EQ.0) WAIT (15000)
WRITE (6,9220)
WRITE (6,9140)
WRITE (6,9140)
WRITE (6,9140)
IF (ISWR(15).EQ.0) WRITE (6,9230)
IF (ISWR(15).EQ.1) WRITE (6,9240)
WRITE (6,9250) LTIM, LDAY, LMON, LM, LYEAR
WRITE (6,9140)
WRITE (6,9260) ISTIM, ISDAY, ISMON, ISM, ISYEAR
WRITE (6,9270) T2
E(6) = .66355
WRITE (6,9280) N, SRUDE(H)
WRITE (6,9140)
WRITE (6,9290)
WRITE (6,9300)
DO 850 I=1,JARAY
SPEED(I,N) = INT(SPEED(I,N) * 10) / 10
IALTF = ALT(I,N) / .3048 + .5
IALT(I,N) = ALT(I,N) + .5
APTEMP = PTEMP(I,N) - 273.15
WRITE (6,9310) I, IALTF, IALT(I,N), IDIR(I,N), SPEED(I,N), TEMP(I, N), APTEMP, DPTEMP(I,N), PRESS(I,N)
850 CONTINUE
C
C PLOT SKEW T / LOG P DIAGRAM

A-186
C.....SET AT(-1 F,1050NB) & (~82.5 F,500NB) ..........................C
    IPEN=-1
    DO 900 I=1,30
    IF(PRESS(I,N).LT.545) GO TO 900
    IX=42.8*(TEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
    IY=-34623.*ALOGT(PRESS(I,N))+104603.
    WRITE(12) IPEN,1,IX,IY
    IPEN=1
900  CONTINUE
    WRITE(12) -1.,-1.,100.,-150
    WRITE (12,9031) 100.,0.,0.,125
9031 FORMAT(415,7HAMBIENT)
    PAUSE
    IPEN=-1
    DO 925 I=1,30
    IF(PRESS(I,N).LT.545) GO TO 925
    IX=42.8*(CDTEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
    IY=-34623.*ALOGT(PRESS(I,N))+104603.
    WRITE(12) IPEN,1,IX,IY
    IPEN=1
925  CONTINUE
    WRITE (12,9032) 100.,0.,0.,125
9032 FORMAT(415,9HDEW POINT)
950    WRITE(12) -1.1,9999,9999
    DELP=1000
    DO 975 I=1,30
    IF(PRESS(I,N).LT.545) GO TO 975
    IF(PRESS(I,N).NE.DELP) GO TO 975
    DELP=DELP-50
    I2=3.28084*IALT(I,N)
    IY=-34623.*ALOGT(PRESS(I,N))+104603.
    WRITE(12,9036) 75.,0.,0.,100.,12
9036 FORMAT(415,15,3H FT)
    WRITE(12) -1.1,850.,IY
    WRITE (12) 1.1,900.,IY
    WRITE(12) -1.1,875.,IY
    WRITE(12,9020) 75.,0.,0.,100.,IALT(I,N)
9020 FORMAT(415,15,7H METERS)
975    CONTINUE
    WRITE(12) -1.1,250.,8500
    WRITE(12,9021) 100.,0.,0.,125
9021 FORMAT(415,8HALTITUDE)
    WRITE(12) -1.1,9999,9999
9010 FORMAT ("DATA NUMBER","&","&")
9015 FORMAT ("&dENTER FILE NUMBER IN 2 DIGIT I FORMAT"&d&8")
9140 FORMAT (7DX)
9190 FORMAT ("=============================================================================")
9220 FORMAT ("+-----------------------------------------------------------------------------")
9230 FORMAT (7X,"SPACE SHUTTLE LAUNCH FROM KSC")
9240 FORMAT ("TITAN IIIC LAUNCH FROM KSC")
9250 FORMAT ("E&dCLaunch time:" , I8, 9X, "Date:" , I2, 1X, 2A2, 1X, I4)
9260 FORMAT (" TIME OF SOUNDING: ",I8,4X,I2,1X,2A2,1X,I4)
9270 FORMAT (" TIME OF PREDICTION: ",I4)
9280 FORMAT ("DATA SET: ",I2,13X,"SURFACE DENSITY(GM/M**3)",F8.2)
9290 FORMAT ("E&dblayer altitude direction speed temp pot-temp 1E&d D P TEMP PRESSURE")
9300 FORMAT ("E&dn No. (FEET) (METERS) (DEGREES) (M SEC) (DEGREE CENTIG 1E&dNRADE) (MILLIBARS")
9310 FORMAT (I2, I7, 2X, I5, 7X, I3, 4X, F4.1, 4X, F4.1, 2X, F5.2, 2X, F4.1, 12X, F7.2)
9850 FORMAT (I2)
9860 FORMAT (26X, I4, 6X, I2, 1X, 2A2, I4)
9865 FORMAT (I4, 3X, I2, 1X, 2A2, I4)
9870 FORMAT (I6, 1X, 2F4.1, F6.1, F6.1, F7.2, 11X, F7.2)
9875 FORMAT (I6, 1X, 2F4.1, F6.1, F6.1, F7.2)
9999 END
SUBROUTINE IOHED(LHEAD,ITIMES)
DIMENSION LHEAD(40,60)
DO 110 I=1,ITIMES
READ (8,9855) (LHEAD(I,J), J=1,40)
WRITE (6,9855) (LHEAD(I,J), J=1,40)
110 CONTINUE
9855 FORMAT (40A2)
RETURN
END
Program SKEW T (Version II)
PROGRAM SKEW
DIMENSION IALT(31,3),IDIR(31,3),SPEED(31,3),TEMP(31,3),PRESS(31,3)
DIMENSION SUR(6),Y(31),ISWR(20),DPTMP(31,3),PTMP(31,3)
DIMENSION Q(30),A(23),B(23),E(15),F(4,4),G(4,4),K(23)
DIMENSION W(31,5),X(40,5),LHEAD(40),ALT(31,3),IDCB(256)
DIMENSION IBUF(80), NAME(3)
DATA NAME/'2HTE.2HMP.2HRD/
C IFMT = 0 DATA IS VAR PT FORMAT
C IFMT = 1 DATA IS FIX PT FORMAT
C IREAD = 0 DATA IS ON DISC
C IREAD = 1 DATA IS ON TAPE
C ** CONVERT RH AND TEMP TO DEMP **
SFY = 10.79.
D = 8.42926604
E = 0.071208271
C ** INITIALIZE DATA FORMATS AND INPUT UNITS **
IFMT = 0
IREAD = 0
WRITE(1,11)
11 FORMAT('E&B ENTER CASE NUMBER ')
WRITE(1,10)
10 FORMAT('E&B ENTER NUMBER OF POINTS ')
READ(1,20) N
WRITE(1,12)
20 FORMAT(I2)
WRITE(1,15)
15 FORMAT('E&B ENTER ISTIT ISDAY ISMON ISMYEAR ')
READ(1,16) ISTIT,ISDAY,ISMON,ISMYEAR
16 FORMAT(I4,I2,2A2,I4)
C ** INITIALIZE DATES **
LTIM = 1500
LDAY = 29
LMON = 2HC
LM = 2HT
LYEAR = 1975
IF(IREAD.EQ.0) CALL OPEN(IDCB,IER,NAME,0)
C CALL DATE
P9=3.14159
M=0
ISWR(1)=0
ISWR(2)=0
ISWR(3)=0
ISWR(4)=0
ISWR(5)=0
ISWR(6)=0
ISWR(7)=0
ISWR(8)=0
ISWR(9)=0
ISWR(10)=0
ISWR(11)=0
ISWR(12)=0
ISWR(15)=1
IUNIT=5
IF (ISWR(1) .EQ. 0) IUNIT=1
8B A(1)=0
IF (ISWR(8) .EQ. 1) GO TO 140
IF (IFMT .EQ. 1) WRITE(6,9010)
C : DEFINITION OF TERMS:
C TEMP(IALT,H)--TEMPERATURE; PRESS(IALT,H)--PRESSURE; DPTEMP(IALT,H)--HUMIDI
C
C ** LOAD DATA FROM TAPE **
IF (IREAD .EQ. 0) GO TO 121
ITIMES=5
CALL IOHED(LHEAD,ITIMES)
C READ <8,*> N
READ (8,9860) LTIM,LDAY,LMON,LM,LYEAR
ITIMES=19
CALL IOHED(LHEAD,ITIMES)
C WRITE (6,9015)
READ (1,9850) IFNO
CALL PTAPE (8,IFNO,0)
C T2=9999
ITIMES=3
CALL IOHED(LHEAD,ITIMES)
8000 READ (8,9865) ISTIM,ISDAY,ISMON,ISM,ISYEAR
IF (ISDAY .EQ. 0) GO TO 8000
WRITE(12) -1,1,1000,9750
WRITE(12,8040)175,0,0,200,ISTIM,ISDAY,ISMON,ISM,ISYEAR
8040 FORMAT(415,20HRAWINSONDE SOUNDING: I5,1HZ,2X,12,1X,2A2,14)
ITIMES=2
CALL IOHED(LHEAD,ITIMES)
READ (8,9870) IALT(I,N),IDIR(I,N),SPEED(I,N),TEMP(I,N),
IDPTEMP(I,N),PRESS(I,N),SURD(N)
DO 120 I=2,30
115 READ (8,9875) IALT(I,N),IDIR(I,N),SPEED(I,N),TEMP(I,N),
IDPTEMP(I,N),PRESS(I,N)
CALL EXEC(13,8,IEQT)
IEQT=IAND(IEQT,2008)
IF (IEQT .GT. 0) GO TO 140
IF (IALT(I,N) .LT. 10) GO TO 115
IF (IALT(I,N) .GT. 10000) GO TO 115
JARRAY=1
120 CONTINUE
IF (IREAD .EQ. 1) GO TO 140
C ** LOAD DATA FROM DISC **
CALL READF(IDCB, IERR, IBUF)
CALL CODE
READ(IBUF,301)
T2 = 9999
CALL READF(IDCB, IERR, IBUF)
CALL CODE
READ(IBUF,302) SURDEN(N)
WRITE(12) -1, 1, 1000, 9750
WRITE(12,8040) 175, 0, 0, 200, ISTIM, ISDAY, ISMON, ISM, ISYEAR
DO 122 I =1, NN
CALL READF(IDCB, IERR, IBUF)
CALL CODE
READ(IBUF,303) IALT(I, N), IDIR(I, N), SPEED(I, N), TEMP(I, N),
1 PRESS(I, N), DPTEMP(I, N)
IF(IALT(I, N).LT.1) GO TO 123
IF(IALT(I, N).GT.10000) GO TO 123
T = 1000./(TEMP(I, N) + 273.15)
E1 = (DPTEMP(I, N)/100.)*10.**-((C - D*T - E*T*T)
C1 = ALOGT(E1) - C
DT = (SQRT(D*D - 4*E*C1) - D)/(2*E)
DT = (1000./DT) - 273.15
DPTEMP(I, N) = DT
JARAY = I

CONTINUE

122 IF (IUNIT .EQ. 2 .OR. IUNIT .EQ. 3) WRITE(6,4020)
4020 FORMAT ("NEED JUMP")
C : N--DATA SET NUMBER; LMON, LDAY, LYEAR, LTIM---LAUNCH DATE/TIME
C :
C ; ISTIM--SOUNDING TIME; T2--PREDICTION TIME
C : CONVERTING SOUNDING TIME FROM ZULU TO EDT - AM , PM
C : ISTIM-ISTIM-400
ISTIM=ISTIM-400
IF(IFMT.EQ.0) ISTIM = ISTIM-200
IF (ISTIM .GT. 0) GO TO 250
ISTIM=2400-ABS(ISTIM)
ISDAY=ISDAY-1
250 IF (ISTIM .GE. 1300) GO TO 260
IF (ISTIM .GE. 1200 .AND. ISTIM .LT.1300) GO TO 270
GO TO 280
260 ISTIM=ISTIM-1200
270 CONTINUE
280 CONTINUE
C : SURDEN(N)=SURFACE DENSITY
C :
C : CONVERT DATA TO METRIC, SORT DATA BY IALT, CAL POT TEMP=PTEMP(IALT,N)
DO 590 I=1, JARAY
590 ALT(I, N)=IALT(I, N)
IF(IFMT.EQ.0) GO TO 509
ALT(I, N)=.3048*ALT(I, N)

A-192
SPEED(I, N) = 0.515 * SPEED(I, N)

L = I

509 IF (L .EQ. 1) GO TO 590
510 IF (ABS(ALT(L, N)) .GT. ALT(L-1, N)) GO TO 590
ALT(31, N) = ALT(L-1, N)
IDIR(31, N) = IDIR(L-1, N)
SPEED(31, N) = SPEED(L-1, N)
TEMP(31, N) = TEMP(L-1, N)
PRESS(31, N) = PRESS(L-1, N)
DPTEMP(31, N) = DPTEMP(L-1, N)
ALT(L-1, N) = ALT(L, N)
IDIR(L-1, N) = IDIR(L, N)
SPEED(L-1, N) = SPEED(L, N)
TEMP(L-1, N) = TEMP(L, N)
PRESS(L-1, N) = PRESS(L, N)
DPTEMP(L-1, N) = DPTEMP(L, N)
L = L-1

570 GO TO 510
590 CONTINUE

C ............... CALCULATE POTENTIAL TEMPERATURE (DEG K) PTEMP(ALT, N)........
DO 690 I = 1, JARAY
ALT(I, N) = ABS(ALT(I, N))
PTEMP(I, N) = ((TEMP(I, N) + 273.15) * (1000 / PRESS(I, N))**0.288)
690 CONTINUE

C ............... PRINT METEOROLOGICAL DATA ......................

J = J-1
C IF (ISWR(12) .EQ. 0) WRITE (6, *) "CTR PRINT"
C IF (ISWR(12) .EQ. 0) WAIT (15000)
IF (IFMT .EQ. 0) GO TO 727
WRITE (6, 9220)
WRITE (6, 9140)
WRITE (6, 9140)
WRITE (6, 9140)
IF (ISWR(15) .EQ. 0) WRITE (6, 9230)
IF (ISWR(15) .EQ. 1) WRITE (6, 9240)
727 IF (IFMT .EQ. 0) WRITE (6, 9333)
IF (IFMT .EQ. 0) WRITE (6, 9334)
IF (IFMT .EQ. 0) WRITE (6, 9335) LTIM, LDAY, LMON, LM, LYEAR
IF (IFMT .EQ. 0) GO TO 728
WRITE (6, 9250) LTIM, LDAY, LMON, LM, LYEAR
WRITE (6, 9140)
728 WRITE (6, 9260) ISTIM, ISDAY, ISMON, ISM, ISYEAR
WRITE (6, 9270) T2
E(6) = .66355
WRITE (6, 9280) H, SURDEN(H)
WRITE (6, 9140)
WRITE (6, 9290)
WRITE (6, 9300)
DO 850 I = 1, LJAY
SPEED(I, N) = INT(SPEED(I, N)*10)/10
IALTF = ALT(I, N)/.3048 + .5
IALT(I, N) = ALT(I, N) + .5
APTEHP(I) = PTEHP(I, N) - 273.15
WRITE (6, 9310) I, IALTF, IALT(I, N), IDIR(I, N), SPEED(I, N), TEMP(I, N), APTENP(I), DPTENP(I, N), PRESS(I, N)
850 CONTINUE

C
C.......... PLOT SKEW T , LOG P DIAGRAM..............................
C ** COMPUTE IY USING V1, V2, V3 FOR LOGRITHMIC PLOT CONVERSIONS **
V1 = 59470.457
V2 = -28656.688
V3 = 3078.846
C
C.......... SET AT(-1 F, 1050MB) & (.825 F, 500MB) ....................
C
IPEN = -1
DO 900 I = 1, NN
IF(PRESS(I, N) LT. 545) GO TO 900
IX = 142.8*(TEMP(I, N) + 273.15) - 10831.*ALOGT(PRESS(I, N)) - 3668.
IY = V1*(ALOGT(PRESS(I, N))) + V2*(ALOGT(PRESS(I, N)))**2 +
V3*(ALOGT(PRESS(I, N)))**3
IY = IY*SFY
WRITE(12) IPEN, 1, IX, IY
IPEN = 1
900 CONTINUE
WRITE(12) -1, -1, 100, -150
WRITE (12, 9031) 100, 0, 0, 125
9031 FORMAT (415, 7HAMBIENT)
C
PAUSE
IPEN = -1
DO 925 I = 1, NN
IF(PRESS(I, N) LT. 545) GO TO 925
IX = 142.8*(DPTENP(I, N) + 273.15) - 10831.*ALOGT(PRESS(I, N)) - 3668.
IY = V1*(ALOGT(PRESS(I, N))) + V2*(ALOGT(PRESS(I, N)))**2 +
V3*(ALOGT(PRESS(I, N)))**3
IY = IY*SFY
WRITE(12) IPEN, 1, IX, IY
IPEN = 1
925 CONTINUE
WRITE(12) -1, -1, 100, -150
WRITE (12, 9032) 100, 0, 0, 125
9032 FORMAT (415, 9HDEW POINT)
950 WRITE(12) -1, 1, 9999, 9999
DELP = 850

A-194
DO 975 I=1,NN
IF(PRESS(I,N).LT.545) GO TO 975
IF(PRESS(I,N).NE.DELP) GO TO 975
DELP=DELP-50
IZ=3.28084*IALT(I,N)
IY=V1*ALOGT(PRESS(I,N)) + V2*(ALOGT(PRESS(I,N)))*2 +
1V3*(ALOGT(PRESS(I,N)))*23
IY = IY*SFY
WRITE(12) -1,1,200,IY
WRITE(12,9030) 75,0,0,100,IZ
9030 FORMAT(415,15,3H FT)
WRITE(12) -1,1,850,IY
WRITE(12) 1,1,900,IY
WRITE(12) -1,1,875,IY
WRITE(12,9020) 75,0,0,100,IALT(I,N)
9020 FORMAT(415,15,7H METERS)
303 FORMAT(5X,14,7X,13,7X,F5.1,4X,F5.1,6X,F5.1,5X,F5.1,19X)
301 FORMAT(80X)
302 FORMAT(59X,F6.1,15X)
975 CONTINUE
WRITE(12) -1,1,250,9500
WRITE(12,9021) 100,0,0,125
9021 FORMAT(415,15,8H ALTITUDE)
WRITE(12) -1,1,9999,9999
9010 FORMAT (1X, "DATA NUMBER", " ", " ")
9015 FORMAT (1X, " ENTER FILE NUMBER IN 2 DIGIT I FORMAT ")
9140 FORMAT (70X)
9190 FORMAT (1X, " ===================================================")
9220 FORMAT (1X, " ++++++++++++++++++++++++ ++++++++++++++++++++++++++")
9230 FORMAT (7X, " SPACE SHUTTLE LAUNCH FROM KSC")
9240 FORMAT (1X, " TITAN IIIC LAUNCH FROM KSC")
9250 FORMAT (1X, " &d launch time: ", 18,9X," Date: ", I2,1X,2A2,1X,I4)
9260 FORMAT (1X, " TIME OF SOUNDING: ", 18,4X,I2,1X,2A2,1X,I4)
9270 FORMAT (1X, " TIME OF PREDICTION: ", I4)
9280 FORMAT (1X, " DATA SET: ", I2,13X," SURFACE DENSITY (GM/M**3)",F8.2)
9290 FORMAT (5X, " LAYER ALTITUDE", 2X, " DIRECTION", 1X, " SPEED", 3X,
1" TEMP", 1X, " DP-TEMP", 1X, " PRESSURE")
1X, " DEGREE CENTIGRADE", 1X, " MILLIBARS")
9333 FORMAT (1X, " ***************************************************************")
9334 FORMAT(1X, " SPACE SHUTTLE SRM DOTE PROGRAM TEST FIRINGS AT THIOKOL 1 WASATCH")
9335 FORMAT(1X, " LAUNCH TIME: ", 18,9X," DATE: ", I2,1X,2A2,1X,I4)
9310 FORMAT (1X, I2,17,2X,I5,7X,13,4X,F4.1,4X,F4.1,2X,F5.2,2X,F4.1,
12X,F7.2)
9850 FORMAT (I2)
9860 FORMAT (26X,I4,6X,I2,1X,2A2,I4)
9865 FORMAT (I4,3X,I2,1X,2A2,I4)
9870 FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2,11X,F7.2)
9875 FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2)

A-195
IF(IFMT.EQ.1) REWIND 8
IF(IFMT.EQ.0) CALL CLOSE(IDC)

999 END

SUBROUTINE IOHEH(LHEAD,ITIMES)
DIMENSION LHEAD(40)
DO 110 I=1,ITIMES
READ (8,9855)(LHEAD(N),N=1,40)
WRITE (6,9856)(LHEAD(N),N=1,40)
110 CONTINUE
9855 FORMAT (40A2)
9856 FORMAT (1X,40A2)
RETURN
END
END$
Program PUFF.
C GAMMA...JET EXIT ELEVATION ANGLE(DEG). HORIZONTAL IS ZERO.
C TE....JET EXIT TEMPERATURE(DEG-K)
C GAME...JET EXIT SPECIFIC HEAT RATIO
C MWE...JET EXIT MOLECULAR WEIGHT
C TINF...ATMOSPHERIC TEMPERATURE(DEG-K)
C PINF...ATMOSPHERIC PRESSURE(ATMOSPHERES)
C GAMINF.ATMOSPHERIC SPECIFIC HEAT RATIO
C MWINF.ATMOSPHERIC MOLECULAR WEIGHT
C APPP1..ENTRAINMENT COEFFICIENT BEFORE TAIL EXCEEDS 15 DIAMETERS
C APPP2..ENTRAINMENT COEFFICIENT AFTER TAIL EXCEEDS 15 DIAMETERS
C CD1..DRAG COEFFICIENT BEFORE TAIL EXCEEDS 15 DIAMETERS
C CD2..DRAG COEFFICIENT AFTER TAIL EXCEEDS 15 DIAMETERS
C THETA...JET EXIT AZIMUTH WITH RESPECT TO X-COORDINATE(DEG)
C VELW...WIND VECTOR COMPONENTS IN XYZ-COORDINATES(CM/SEC)

C OPEN INPUT DATA FILE &PUFFD
CALL OPEN(IDCB,IERR,NAME,0)
C READ AND PRINT OUT INPUT TEST DATA
WRITE(6,303)
303 FORMAT(1H1,"INPUT DATA IS AS FOLLOWS: ")
DO 320 I=1,5
   CALL BLANK(IBUF)
   CALL READF(IDCB,IERR,IBUF)
   CALL CODE
   READ(IBUF,301) ICARDS
301 FORMAT(40A2)
   WRITE(6,302) (ICARDS(N),N=1,40)
302 FORMAT(1H1,40A2)
320 CONTINUE
C REWIND AND READ INPUT DATA TO PROCESS
CALL RWNDF(IDCB,IERR)
CALL BLANK(IBUF)
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,304) TOFF,TMAX,DTI,UE
304 FORMAT(5(7X,F8.2))
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,304) GAMMA,TE,GAME,MWE,TINF
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,306) PINF,GAMINF,MWINF,APPP1,APPP2
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,307) CD1, CD2, THETA, VELW(1), VELW(2)
CALL READF(IDCB,IERR,IBUF)
CALL CODE
READ(IBUF,305) VELW(3), IPRINT,IFLAG,IUNITS
305 FORMAT(7X,F8.0,3(7X,I8))
306 FORMAT(5(7X,F8.5))
307 FORMAT(5(I7,F8.3))
C WRITEC(6,388) T0FF,TMAX,DTI,R,UE
C WRITEC(6,388) GAMMA,TE,GAME,MWE,TINF
C WRITEC(6,388) PINF,GAMINF,MWINF,APPP1,APPP2
C WRITEC(6,388) CD1,CD2,THETA,VELW(1),VELW(2)
C WRITEC(6,389) VELW(3),IPRINT,IFLAG,UNITS
C388 FORMAT(1H,"SAMPLE INPUT",/,'5E20.6')
C389 FORMAT(1H,'E20.6',5I10)
VELW(2) = 0.
APPP=APPP1
CD=CD1
C COMPUTE SOME OTHER INVARIANT PARAMETERS
D15=30.*R
PE=PINF
PC=PE
RE=UGC/MWE
RINF=UGC/MWINF
CP=RE*GAME/(GAME-1.0)
CPINF=RINF*GAMINF/(GAMINF-1.0)
RHOE=PE/RE/TE
RHOINF=PINF/RINF/TINF
TRAD=THETA/RAD
GRAD=GAME/RAD
UEX=UE*COS(GRAD)*COS(TRAD)
UEY=UE*COS(GRAD)*SIN(TRAD)
UEZ=UE*SIN(GRAD)
AE=PI*R **2
MEDOT=RHOE*AE*UE
C ESTABLISH OUTPUT CONSTANTS
C1=1.0
C2=1.0
C3=1.0
IF (IUNITSEQ.0) GO TO 13
C1=0.01
C2=1.0E-6
C3=0.001
13 CONTINUE
C WRITEC(6,200) UGC,MEDOT,RHOE,AE,RHOINF,RE,RINF,CP,CPINF
200 FORMAT("","9E12.5")
WRITEC(6,210)
C INTEGRATE FOR TMAX SECONDS, PRINT EVERY IPRINT STEPS.
10 CONTINUE
DT=0.1
J=J+1
IPR=IPR+1
C ALWAYS USE DT=DTO DURING FIRST 0.1 SEC OF JET ON
DT=DTI
C IF (T.GE.0. AND. T.LT.0.099) DT=DTO
IF(T.GE.0. AND. T.LT.0.1.) DT = DTO
CALL RK4
C IF TAIL LENGTH GT 15 DIAMETERS, CHANGE ENTRAINMENT COEF  
IF (LT.LT.D15.OR.APPP.EQ.APPP2), GO TO 11  
APPP=APPP2  
CD=CD2  
WRITE(6,120)  
120 FORMAT("O CLOUD TAIL EXCEEDS 15 DIAMETERS. ENTRAINMENT COEF INCREASES")  
WRITE(6,210)  
11 CONTINUE  
IF (T.LT.TOFF.OR.IOFF.EQ.1), GO TO 12  
WRITE(6,130) TOFF  
130 FORMAT(1H ,"JET SHUT OFF AT T = ",F6.2," SEC")  
WRITE(6,210)  
2 7X,"ME",6X,"MINF",6X,"VOL")  
IOFF=1  
12 CONTINUE  
IF (IPR.LT.IPRINT), GO TO 10  
C WRITE INTEGRATION VARIABLES AND CLOUD DIMENSIONS  
IPR=0  
C SET MODE TO 1  
MODE=1  
C CALL EVAL1(MODE)  
IF (IFLAG.EQ.0), GO TO 9  
WRITE(6,210)  
9 CONTINUE  
C CONVERT TO METERS, KG IF REQUIRED  
X=V(6)*C1  
Y=V(7)*C1  
Z=V(8)*C1  
VX=DV(6)*C1  
VY=DV(7)*C1  
VZ=DV(8)*C1  
ELT=LT*C1  
XT=V(9)*C1  
YT=V(10)*C1  
ZT=V(11)*C1  
ELS=(LT+H-CR)*C1  
RCLD=CR*C1  
M1=ME*C3  
M2=MINF*C3  
VOL=VC*C2  
WRITE(6,220), T,X,Y,Z,VX,VY,VZ,TC,XT,YT,ZT,ELT,ELS,RCLD,M1,M2,VOL  
220 FORMAT(" ",F5.2,6F7.2,F7.0,6F7.2,3F9.1)  
IF (T.LT.TMAX), GO TO 10  
CALL CLOSE(IDC,B,IERR)  
STOP  
END
SUBROUTINE BLANK(IBUF)
DIMENSION IBUF(40)
DATA IBLK/2H /
DO 66 N=1,40
  IBUF(N) = IBLK
RETURN
END

SUBROUTINE DERIV
REAL MWE, MEDOT, MWINF, LT
COMMON V(11), DV(11), T, DT, NV, G(3), RAD, PI, R, UGC, MEDOT, TOFF, TC,
  1ME, MINF, VC, RHOE, CPE, RE, GAME, MWE, TE, PE, UEX, UEY, UEZ, UE,
  2HC, CD, RHOINF, CPIINF, RINF, GAINF, MWINF, TINF, PINF, THETA,
  3GAMMA, PC, APPP, YELW(3), H, CR, LT, XC, ACS, ASP, IFLAG
REAL MA, MC, ME, M1X, M1Y, M1Z, M2X, M2Y, M2Z, M3X, M3Y, M3Z, MINF, MOMX,
  1MOMY, MOMZ, MDTINF
EQUIVALENCE (V(1), MC), (V(2), MOMX), (V(3), MOMY), (V(4), MOMZ),
  1(V(5), EC), (V(6), CGX), (V(7), CGY), (V(8), CGZ), (V(9), STX),
  2(V(10), STY), (V(11), STZ)
DIMENSION UINF(3)
DATA DXL, DYL, VCL, SCGL/4*0. /
MODE=0
IF (T.NE.0.0) GO TO 10

COMPUTE INITIAL DERIVATIVES
DV(1)=MEDOT
DV(2)=MEDOT*UEX
DV(3)=MEDOT*UEY
DV(4)=MEDOT*UEZ
DV(5)=MEDOT*CPE*TE
DV(6)=UE*0.5
DV(7)=0.
DV(8)=0.
DV(9)=0.
DV(10)=0.
DV(11)=0.
ME=0.
RETURN

CALL EVALI(MODE)
RETURN
END

SUBROUTINE EVALI(MODE)
COMMON V(11), DV(11), T, DT, NV, G(3), RAD, PI, R, UGC, MEDOT, TOFF, TC,
  1ME, MINF, VC, RHOE, CPE, RE, GAME, MWE, TE, PE, UEX, UEY, UEZ, UE,
  2HC, CD, RHOINF, CPIINF, RINF, GAINF, MWINF, TINF, PINF, THETA,
  3GAMMA, PC, APPP, YELW(3), H, CR, LT, XC, ACS, ASP, IFLAG
REAL MWE, MEDOT, MWINF, LT
REAL MA, MC, ME, M1X, M1Y, M1Z, M2X, M2Y, M2Z, M3X, M3Y, M3Z, MINF, MOMX,
  1MOMY, MOMZ, MDTINF
EQUIVALENCE (V(1), MC), (V(2), MOMX), (V(3), MOMY), (V(4), MOMZ),
  1(V(5), EC), (V(6), CGX), (V(7), CGY), (V(8), CGZ), (V(9), STX),
  1(V(10), STY), (V(11), STZ)
DIMENSION UINF(3)
DATA DXL,DYL,YCL,SCGL/4*0./
C THIS ENTRY USED TO FIND CLOUD SHAPE. NOT USED WHEN INTEGRATING.
C MODE=1
10 CONTINUE
C AT TOFF SET MEDOT=0 + HOLD ME CONSTANT
IF (T.GE.TOFF) GO TO 12
ME=MEDOT*T
GO TO 14
12 CONTINUE
IF (MEDOT.EQ.0) GO TO 14
ME=MEDOT*TOFF
MEDOT=0.
14 CONTINUE
MINF=MC-ME
CPC=(MINF*CP INF+ME*CPE)/MC
TC=EC/(MC*CPC)
RC=(MINF*RINF+ME*RE)/MC
RHOC=PC/(RC*TC)
VC=MC/RHOC
DX=CGX-STX
DY=CGY-STY
DZ=CGZ-STZ
SCG=SQR T(DX*DX+DY*DY+DZ*DZ)
C CALL CLOUD SHAPE SUBROUTINE TO GET ACS
VZ=VC
CALL SHAPE(VZ,SCG)
C IF MODE=1, EVALUATE CLOUD SHAPE BUT NO DERIVATIVES REQUIRED.
IF (MODE.EQ.1) GO TO 20
MA=0.5*RHO INF*VC
CALL WIND (CGZ,UINF)
UCX=(MOMX+MA*UINF(1))/(MC+MA)
UCY=(MOMY+MA*UINF(2))/(MC+MA)
UCZ=(MOMZ+MA*UINF(3))/(MC+MA)
UC=SQR T(UCX**2+UCY**2+UCZ**2)
ELS=(LT+H-CR)/SCG
CSX=STX+DX*ELS
CSY=STY+DY*ELS
CSZ=STZ+DZ*ELS
ELT=LT/SCG
SLX=STX+DX*ELT
SLY=STY+DY*ELT
SLZ=STZ+DZ*ELT
SL=SQR T(SLX*SLX+SLY*SLY+SLZ*SLZ)
ST=SQR T(STX**2+STY**2+STZ**2)
URX=UINF(1)-UCX
URY=UINF(2)-UCY
URZ=UINF(3)-UCZ
UR=SQR T(URX**2+URY**2+URZ**2)
MDTINF=RHO INF*UR*ACS*APP
C H = H E D O T * U E X
C M I = H E D O T * U E Y
C M I Z = H E D O T * U E Z
C C O N S T = 2.0 * M A - M C
C M 2 X = G ( 1 ) * C O N S T
C M 2 Y = G ( 2 ) * C O N S T
C M 2 Z = G ( 3 ) * C O N S T
C C O N S T = U R * A S P * C D * R H O I N F
C M 3 X = U R X * C O N S T
C M 3 Y = U R Y * C O N S T
C M 3 Z = U R Z * C O N S T
C E 1 = H E D O T * C P E * T E
C E 2 = M D T I N F * C P I N F * T I N F
C S B C O N = 1.355 E - 12 * 41.293
C E M I S S = 0.4
C E 3 = E M I S S * S B C O N * A C S * ( T I N F * 4 - T C * 4)
C E 4 = 2.0 * P I N F * M E D O T / R H O I N F
C D V ( 1 ) = M E D O T + M D T I N F
C D V ( 2 ) = M 1 X + M 2 X + M 3 X
C D V ( 3 ) = M 1 Y + M 2 Y + M 3 Y
C D V ( 4 ) = M 1 Z + M 2 Z + M 3 Z
C D V ( 5 ) = E 1 + E 2 + E 3
C D V ( 6 ) = U C X
C D V ( 7 ) = U C Y
C D V ( 8 ) = U C Z
C I F ( T . L T . T O F F ) G O T O 3 0
C D V ( 9 ) = U E X * D X * L T / ( S T X * S C G + D X * L T + 1.0 E - 9 ) + U C X
C D V ( 1 0 ) = U E Y * D Y * L T / ( S T Y * S C G + D Y * L T + 1.0 E - 9 ) + U C Y
C D V ( 1 1 ) = U E Z * D Z * L T / ( S T Z * S C G + D Z * L T + 1.0 E - 9 ) + U C Z
C 3 0 C O N T I N U E
C I F ( I F L A G . E Q . 0 ) G O T O 9
C F D E L T A = M 3 X - M A * D V ( 6 )
C V B = V C / R E ** 3
C X B = X C G / R E
C W R I T E D E B U G O U T P U T
C 9 C O N T I N U E
C 1 0 0 F O R M A T ( 1 H 0 , " M E , M I N F , C P C , T C , R C , R H O C = ", 6 E 1 5 . 5 / )
C 1 " M E , M I N F , C P C , T C , R C , R H O C = ", 6 E 1 5 . 5 /
C 3 " M 1 Y , M 1 Z , M 2 X , M 2 Y , M 2 Z , M 3 X , M 3 Y , M 3 Z , E 1 , E 2 , E 3 , E 4 = ", 6 E 1 5 . 5 /
C 4 " M 3 Y , M 3 Z , E 1 , E 2 , E 3 , E 4 = ", 6 E 1 5 . 5 /
C 6 " X B , V B , F D E L T A , A C S , T , S T = " , 6 E 1 5 . 5 )
C 2 0 C O N T I N U E
C R E T U R N
C E N D
C S U B R O U T I N E W I N D ( H , U I N F )
DIMENSION UINF(3)
COMMON DUM(24), IDUM6, DUM7(37), YELW(3), DUM8(6), IDUM8
DO 100 I=1,3
UINF(I) = YELW(I)
100 CONTINUE
RETURN
END

SUBROUTINE SHAPE(VZ, CG)
DIMENSION XCOF(4), COF(4), ROOT(3), ROOTI(3)
DOUBLE PRECISION Q, QQ, A, B, Y1, PBY3CU, ROOTQ, XBAR, PBAR, RE6
REAL LT
COMMON DUM1(24), IDUM9, G(3), RAD, PI, R, DUM2(9),
IDUM3(25), H, CR, LT, XC, ACS, ASP, IFLAG
DATA RELAST/0., H/0./
C COMPUTE CONSTANTS ONLY ONCE UNLESS RE CHANGES VALUE
IF (R.EQ.RELAST) GO TO 2
RELAST=R
RE2=R*R
RE4=RE2*RE2
RE6=RE4*RE2
DE=2.0*R
WRITE(6,341) RELAST, R, RE2, RE4, RE6, DE
341 FORMAT(1H "RELAST, R, RE2, RE4, RE6, DE", /, 1H , 6E20.6)
SE=PI*RE2
SEINV=1.0/SE
CON1=0.5/(SE*DE)
PBY3CU=(-RE4)**3
WRITE(6,342) SE, PI, SEINV, CON1, PBY3CU, XCOF(1)
342 FORMAT(1H "SE, PI, SEINV, CON1, PBY3CU, XCOF(1)", /, 1H , 6E20.6)
XCOF(2)=SE/24.0
XCOF(3)=0.
XCOF(4)=-PI*PI/(72.0*SE)
WRITE(6,343) XCOF(2), XCOF(3), XCOF(4)
343 FORMAT(1H "XCOF(2), XCOF(3), XCOF(4)", /, 1H , 6E20.6)
2 CONTINUE
XBAR=CG/DE
PBAR=VZ*CON1
Q=-576.*PBAR*(PBAR-XBAR)*RE6
QQ=PB3CU+(Q*0.5)**2
WRITE(6,940) Q, QQ, PBY3CU, PBAR, XBAR, RE6
940 FORMAT(1H "Q, QQ, PB3CU, PBAR, XBAR, RE6", /, 1H , 6E20.6)
IF (QQ.LT.0.) GO TO 20
ROOTQ=DSQRT(QQ)
A=(DABS(-Q*0.5+ROOTQ )**0.3333333
B=(DABS(-Q*0.5-ROOTQ )**0.3333333
Y1=A+B
H=DSQRT(Y1)
IF (IFLAG.EQ.0.) GO TO 30
Y2=-(A+B)*0.5
Y3=(A-B)*0.5*1.73205

A-205
WRITE(6,105) Y1,Y2,Y3
105 FORMAT(" ",20X,"Y1,Y2,Y3=",3(1PE12.4))
GO TO 30
20 CONTINUE
XCOF(1)=YZ*(YZ*0.5*SEINV-CG)
M=3
WRITE(6,9302) XCOF(1)
9302 FORMAT(1X,"XCOF1=",E12.5)
CALL POLRT(XCOF,COF,M,ROOTR,ROOTI,IER)
C FIND SMALLEST POSITIVE REAL ROOT
RMIN=1E10
DO 10 I=1,M
IF(ROOTI(I).EQ.0.0.AND.ROOTR(I).LT.RMIN.AND.ROOTR(I).GT.0.) RMIN=
1 ROOTR(I)
10 CONTINUE
IF (IFLAG.EQ.0) GO TO 6
DX=1.0/(288.*PBAR)
XMAX=PBAR+DX
XMIN=PBAR-DX
C TEST NATURE OF ROOTS
IF (XBAR;LT.XMAX.AND.XBAR.GT.XMIN.AND.XBAR.GT.PBAR) WRITE(6,101)
IF (XBAR.LT.XMAX.AND.XBAR.GT.XMIN.AND.XBAR.LT.PBAR) WRITE(6,102)
IF((XBAR.GT.XMAX.OR.XBAR.LT.XMIN).AND.XBAR.GT.PBAR) WRITE(6,103)
IF((XBAR.GT.XMAX.OR.XBAR.LT.XMIN).AND.XBAR.LT.PBAR) WRITE(6,104)
101 FORMAT("
3 REAL ROOTS, 2 POSITIVE, 1 NEGATIVE",58X,"REAL",6X,1 "IMAJ")
102 FORMAT("
3 REAL ROOTS, 1 POSITIVE, 2 NEGATIVE",58X,"REAL",6X,1 "IMAJ")
103 FORMAT("
1 REAL ROOT, NEGATIVE
1 "IMAJ")
104 FORMAT("
1 REAL ROOT, POSITIVE
1 "IMAJ")
WRITE(6,100)XBAR,XMAX,XMIN,PBAR,IER,((ROOTR(I),ROOTI(I)),I=1,M)
100 FORMAT(1H , "BAR,XMAX,XMIN,PBAR,IER,ROOTS=",4(1PE12.4),I3,8X,1 2E12.4.5(/90X,2E12.4))
6 CONTINUE
C IF NO POSITIVE ROOTS, WRITE ERROR MESSAGE
IF (RMIN.EQ.1E10) GO TO 40
H=SQRT(RMIN)
30 CONTINUE
CR=0.5*(CR+R/RH)
LT=Y2*SEINV-0.5*H-PI*H**3/(6.*SE)
XC=H+LT-CR
ASP=PI*CR**2
ACS=2.*PI*(R*LT+CR*H)
C DO NOT INCLUDE LT IN SURFACE AREA IF IT IS NEGATIVE
IF (LT.LT.0.) ACS=2.*PI*CR*H
C CORRECT PROJECTED AREA IF NOT SPHERICAL
IF (XC.LT.LT) ASP=SE
RETURN
40 CONTINUE
10.6 FORMAT(*PROGRAM HALT. POSITIVE ROOT FOR N NOT FOUND.*)
WRITE(6,100)XBAR, XMAX, XMIN, PBAR, IER, (ROOTR(I),ROOTI(I)), I=1,3
WRITE(6,106)
STOP
END
SUBROUTINE RK4
DIMENSION OLD(11), B(11)
C THIS IS A 4TH ORDER RUNGE-KUTTA INTEGRATOR
COMMON Y(11), DV(11), T, DT, NV, DUCK(46), IDUCK
DATA J/0/
OLDT=T
DO 2 J=1, NV
2 OLD(J)=Y(J)
CALL DERIV
T=OLDT+0.5*DT
DO 4 J=1, NV
B(J)=DT*DV(J)
4 Y(J)=OLD(J)+0.5*B(J)
CALL DERIV
DO 6 J=1, NV
TMP=DT*DV(J)
B(J)=B(J)+2*TMP
6 Y(J)=OLD(J)+0.5*TMP
CALL DERIV
DO 8 J=1, NV
T=OLDT+DT
CALL DERIV
DO 10 J=1, NV
10 Y(J)=OLD(J)+(B(J)+DT*DV(J))/6
RETURN
END
SUBROUTINE POLRT(XCOF, COF, M, ROOTR, ROOTI, IER)
DIMENSION XCOF(1), COF(1), ROOTR(1), ROOTI(1)
DOUBLE PRECISION X0, Y0, X, XPR, YPR, UX, UY, V, YT, XT, U, XT2,
1 YT2, SUMSQ, DX, DY, TEMP, ALPHA
IFIT=0
M=M
IER=0
IF(XCOF(N+1))10, 25, 10
10 IF(N) 15, 15, 32
C SET ERROR CODE TO 1
15 IER=1
20 RETURN
C SET ERROR CODE TO 4
25 IER=4
GO TO 20
C SET ERROR CODE TO 2
30 IER=2
GO TO 20

32 IF(H-36) 35,35,30
35 NX=N
NXX=N+1
N2=1
KJ1=N+1
DO 40 L=1,KJ1
MT=KJ1-L+1
40 COF(MT)=XCOF(L)
C SET INITIAL VALUES
C
45 XO=.00500101
YO=0.01000101
ZERO INITIAL VALUES COUNTER
C
50 IN=0
50 X=XO
C INCREMENT INITIAL VALUES AND COUNTER
C
XO=-10.0*YO
YO=-10.0*X
C
C SET X AND Y TO CURRENT VALUE
C
X=XO
Y=YO
IN=IN+1
GO TO 59
55 IFIT=1
XPR=X
YPR=Y
C EVALUATE POLYNOMIAL AND DERIVATIVES
C
59 ICT=0
60 UX=0.0
UY=0.0
V=0.0
YT=0.0
XT=1.0
U=COF(N+1)
IF(U) 65,130,65
65 DO 70 I=1,N
L=N-I+1
TEMP=COF(L)
XT2=X*XT-Y*YT
YF = X \cdot YT + Y \cdot XT
U = U + \text{TEMP} \cdot XT \cdot YT2
V = V + \text{TEMP} \cdot YT \cdot YT2
FI = 1
UX = UX + FI \cdot XT \cdot TEMP
UY = UY - FI \cdot YT \cdot TEMP
XT = XT \cdot YT2
SUMSQ = UX \cdot UX + UY \cdot UY
IF(SUMSQ) 75, 110, 75
DX = (V \cdot UY - U \cdot UX) / SUMSQ
X = X + DX
DY = -(U \cdot UX + V \cdot UY) / SUMSQ
Y = Y + DY
IF(DABS(DY) + DABS(DX) - 1.0D-05) 100, 80, 80
C STEP ITERATION COUNTER
C ICT = ICT + 1
IF(ICT - 500) 60, 85, 85
IFT IFIT) 100, 90, 100
IFT IFIT - 5) 50, 95, 95
C SET ERROR CODE TO 3
C IER = 3
GO TO 20
100 DO 105 L = 1, NXX
MT = KJ1 - L + 1
TEMP = XCOF(MT)
XCOF(MT) = COF(L)
105 COF(L) = TEMP
ITEMP = N
N = NX
NX =ITEMP
IF(IFIT) 120, 55, 120
110 IF(IFIT) 115, 50, 115
115 X = XPR
Y = YPR
120 IFIT = 0
122 IF(DABS(Y) - 1.0D-04 * DABS(X)) 135, 125, 125
125 ALPHA = X + X
SUMSQ = X \cdot X + Y \cdot Y
N = N - 2
GO TO 140
130 X = 0.0
NX = NX - 1
135 Y = 0.0
SUMSQ = 0.0
ALPHA = X
N=N-1
140 COF(2)=COF(2)+ALPHA*COF(1)
145 DO 150 L=2,N
150 COF(L+1)=COF(L+1)+ALPHA*COF(L)-SUMSQ*COF(L-1)
155 ROOTI(N2)=Y
ROOTR(N2)=X
N2=N2+1
IF(SUMSQ) 160,165,160
160 Y=-Y
SUMSQ=0.0
GO TO 155
165 IF(N) 20,20,45
END
END$
Environmental Effects from SRB Exhaust Effluents — Technique Development and Preliminary Assessment


Science Applications, Inc.
2109 W. Clinton Avenue, Suite 800
Huntsville, Alabama 35805

National Aeronautics and Space Administration
Washington, D.C. 20546

Prepared under the sponsorship of the Atmospheric Diffusion/Environmental Effects Technical Task Team, Dr. J. B. Stephens (Aerospace Environment Division, Space Sciences Laboratory, NASA/Marshall Space Flight Center), Technical Task Leader

This report documents results of a study which had as its primary objective the development of techniques to determine the environmental effects from the Space Shuttle SRB exhaust effluents. The study developed many new and needed tools which were used to perform a preliminary climatological assessment. This preliminary study will be used to guide the future full-scale climatological assessment. The exhaust effluent chemistry study was performed and, neglecting several possibly important effects, the exhaust effluent species determined. A reasonable exhaust particle size distribution has been constructed for use in future nozzle analyses and for the deposition model. The effects of scavenging and absorption were not included in the preliminary assessment. The preliminary assessment was used to identify problems that may be associated with the full-scale assessment; therefore, these preliminary air quality results should be used with caution in drawing conclusions regarding the environmental effects of the Space Shuttle exhaust effluents.

17. Key Words
Rocket Motor Exhaust Diffusion
Rocket Exhaust Cloud Rise
Space Shuttle Environmental Studies
Atmospheric Diffusion
REED Description

18. Distribution Statement
Category 34

19. Security Classification (of this report)
Unclassified

20. Security Classification (of this page)
Unclassified

21. No. of Pages
340

22. Price
$10.00

* For sale by the National Technical Information Service, Springfield, Virginia 22161

*U.S. GOVERNMENT PRINTING OFFICE: 1977 - 735-078/49