DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
ABSTRACT

This manual provides a complete and detailed description of the dynamic thermodynamic sea ice model (PIFS-N), including all the necessary information for maintenance programmer personnel required to maintain the system.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section 1</th>
<th>General Description</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Purpose of the Maintenance Manual</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Terms and Abbreviations</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 2</th>
<th>System Description</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>System Application</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Security and Privacy</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>General Description</td>
<td>5</td>
</tr>
<tr>
<td>2.4</td>
<td>Program Description</td>
<td>8</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Program ICEMDL</td>
<td>10</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Input Model Subroutines</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2.1</td>
<td>Subroutine MESH</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2.2</td>
<td>Subroutine BNDRY</td>
<td>26</td>
</tr>
<tr>
<td>2.4.2.3</td>
<td>Subroutine OCEAN</td>
<td>28</td>
</tr>
<tr>
<td>2.4.2.4</td>
<td>Subroutine INITIAL</td>
<td>29</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Processing Modules Subroutines</td>
<td>32</td>
</tr>
<tr>
<td>2.4.3.1</td>
<td>Subroutine XSUM</td>
<td>32</td>
</tr>
<tr>
<td>2.4.3.2</td>
<td>Subroutine FORM</td>
<td>32</td>
</tr>
<tr>
<td>2.4.3.3</td>
<td>Subroutine PLAST</td>
<td>34</td>
</tr>
<tr>
<td>2.4.3.4</td>
<td>Subroutine RELAX</td>
<td>36</td>
</tr>
<tr>
<td>2.4.3.5</td>
<td>Subroutine FELLIP</td>
<td>38</td>
</tr>
<tr>
<td>2.4.3.6</td>
<td>Subroutine FELLD1</td>
<td>39</td>
</tr>
<tr>
<td>2.4.3.7</td>
<td>Subroutine ADVECT</td>
<td>40</td>
</tr>
<tr>
<td>2.4.3.8</td>
<td>Subroutine DIFFUS</td>
<td>41</td>
</tr>
<tr>
<td>2.4.3.9</td>
<td>Subroutine HEAT</td>
<td>43</td>
</tr>
<tr>
<td>2.4.3.10</td>
<td>Subroutine BUDGET</td>
<td>47</td>
</tr>
<tr>
<td>2.4.3.11</td>
<td>Subroutine GROWTH</td>
<td>48</td>
</tr>
<tr>
<td>2.4.3.12</td>
<td>Subroutine ADJUST</td>
<td>50</td>
</tr>
<tr>
<td>2.4.3.13</td>
<td>Subroutine MEAN</td>
<td>51</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Output Module Subroutines</td>
<td>52</td>
</tr>
<tr>
<td>2.4.4.1</td>
<td>Subroutine DIVERG</td>
<td>52</td>
</tr>
<tr>
<td>2.4.4.2</td>
<td>Subroutine UVPLOT</td>
<td>53</td>
</tr>
<tr>
<td>Section</td>
<td>Subroutine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.4.4.3</td>
<td>Subroutine BUOYD</td>
<td>54</td>
</tr>
<tr>
<td>2.4.4.4</td>
<td>Subroutine HAPLOT</td>
<td>56</td>
</tr>
<tr>
<td>2.4.4.5</td>
<td>Subroutine GROWDEC</td>
<td>57</td>
</tr>
<tr>
<td>2.4.4.6</td>
<td>Subroutine PRNT</td>
<td>58</td>
</tr>
<tr>
<td>3</td>
<td>Environment</td>
<td>59</td>
</tr>
<tr>
<td>3.1</td>
<td>Equipment Environment</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Support Software</td>
<td>59</td>
</tr>
<tr>
<td>3.3</td>
<td>Data Base</td>
<td>59</td>
</tr>
<tr>
<td>3.3.1</td>
<td>General Characteristics</td>
<td>60</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Organization and Detailed Description</td>
<td>61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Program Maintenance Procedures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Conventions</td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td>Verification Procedure</td>
<td>64</td>
</tr>
<tr>
<td>4.3</td>
<td>Error Conditions</td>
<td>64</td>
</tr>
<tr>
<td>4.4</td>
<td>Special Maintenance Procedures</td>
<td>66</td>
</tr>
<tr>
<td>4.5</td>
<td>Special Maintenance Programs</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Listings</td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX A  ICEMDL Flowcharts
APPENDIX B  Grid Structure
APPENDIX C  Listings
SECTION 1  GENERAL DESCRIPTION

1.1 Purpose of the Program Maintenance Manual

The objective of this Program Maintenance Manual for the Dynamic Thermodynamic Sea Ice Model, PIFS-N, is to provide the maintenance programmer personnel with the information necessary to effectively maintain the system.

1.2 Background

Sea ice forecasting programs have been clearly connected with fleet polar operations since the early 1950's. When organizing and conducting the sea lift operations required for the establishment and resupply of arctic bases such as Thule, Greenland and the Distant Early Warning Line across Alaska and Canada, ice reconnaissance and forecasting services were requested.

Since the advent of the underice submarine operations in 1957, ice intelligence prediction services have been requested by submarine commands. Their operational forecast requirements deviated markedly from surface ship or icebreaker requirements. The submarine operator is primarily interested in knowing the distribution, in frequency and size, and depth of ice pressure ridges which may constitute a hazard to underice navigation particularly in shoal water.

The surface ship is primarily interested in knowledge of ice concentration, the distribution of the various stages of ice development and floe sizes.

Escalations of operational activities
in ice-covered waters over the past seven years with attendant navigation problems have focused Navy and national attention on the need for a more reliable sea ice forecasting program.

The recent establishment of the Navy NOAA Joint Ice Center is a strong indication that Navy and other governmental departments are moving forward to meet existing and full operational requirements.

In recent years, important contributions have been made to understand the dynamics of sea ice through the Arctic Ice Dynamics Joint Experiment (AIDJEX). Several mathematical models have been developed. A continuing effort is needed to evaluate the applicability of these models for sea ice forecasting.

The dynamic-thermodynamic Sea Ice Model was developed by the W. D. Hibler III (Hibler, 1979). The Naval Air Systems Command (NAIR-270G) tasked the Naval Ocean Research and Development Activity's Polar Oceanography Branch to implement the model utilizing the Fleet Numerical Oceanography Center (FNOC) environmental data base and Cyber 203.

The model uses atmospheric forecast and analysis data available at FNOC in the operational data base. The model outputs forecasts of ice drift, concentration, thickness, convergence/divergence plus ice and open water growth.
There have been a number of sea ice models appearing in the literature in the recent years. A list of applicable references to this model is presented below:


vi) FNWC User Manual


1.3 Terms and Abbreviations

FNOC

Fleet Numerical Oceanography Center

NORDA

Naval Ocean Research and Development Activity

NPOC

Naval Polar Oceanography Center

NEDN

Naval Environmental Data Network
Section 2  

SYSTEM DESCRIPTION

2.1  
System Application

Fleet Numerical Oceanography Center is a large computer complex, tasked with the mission of providing worldwide environmental support to the U.S. fleets. To accomplish this mission, FNOC collects meteorological report data, analyzes the data and predicts changes in environmental conditions.

The Dynamic Thermodynamic Sea Ice model developed by Hibler is modified to use FNOC environmental data. The modified model gets input data from the FNCC data base. These data consist of wind data, surface pressure, surface vapor pressure, air specific humidity, air temperature, incoming and outgoing radiation on the FNOC 63 x 63 Northern Hemisphere grid (A01, A10, A11, A12, A16, A16, A20, A21), interpolates these data to the model grid to get initial boundary conditions.

The model computes ice drift, ice concentration, thickness, convergence/divergence rate plus thick and thin ice growth/decay rate.

2.2  
Security and Privacy

The sea ice model does not currently have access to classified information nor does it produce classified output. Hence, security should be treated at appropriate levels, and the user is responsible for protection of any material used by the model.
2.3 General Description

The dynamic thermodynamic sea ice model has been modified to run on the Cyber 203 utilizing the FNOC environmental data base, FNOC library and plotting facility. In general, the modified model can be divided into three modules: Input Module, Processing Module, Output Module.

Input module creates a model grid on the FNOC 63 x 63 Polar Stereographic grid, sets boundary and initial conditions for the model and obtains and interpolates input data to each model grid point.

The processing module can be divided into two parts which process two different mechanisms: the dynamic mechanism and the thermodynamic mechanism.

The heart of the dynamic part is subroutine RELAX which computes ice drift for the model.

The thermodynamic mechanism is processed by subroutine HEAT which estimates ice thickness, concentration and thick and thin ice growth/decay rate.

The output module calls many FNCC routines to format output data into CRANDIC format, and stores them for plotting and printing. A functional diagram of the model is shown in Figure 1. Figure 2 illustrates more detail of each functional division.
Figure 1

The functional diagram of the Dynamic Thermodynamic Sea Ice Model

Input Module
Create the model grid
Set up boundary and initial conditions
Get input data

Processing Module
Dynamic Part:
Calculate ice drift
Thermodynamic Part:
Calculate ice thickness, coverage, growth/decay rate

Output Module
Store output on CRANDIO
for plotting and print output
Figure 2

The chart of interrelationship of the major components of the system

Create the model grid
Set up boundary and initial conditions
Get input data: winds, dynamic height, currents, surface pressure

Momentum Equations
Numerical finite difference solution to obtain ice velocities
Viscous plastic sea ice constitutive law

Dynamic thermodynamic evolution of ice thickness characteristics
Estimation of ice strength from the ice thickness and concentration

Thermodynamic Model
Thermodynamic input: air temperature, solar radiation, air pressure

Output: ice thickness, drift, coverage, ice growth rate, stored in ZRANDIO. and printed
2.4 **Program Description**

The following list of subroutine names and functions is provided for easy reference:

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM ICEMDL</td>
<td>Main driving program for the model</td>
</tr>
</tbody>
</table>

**Input Module**

<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESH</td>
<td>Calculates FNOC (I, J) grid points for the model grid;</td>
</tr>
<tr>
<td>BNDRY</td>
<td>Sets up boundary masks;</td>
</tr>
<tr>
<td>OCEAN</td>
<td>Computes ocean surface currents;</td>
</tr>
<tr>
<td>INITIAL</td>
<td>Obtains initial atmospheric data.</td>
</tr>
</tbody>
</table>

**Processing Module**

<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM</td>
<td>Sets up forces, drag coefficients, non-linear viscosities for use in each time step;</td>
</tr>
<tr>
<td>XSUM</td>
<td>Sums a vector;</td>
</tr>
<tr>
<td>PLAST</td>
<td>Calculates non-linear viscosities for plastic flow, used in FORM</td>
</tr>
<tr>
<td>UVDDFF</td>
<td>Converts U, V to direction and magnitude and vice versa</td>
</tr>
<tr>
<td>INTRP</td>
<td>Performs interpolation from the FNOC grid to the model grid;</td>
</tr>
</tbody>
</table>
RELAX Solves linearized momentum balance with spatially varying bulk and shear viscosities. Uses sequential over-relaxation technique;

FELLIP Calculates finite differences for use in RELAX;

FELLDI Estimates thickness and concentration at "open boundary" grid cells based on adjacent values;

ADJUST Performs advection of ice thickness and concentration;

DIFFUS Drives subroutine for thermodynamic calculations;

HEAT Computes thin and thick ice growth/decay rate;

BUDGET Computes averaged input variables for staggered grid system;

GROWTH Calculates changes in concentration and mean ice thickness due to growth and redistribution due to ridging.

Output Module

SUBROUTINE DIVERG Output convergence/divergence in CRANDIO format;

UVPLT Outputs direction and magnitude of ice drift in CRANDIC format;

HAPLOT Outputs ice thickness and concentration in CRANDIO format

PRNT Provides hard copy print out of model variable arrays;

STATPRT Prints resource use statistics;

GROWPEC Outputs ice growth/decay rates in CRANDIO format.
2.4.1 **Program ICEMDL**

A brief description of the theory involved in this model is initially presented in order to help clarify and provide a reference for the subroutine descriptions subsequently presented.

The main driving program, ICEMDL, is designed to call the subroutines which set up the initial conditions, perform the equation integration and output the specified forecasts in CRANDIC format on the CDC CYBER 203 at FNOC.

The overall structure essentially consists of three main components:

1. input and initial conditions;
2. equation integration;
3. output processing.

The initial conditions are determined from the atmospheric data available through the FNCC operational data base. A model grid is defined to cover the arctic ocean basin and is a subset of the FNCC hemispheric grid (Figure 1). The grid contains a square mesh with a distance of approximately 120 Km between grid points. Atmospheric variables are interpolated from the FNOC hemispheric grid to the model grid through the use of a 16 point Bessel interpolation scheme. The following atmospheric parameters are interpolated to the model grid:

1. surface wind \((u,v)\);
2. surface pressure;
3. surface vapor pressure;
4. surface air temperature;
5. incoming solar radiation;
6. total heat flux at the surface;
7. sensible plus evaporative heat flux at the surface.
Input processing begins by determining the boundary masks to be used in the simulation. The boundary masks define the coastline configuration present within the model grid system. Further output processing involves the incorporation of the initial ocean currents and initial surface wind components. Other input variables listed above are obtained during the processing of the thermodynamic module which accesses the input module to obtain the needed atmospheric thermodynamic variables.

The overall structure of the processing model can also be considered to exist in the three main portions defined below. The first is a momentum balance which includes air and water stresses, coriolis force, internal ice stress and ocean tilt. Non-linear boundary layers for both the air-ice and ocean-ice surfaces are used. A key component is the force due to internal ice stress. This is defined by a constituitive law which relates the ice stress to the strain rate and ice strength. For this model a viscous-plastic constituitive law is followed.

The second feature of the processing module consists of continuity equations describing the evolution of the thickness characteristics on two levels. Two categories of ice thickness are assumed; thin ice (less than 0.5 m in thickness) and thick ice (greater than 0.5 m in thickness). To keep account of these categories two variables are maintained; ice thickness per unit area and the ice concentration which is defined as the fraction of area of a grid cell covered by thick ice. Thermodynamic terms are included on these continuity calculations.

The final component is an ice strength value which is taken to depend linearly upon the ice thickness and exponentially on the ice concentration.
The coupled non-linear equations are treated as an initial value problem using energy conserving finite differences. The momentum equations are integrated implicitly in order to avoid a time limit constraint. A relaxation technique is used on the set of simultaneous equations at each time step.

The numerical scheme uses a staggered grid which allows ice strength and ice velocities to vary in space. To a large degree this staggered grid is patterned after those used in primitive equation ocean models.

As mentioned above, initial conditions at all points and ice velocities at the boundaries are thereafter required to initiate the integration of the system of equations. The most natural condition is to take the ice velocity to be zero at the boundary. This can be accomplished at land boundaries or open ocean boundaries where there is no ice. This boundary condition does not affect the ice motion in such circumstances since, in the absence of ice the strength is zero. It is also possible to set an "open" boundary condition by setting the strength equal to zero near a boundary. This type of open boundary is used at the Spitzbergen-Greenland passage to form a natural inflow/outflow region (Figure 1).

In the computer code, three boundary masks are used to define the "closed" and "open" boundaries. Consequently by altering these masks, highly irregular boundaries may be taken into account.

Because of the strong ice interaction, the momentum equations are parabolic in form and hence have few numerical instability problems over longterm integrations. To avoid non-linear instabilities in longterm simulations which can arise from the non-linear advection terms in the continuity equations, small
biharmonic and harmonic terms are added to the continuity equation.

A thermodynamic system is incorporated which specifies the ice growth rate as a function of thickness and time of year.

**Input to ICEMDL**

Input to the ICEMDL program exists on a number of files which must be set up before the actual model is run. Only a portion of these files are actually accessed by PROGRAM ICEMDL. The remainder of the input files is accessed by input module subroutines.

The input files are labeled as TAPE7=IN and TAPE8=DATE in the CY203 convention. TAPE8 contains the date time groups of the period for the model integration. The date time group (DTG) is the standard format defined at FNOC consisting of:

YYMMDDHH
where
YY = year;
MM = month;
DD = day;
HH = hour.

Each DTG is kept in one word and is valid for one day (time step). The DTG is used to access the proper fields from the CRANDIC data base consisting of the atmospheric data maintained by FNOC. Three DTG values are input to ICEMDL at each time step requiring new atmospheric data. The initial DTG values input for the very first time step are set up differently than the other DTG values. For all accesses of the DTG file the
first DTG contains the DTG of the current day valid for that time step. The second DTG contains the DTG for 12 hours before the current day. For the initial group of DTG's the third position contains the DTG of the last time step of the previous run. This DTG is used to read the final fields produced by the previous run. These fields are used to restart the model simulation. Therefore, if a run was made to simulate the month of January (ending January 31, 1981) and a new run was to begin for February the initial line of the TAPE8 file would be;

\[81020100\ 81013112\ 81013100.\]

The DTG, 81020100, is the current DTG for the start of the simulation. The DTG 81013100 is the last day of the previous simulation and will be used to access the data for restarting the model. In reality input atmospheric data is entered into the model every 4 days (time steps). Therefore the DTG values, held in the model, change in increments of 4 days.

The file TAPE7 contains various types of input data used by the model. TAPE7 is accessed by ICEMDL to set the following variables:

\[\begin{align*}
NX, NY & \quad \text{dimension of the grid which holds the momentum variables;} \\
NXL, NYL & \quad \text{dimension of the grid which holds the thermodynamic variables;} \\
N3 & \quad \text{number of points in the momentum variable grid;} \\
N4 & \quad \text{number of points in the thermodynamic variable grid.}
\end{align*}\]
ICEMDL Processing

The processing within ICEMDL begins by accessing the input files described in 2.4.1.1. The next processing involves the accessing of input module subroutines. Subroutine MESH is the first input module routine which is accessed. This routine defines the model grid in terms of the FNOC hemispheric grid. MESH returns the i,j values of the model grid defined in terms of hemispheric grid.

Subroutine BNDRY is accessed after MESH. BNDRY returns the boundary masks to ICEMDL.

Subroutine OCEAN is called to formulate the u, v components of the surface ocean currents and returns them to ICEMDL.

At this point, some basic housekeeping tasks are performed such as presetting several arrays to zero and accessing COPEN which is an FNOC routine which "opens" the CRANDIC file to be accessed by the model.

An error condition develops if the routine COPEN finds that it is unable to open the CRANDIC files. The program will terminate, stating that there is a COPEN ERROR on the STOP line of the dayfile.

Subroutine INITIAL is the final input module routine accessed by ICEMDL. Two calls are made to obtain the u and v atmospheric wind components respectively.

The next section of ICEMDL accesses the CRANDIC data base written by previous runs of the model. The variables; ice thickness, ice concentration and ice drift are read to restart the model simulation.

At this point all necessary information is available for the start of processing. The above described code is never executed again during the current run.
The actual equation integration starts with a series of calls to routines FORM and RELAX. FORM computes all parameters necessary for use in the relaxation scheme. These include the air and water drag coefficients, forces terms, ice strength terms, and the viscosity terms. Subroutine RELAX performs the relaxation technique.

There is a sequence of two calls to these routines. The first performs the prediction portion of the momentum time step. Before the first RELAX call, the third level of ice velocities and the centered ice velocities, held in UICEI, VICEC, are set equal to the level of ice velocities by Hibler (1980). During this procedure, the time step is halved. FORM is called in this first step to use the present ice velocity values to linearize the momentum equations.

The second calls of FORM and RELAX amount to the main forward time step of the "corrector" section of this "predictor-corrector" method. In this case the "predicted" values of the ice velocities are used in the second FORM call to estimate the viscosity parameters.

After the momentum equations have been implicitly stepped forward using the relaxation technique, several diagnostic calculations are carried out. The squared ice velocity and the squared ice velocity difference between the times, t and t+1 are computed. This is a simple measure of the change taking place in the ice velocity field during the time step advance.

The predicted ice velocity values are contained in the first level of the UICE and VICE arrays.

The ice velocity and divergence values are passed to the output module routines, UVPLLOT and
DIVERG respectively.

Following this momentum time step, the thermodynamic equations are explicitly stepped forward in time. Subroutine ADVECT is called to handle the dynamical portions of the continuity equations for ice thickness and concentration. Subroutine HEAT is called to obtain the thick and thin ice growth rates through the use of a heat budget. Subroutine GROWTH is called to heat the thermodynamic portion of the continuity equations.

This concludes processing for the time integration. the remainder of the code is used to monitor diagnostic variables and the output modules.

Subroutine XSUM is used to obtain the total ice held within the grid, excluding outflow cells. The following diagnostic values are then computed;

i) Total open water growth for each grid cell; HDIFF;

ii) Net open water growth for the basin; GRSUM1;

iii) Net ice growth for the basin; FHSUM;

iv) Total ice in the outflow cells; TOUT.

The ice held in the outflow cells is explicitly determined through variables, THEFF and THEFF1. The variable THEFF1 contains the amount of ice in the open cells and is computed at the beginning of the time step.

The output module begins with a list of PRINT statements which form the hard copy output. Hard copy output is printed every four time steps. The
variable, LSTEP, is used to count time steps and branch to the output section on the fourth time step.

Subroutine PRNT is used to print the model arrays.

Subroutine ADJUST is called after the printed output is formulated. ADJUST is used to define the amount of ice held in the outflow cells for use at the beginning of the next time step.

Subroutine GROWDEC and HAPLOT are used in the output module to output the ice thickness, ice concentrations and growth/decay rates to the CRANDIO file maintained by the model.

The final function performed in ICEMDL is to decide if processing is complete. The variable ITSTEP which is input from TAPE8 defines the total number of time steps minus one to be used in the run. The variable ICOUNT is used to keep track of how many steps have currently been executed.

If more time steps are required, a check is made to determine whether new atmospheric wind data is needed. New wind data is used every four days. The variable, LSTEP, defines the fourth time step as described above. Subroutine INITIAL is called to access the CRANDIO file containing the wind data if needed. The program continues processing until time steps are no longer desired.

If the number of desired time steps has been completed, the diagnostic variables of:

i) outflow;
ii) net ice growth;
iii) net open water growth;

are written to the file TAPE3 for use in a restart run if
desired. Subroutine STATPRT is called to print time use statistics on various routines used.

**ICEMDL Output**

The output of ICEMDL consists of forecasts of the following variables:

i) ice thickness;
ii) ice concentration;
iii) ice drift;
iv) ice divergence/convergence;
v) ice strength;
vi) ice growth.

These variables are output in printed form and also in CRANDIC files on the Cy203. CRANDIC is the type of file used operationally at FNOC, on the Cy203, for maintenance of the environmental data base. CRANDIC is analogous to ZRANDIO on the Cy170's and 6600's at FNOC.

The output CRANDIO files contain one record, produced every four time steps, for each of the above 6 variables. Specific information as to the format and structure of a CRANDIC file can be found in the appropriate FNOC technical write-up.

The records are labeled with the catalog name of each variable. The date of the record, and a tau value. The catalog names are defined as:

i) FFF - ice speed;
ii) DDD - ice direction;
iii) THK - ice thickness;
iv) CON - ice concentration;
v) PRS - ice strength;  
vi) HDF - ice growth;  
vii) GAR - open water growth;  
viii) DIV - convergence/divergence.

**Interfaces**

Program ICEMDL interfaces with subroutines which comprise the input, processing and output modules. The following table defines all interfaces between these routines and ICEMDL. Specific arrays are defined under Tables and Items.

<table>
<thead>
<tr>
<th>Subroutine Name</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESH</td>
<td>receives - NX, NY, NX1, NY1</td>
</tr>
<tr>
<td></td>
<td>NUMBER - number of points in grid;</td>
</tr>
<tr>
<td></td>
<td>returns - GRDI - i grid points;</td>
</tr>
<tr>
<td></td>
<td>GRDJ - j grid points;</td>
</tr>
<tr>
<td>BNDRY</td>
<td>receives - NX, NY, NX1, NY1</td>
</tr>
<tr>
<td></td>
<td>returns - HEFFM - thermodynamic variables</td>
</tr>
<tr>
<td></td>
<td>boundary mask;</td>
</tr>
<tr>
<td></td>
<td>UVM - momentum variables boundary mask;</td>
</tr>
<tr>
<td></td>
<td>OUT - outflow boundary mask;</td>
</tr>
<tr>
<td>OCEAN</td>
<td>receives - NX, NY, GRDI, GRDJ, UVM;</td>
</tr>
<tr>
<td></td>
<td>returns - GWATX - ocean current u components</td>
</tr>
<tr>
<td></td>
<td>GWATY - ocean current v components;</td>
</tr>
<tr>
<td>COPEN</td>
<td>receives - IFILE - CRANDIO file name</td>
</tr>
<tr>
<td></td>
<td>returns - ISTAT - status of file open attempt;</td>
</tr>
</tbody>
</table>

20
INITIAL receives - file unit number, NUMBER, GRDI, GRDJ, IDTG, DTG array, ITAU - tau value;
returns - variables read from FNOC database;
CREADER receives - IFILE - array for data;
LABEL - record name;
catalog name;
record length;
returns - read status, IS;
DDFFUN receives - DD, FF - direction and force of current or drift;
returns - U, V components;
XSUM receives - HEFF - ice thickness array NX1, NY1;
returns - total of array - THEFF or THEFF1;
PRNT receives - array name;
dimensions of array;
positions to be printed;
FORM receives - UICE, VICE, ETA, ZETA, AMASS, GAIRX, GAIRY, GWATX, GWATY,
OUT, HEFFM, NX, NY, NX1, NY1, HEFF, AREA;
returns - DRAGS, DRAGA, DIV;
RELAX receives - UICE, VICE, ETA, ZETA, AMASS, GAIRX, GAIRY, GWATX, GWATY,
DRAGS, DRAGA, OUT, HEFFM, NX, NY, NX1, NY1, HEFF, AREA;
returns - UICE, VICE;
DIVERG receives - DIV, NX, NY, GRDI, GRDJ, IDTG, ITAU;
UVPLOT receives - UICEC, VICEC, GRDI, GRDJ, NX, NY;
ADVECT receives - NICEC, VICEC, HEFF, DIFFI, LAD, HEFFM, NX, NY, NX1, NY1; returns - HEFF or AREA, DIFFI;

HEAT receives - GRDI, GRDJ, HEFF, AREA, GAIRX, Gairy, ITAU, IDTG, NX1, NY1, NUMBER;
returns - FC, FHEFF;

GROWTH receives - HEFF, AREA, HC, A22, FHEFF, FO, HCCOR, HEFFM, OUT, NX1, NY1;
returns - HEFF, AREA, GAREA;

HAPLOT receives - HEFF, AREA, ITAU, GRDI, GRDJ, NX, NY;

GRCDWDEC receives - HDIFF, FHEFF, GAREA, IDTG, ITAU, NX1, NY1;

ADJUST receives - HEFF, AREA, OUT, HEFFM, NX, NY, NX1, NY1;
returns - HEFF, AREA.

CEMDL Tables and Items

The following lists define all common blocks and major variable items used in ICEMDL.

I. Common Blocks

/BUOY/ BUCYI, BUCYJ - Buoy position in terms of the PNCC I, J grid;
BX, BY - Buoy positions in terms of the model x, y grid;

/FORCE/ FORCEx - x component of external force plus ice pressure gradient;
FORCCEY - y component of external force plus ice pressure gradient;
/STEP/  DELTAT - Time step in seconds;
DELTAx, - mesh size in meters, x, y
DELTAy  directions respectively;
/PRESS/ IDENT - CRANDIC record identification
          block;
DATA    - hemispheric atmospheric data;
FILL    - filler to put the block on small
          page boundary (required by CRANDIC
          software);
/IJ/    IJ63 - I,J grid points of SKILES current
          data;
/DD/    ID - CRANDIC identification block for
          ice drift direction;
DD      - ice drift directions;
FILLD   - filler for small page boundary;
/FF/    IF, FF, - Same as /DD/ except for ice drift
          speeds;
/AR/    IDA, ARRAY, - Same as /DD/ except for ice
          concentration.

Variable Items:
UICE    - u component of ice velocity
VICE    - v component of ice velocity
ETA     - non linear shear viscosity
ZETA    - non linear bulk viscosity
GAI RX  - u component of wind
GAI RY  - v component of wind
AMASS   - ice mass per unit area
HEFF    - mean ice thickness per unit area
AREA    - fraction of area covered by thick ice
UICEC   - Intermediate ice velocities for
          use in semi-implicit time step
VICEC   - u component of ocean currents
GWATX   - v component of ocean currents
GWATY   - v component of ocean currents
STRESS - XX, YY and XY components of ice stress
FHEFF - growth rate of thick ice
FO - growth rate of thin ice
DRAGA - water drag plus Coriolos parameter
DRAGS - water drag plus inertial term
DIV - ice convergence/divergence

2.4.2 Input Module Subroutines

2.4.2.1 Subroutine MESH

Subroutine MESH is used to create the model grid. The grid is constructed as a subset of the FNCC hemispheric grid.

Input to MESH

Subroutine MESH receives input from 2 sources. The first source is the formal parameters from ICEMDL. MESH also reads the input file labeled TAPE7. The following variables are contained in the formal parameter list:

i) GRDI, GRDJ - computed in MESH, and contain the i, j coordinates of the model grid

ii) NX, NY, NX1, NY1, N - grid sizes, defined in TAPE 7.

The following variables are obtained from the input file TAPE7:
i) \( I_0, I_1 \) - Defining grid points on the FNOC hemispheric grid;

ii) \( J_0, J_1 \) - Defining grid points on the FNOC hemispheric grid;

iii) \( N \) - The number of points in each row and column in the model grid.

Subroutine MESH creates a square grid. The variables \( I_0, I_1, J_0, J_1 \) are defined as follows:

\[
\begin{align*}
\text{x} & \quad \text{x} \\
\text{i, j} & \quad I_1, J_1 \\
\text{x} & \quad \text{x} \\
I_0, J_0 & \quad i, j
\end{align*}
\]

Therefore \( I_0, J_0 \) define the bottom left corner of the desired model grid and \( I_1, J_1 \) define the upper right corner of the model grid. MESH fills in the remainder of the grid, depending upon \( N \). All \( i, j \) coordinates are in reference to the hemispheric, 63 x 63 grid of FNOC.

**Processing**

The processing of subroutine MESH begins by reading all necessary input data from TAPE7. From the variable, \( N \), the values of \( NX, NY \) and \( NX_1, NY_1 \) are computed. The actual grid is then formulated using the specified corner points and the number of points desired in each row (column). The column points are defined, stored in GRDJ, followed by the row points stored in GRDI.
After the entire grid is formed, the MESH size is computed utilizing the characteristics and map factor of the polar stereographic grid. The map factor of the model grid is also computed.

The final function of the routine is to compute the initial buoy positions in terms of the newly constructed model grid.

Output

Subroutine MESH produces printed output specifying the following:

i) Corner grid points;
ii) mesh size of model grid;
iii) map factor;
iv) buoy grid locations.

Interfaces

Subroutine MESH does not call any other subroutines.

Tables and Terms

All common blocks and major variables, used in MESH are defined under the ICEMDL section.

2.4.2.2 Subroutine BNDRY

Subroutine BNDRY is called to set up the boundary masks for the thermodynamic, momentum and outflow grids. A boundary mask consists of an array which
contains a 1 in grid locations where computations are performed and a 0 on all boundary points. By altering these masks one can obtain any desired boundary or coastline configuration.

Input

The respective boundary masks are read from the input file labeled TAPE7. The grid sizes are input through the formal parameter list interfaced with ICEMDL.

Processing

The standard FORTRAN READ function is used to access TAPE7 and move the boundary masks to the respective locations.

Output

The arrays, UVM, HEFFM and OUT are created by BNDRY.

Interfaces

Subroutine BNDRY does not call any other subroutines.

Tables and Items

The following major variables are used;

UVM - boundary mask for momentum variables;
HEFFM - boundary mask for thermodynamic variables
CUT - boundary mask containing outflow points.

2.4.2.3 Subroutine OCEAN

Subroutine OCEAN is used to define the surface ocean currents. These currents are defined by the SKILES sea ice drift model used at FNOC.

Input

Ocean current directions are read from the input file TAPE7. These direction values are on the model grid. The values were interpolated from the SKILES grid to the current grid and the results placed on TAPE7.

The model grid size and i, j coordinates of the model grid are passed to OCEAN from ICEMDL through the formal parameter list.

Processing

The ocean current magnitudes are located in the DATA statement defining the array WF. The ocean current magnitudes are input from TAPE7 through the use of a standard READ. The current magnitudes are converted from cm/sec to m/sec and placed into a model grid. The current directions, on TAPE7, also needed to be multiplied by 10. For example a direction of 270 is read in as 27.

The current directions and magnitudes are converted to u, v components with the FNOC routine.
DDFFUV. Details pertaining to the function of this routine are contained in the standard FNOC subroutine write-ups. The code is placed within this program because the library is not available to the CY203 at the time of this writing.

Output

The arrays GWATX, GWATY are produced in OCEAN.

Interfaces

Subroutine OCEAN interfaces with subroutine DDFFUV which is a standard FNOC library routine used to convert a direction and magnitude to u, v components.

Tables and Items

All major variables contained in OCEAN are defined under ICEMDL.

2.4.2.4 Subroutine INITIAL

Subroutine INITIAL is used to access FNOC CRANDIC data on the CY203 and place the specified atmospheric data into the model grid.

Input

The main input to INITIAL is obtained from ICEMDL through the formal parameter list. The
variable, NO, specifies which field is to be accessed from the CRANDIO data base. Value of NO specifies which position in array IRCD is to be used. IRCD holds the catalog names of the various atmospheric data input to the model. The array, IDTG, contains the data time group to be used in reading the data. ITAU is the tau value and N is the number of rows (columns) in the model grid. The arrays GRDI, GRDJ are defined as in MESH.

Processing

The first section of code in INITIAL is setting up the CRANDIC record name. This is a 2 word ASCII label. The first word contains the following:

i) Catalog name of data;
ii) flaps character;
iii) length of record.

The second word contains the DTG. All of this information is masked together in array LABEL. The FNOC routine, CHECKNC, is used to determine whether the specified data is present on the CRANDIC file. If not present the program terminates with the following message:

STOP CHECKNC NO DATA INITIAL.

Providing the data is present, the FNOC routine CREADER is used to read the hemispheric data into array, DATA.

Subroutine INTRP is used to interpolate the hemispheric data to the model grid. INTRP is an FNOC subroutine which performs the interpolating function.
A detailed description of the operations of INTRP is available in the FNOC utility subroutine documentation.

**Outputs**

Subroutine INITIAL provides for a hard copy print of the requested record label. The array, GAIR, is used to store the resultant data placed in the model grid.

**Interfaces**

Subroutine INITIAL interfaces with the CRANDIO data base software at FNOC plus a utility library routine, INTRP. Detailed descriptions on all these routines are available as standard products of FNOC.

**Tables and Items**

The major variables, used in INITIAL, are defined in ICEMDL. The array IRCD is used to maintain the respective catalog names of the CRANDIO data. These catalog names are defined as follows:

i) A20 - u wind components;
ii) A21 - v wind components;
iii) A10 - atmospheric temperature at the surface;
iv) A01 - surface pressure;
v) A12 - surface vapor pressure;
vi) A11 - shortwave radiation;
vii) A18 - total heat flux;
viii) A16 - sensible plus evaporative heat flux.
2.4.3  Processing Module

2.4.3.1  Subroutine XSUM

All input to XSUM is provided by ICEMDL, through the parameter list.

**Processing**

Subroutine XSUM computes a simple sum of an array.

**Output**

The sum, contained in SI, is produced by XSUM.

**Interfaces**

No subroutines are called by XSUM.

**Tables and Items**

No new major variables are defined in XSUM.

2.4.3.2  Subroutine FORM

Subroutine FORM is used to calculate the drag coefficients, external forces and ice strength parameters for use in the time integration of the momentum equations.
Input

Subroutine FORM receives all required input from ICEMDL through the formal parameters. All of the variable definitions passed to FORM are presented under ICEMDL. Certain constants are defined as follows:

i) FCOR - average coriolis parameter;
ii) RHOAIR - air density;
iii) SINWIN - sine of air turning angle;
iv) COSWIN - cosine of air turning angle;
v) SINWAT - sine of ocean turning angle;
vii) COSWAT - cosine of ocean turning angle;
vii) ECCEN - ratio of the axes of the plastic yield curve.

Processing

Subroutine FORM initially calculates the variables required to obtain the external forces operating upon the ice. Within the first major loop the ice mass per unit area, coriolis, non-linear water stress coefficient and antisymmetric water drag, DRAGA, are computed. The next major loop calculates the non-linear air stress and the symmetric water drag term, DRAGS. These terms are calculated in a separate loop due to a different grid requirements of the GAIRX, GAIRY (wind component) term (See Appendix A).
The remaining portion of the subroutine deals with finalizing the external force terms and accessing Subroutine PLAST to compute non-linear shear and bulk viscosities respectively. Before PLAST is called, the ice strength, PRESS, is computed. After PLAST is called the computed viscosities are set to zero at the outflow points by multiplying by the boundary mask OUT.

Finally, the external force components term is combined with the ice pressure gradient.

Output

The results of processing in routine FCRM are stored in DRAGS, DRAGA, FORCEx, FORCEy, ETA, ZETA, and PRESS. These variables are defined under ICEMDL.

Interfaces

Subroutine FCRM interfaces with routine PLAST. Subroutine PLAST calculates the non linear viscosities based on a Plastic yield curve.

Tables and Items

Subroutine FCRM operates upon a number of major variables which are defined under ICEMDL.

2.4.3.3 Subroutine PLAST

Subroutine PLAST is used by subroutine FORM to calculate the non linear viscosity terms based on a plastic yield curve.
Input

Subroutine PLAST receives all input parameters from FORM, through the formal parameter list.

Processing

The first main loop of PLAST uses the ice drift components to calculate the XX, XY, YY strain rates of the ice (E11, E12, E22 respectively). These components are used, with the constitutive law to calculate the non-linear bulk viscosity. The bulk viscosity is in turn used to compute the non-linear shear viscosity.

The final computation within PLAST is the calculation of the XX, XY, YY components of ice stress plus the divergence is calculated from the XX, YY strain rates.

Output

The main output of PLAST is contained in the arrays ETA, ZETA and sent to FORM through the parameter list.

Interfaces

Subroutine PLAST calls no other subroutines.

Tables and Items

The main variables of PLAST which have not been identified previously are;

i) E11, E12, E22 - XX, XY, YY strain rates respectively.
2.4.3.4 Subroutine RELAX

Subroutine RELAX is the main routine of the processing module. This routine applies a relaxation technique to the dynamical equations for their numerical integration in time. Much numerical detail is contained in the routine and described by Hibler (1979).

Input

Subroutine RELAX receives all input from ICEMDL through the parameter list.

Processing

The processing of RELAX is broken into a number of separate modules. The first 3 major loops perform operations involving the previous time values of u, v plus the evaluation of the diagonal components of the computation matrix.

The main loop (103) performs the iterative relaxation scheme. This loop performs all necessary calculations to obtain a new estimate of the u, v ice drift components (UICE, VICE). After the new components are calculated, 2 checks are made. The first check examines the number of iterations completed to determine if more than 1300 have taken place. If so, the routine shall end printing a message stating more than 1300 iterations have occurred with no convergence. The second check searches for the 100th iteration. At this point the routine switches from a over relaxation scheme to a straight relaxation scheme.

After the checks have been executed, the difference between the new solution and previous
iterative solution is computed. If the difference lies within an accepted tolerance the routine ends. If the difference does not meet the tolerance specification another iteration is performed. The old iteration value is stored in the third level of arrays UICE, VICE while the new iterative solution is placed in the first level of arrays.

The relaxation code is made more complex by the separation of the finite difference computations into subroutines, FELLIP and FELLDI. The code is also generalized to fit into the predictor-corrector scheme previously defined. The parameter, THETA, defines whether backwards, centered, or forward time steps are used. A value of 0.5 initiates a centered time and a value of 0 dictates a forward step.

Outputs

A printed output message is made at the completion of the relaxation procedure. The message states the number of iterations used and the value of the difference between iterations at the end. The results of the processing are placed in UICE, VICE, level 1.

Interfaces

Subroutine RELAX interfaces with routines FELLIP and FELLDI. These routines calculate finite differences used in the relaxation routine.

Tables and Items

A large number of variables are used internally by RELAX. All major variables, however, are defined under the description of ICEMDL.
Subroutine FELLIP

Subroutine FELLIP is used to calculate finite differences used in the relaxation technique.

Input

All required input to FELLIP is provided in the parameter list and passed to FELLIP from RELAX.

Processing

Subroutine FELLIP operates on one grid position at a time. The position is defined by the input variable, \( i, j, k \). FELLIP is called a number of times producing various terms needed by the relaxation routine at each call.

The code remains a very straight-forward calculation of the finite difference approximations of the specified terms.

Output

The array, \( F \), holds all resultant calculations.

Interfaces

No further subroutines are called by FELLIP.

Tables and Items

No new major variables are used in FELLIP.
2.4.3.6  **Subroutine FELLD1**

Subroutine FELLD1 is also used to calculate finite differences for the relaxation code.

**Inputs**

All required input to FELLD1 is supplied in the parameter list by RELAX.

**Processing**

Unlike FELLIP, FELLD1 operates upon the entire grid during one call. The level of the array to be altered is specified by input variable, K (e.g., 1, 2, or 3). The input variable, S1, defines the differencing constants.

**Outputs**

The output of FELLD1 is held within the array, F.

**Interfaces**

No subroutines are called by FELLD1.

**Tables and Items**

No new major variables are used in FELLD1.
2.4.3.7 Subroutine ADVECT

Subroutine ADVECT handles the dynamical portions of the thermodynamic continuity equations.

Input

Subroutine ADVECT receives all necessary input from ICEMDL through the formal parameter list. These variables are defined under ICEMDL and any functions performed by these variables are defined below.

Processing

Subroutine ADVECT performs the explicit time stepping procedure of the dynamical portions of the thermodynamic continuity equations. The routine is designed to operate in two separate finite difference schemes. The input variable, LAD, determines whether a backward Euler or Leapfrog scheme is followed. If LAD is 1.0 then the leapfrog scheme is followed, otherwise the Backward Euler is used.

Initially the ice thickness (HEFF) and ice concentrations (AREA) are stepped forward in time by transferring the grid point values to the next lower level. Therefore, the current values are moved to level 2 of the array while the new values are put into level 1. The subroutine is called twice by ICEMDL, one time processing HEFF and secondly processing AREA.

The finite difference approximation to the respective variable and a following check on the finite differencing scheme are the next major processing actions. If the Backward Euler scheme is used (LAD is 0) the scheme is continued to finish the necessary
computations required by this scheme.

After the computation is complete for both schemes, the subroutine DIFFUS is called to calculate a smoothing term which is used to comprise the final data value.

Output

The array HEFF, which is internal to ADVECT, holds the final calculations. These values are transferred to HEFF and AREA in ICEMDL.

Interfaces

Subroutine ADVECT utilizes routine DIFFUS for calculation of a smoothing operator used to reduce the effort of non linear instabilities arising from non linear terms in the continuity equations.

Tables and Items

No major variables are used in ADVECT which have not been previously defined.

2.3.3.8 Subroutine DIFFUS

Subroutine DIFFUS is used to calculate small diffusion terms which are used to reduce instabilities within the non linear continuity equation.
Inputs

All required input for DIFFUS is input to the routine by ADVECT through the formal parameter list. Major input variables are defined under ICEMDL. Any important functions performed by these variables in DIFFUS are detailed below.

Processing

Subroutine DIFFUS applies a simple smoothing operator to obtain a small diffusion term for the respective field being analyzed. This term is applied to the results of the computations of the dynamical portions of the continuity equations.

Output

The diffusion terms are stored in the third level of the array, HEFF.

Interfaces

No major subroutines are accessed by DIFFUS.

Tables and Items

Subroutine DIFFUS introduces no previously defined variables.
2.4.3.9  

**Subroutine HEAT**

Subroutine HEAT is the driving routine for the heat budget code. This budget solves for the thermodynamic growth rate of the thick and thin ice.

**Input**

Subroutine HEAT receives all input variables from ICEMDL, through the formal parameter list. Subroutine HEAT also accesses the CRANDIO data base which contains the FNOC environmental data. The following data records are accessed:

1) air temperature;
2) surface pressure;
3) surface vapor pressure;
4) incoming solar radiation;
5) sensible heat flux;
6) sensible plus evaporative heat flux.

The data is read from the CRANDIC file through the use of subroutine INITIAL of the input module.

**Processing**

The primary function of subroutine HEAT is to set up all necessary variables for the heat budget calculations, performed in subroutine BUDGET.

The wind data is calculated in the first main loop of HEAT. This variable is calculated from GAIRX and Gairy which contain the u and v components respectively.
Subroutine INITIAL is called to read the air temperature data at the surface from the CRANDIO data base. Subroutine AVG is called to compute the grid cell average of the air temperature data. Subroutine AVG is used for all thermodynamic variables, within HEAT to define them within the staggered grid. The temperature data is finally converted from centigrade to Kelvin.

Subroutine INITIAL and AVG are again used to define the atmospheric surface pressure and vapor pressure. These variables are used to calculate the moisture at the surface which is stored within QA.

Subroutine INITIAL is finally used to retrieve the shortwave radiation, sensible heat flux and sensible plus evaporative heat flux. The variables are converted from the CGS system to the MKS system and used to calculate the incoming longwave radiation. At the end of the processing the following variables have been set up for use in BUDGET:

   i)   TAIR - air temperature;
   ii)  QA   - surface moisture;
   iii) FSH  - incoming shortwave radiation;
   iv)  FLC  - incoming longwave radiation.

The final preparation before BUDGET is called is the definition of the mixed layer depth (HMIX) plus the temperature of the mixed layer and temperature of the ice, TMIX and TICE respectively.

The variable, KOPEN, is used as a flag to BUDGET to determine whether thin ice or thick ice growth rates are evaluated. Subroutine BUDGET is then called to calculate the growth rate of each category. The
total growth rate is then calculated and stored in FHEFF.

Output

The environmental parameters listed under the Tables and Items section are output from HEAT.

Interfaces

Subroutine HEAT interfaces with two main subroutines, INITIAL and BUDGET. Subroutine INITIAL is used to retrieve data from the CRANDIC data base and BUDGET is used to calculate the thin and thick ice growth. The parameter list for INITIAL is defined as follows:

i) parameter 1 - number, indicating which catalog name is to be used in INITIAL;

ii) parameter 2 - array used to store the environmental data returned from INITIAL;

iii) parameter 3 - i grid points of the grid;

iv) parameter 4 - j grid points of the grid;

v) parameter 5 - DTG required;

vi) parameter 6 - number of points in the grid;

vii) parameter 7 - tau value.

45
The parameter list of BUDGET is defined as follows:

i) parameter 1 - ice thickness;
ii) parameter 2 - growth rate returned to HEAT;
iii) parameter 3 - flag used to determine whether thin or thick ice growth is calculated;
iv) parameter 4 - grid size;
v) parameter 5 - grid size;
vi) parameter 6 - wind data;
vii) parameter 7 - ice temperature;
viii) parameter 8 - mixed layer temperature;
viii) parameter 9 - air temperature;
ix) parameter 10 - surface moisture;
x) parameter 11 - incoming longwave.

Tables and Items

The following new major variables are defined in HEAT:

i) TICE - ice temperature;
ii) TMIX - mixed layer temperature;
iii) TAIR - air temperature;
iv) QA - surface moisture;
v) FLO - incoming longwave radiation;
v) PS - surface pressure and sensible heat flux;
vii) CS - surface vapor pressure and sensible plus evaporative heat flux;
viii) UG - wind values.
The following common block is defined:

/RAD/ - contains incoming shortwave radiation.

2.4.3.10 Subroutine BUDGET

Subroutine BUDGET is used to calculate the thin and thick ice growth rates by using a simple heat budget.

Input

Subroutine BUDGET receives all major input variables from HEAT. Most of these are passed through the formal parameter list. One variable, FSH, is passed in common block /RAD/.

Processing

The initial code of BUDGET is dedicated to defining all necessary constants used in the heat budget calculations.

A branch is made, depending upon the value of KOPEN. If KOPEN is less than zero, processing continues to compute the thin ice growth rate. If KOPEN is greater than zero processing jumps to compute the thick ice growth rate.

Continuing with the the ice growth, the main heat budget equation components, and growth rate derived within loop 101. Subroutine BUDGET ends processing here for this branch.
When the thick ice growth rate is computed, there are two main components of the heat budget calculation. The ice temperature, TICE, is solved for iteratively (Newton-Raplsion technique) and used in the head budget equation. When the ice temperature values are relatively stable between iterations, processing finishes returning the thick ice growth rate.

**Output**

The output of BUDGET is contained in the following variables:

i) FHEFF - thick ice growth rate;
ii) FO - thin ice growth rate.

**Interfaces**

Subroutine BUDGET calls no other subroutines.

**Tables and Items**

The variable, ALB, defining the surface albedo, is the only major variable not previously defined and used within BUDGET.

2.4.3.11 **Subroutine GROWTH**

Subroutine GROWTH is used to calculate the change in ice thickness and concentration due to growth/decay.
Input

All required input variables to GROWTH are passed from ICEMDL through the formal parameter list. No variables, not previously defined, are input to GROWTH.

Processing

The processing of GROWTH is contained in one major loop. The amount of ice grown and melted during the time step. The changes, reflected by the growth/decay rates, are added to the ice thickness, HEFF, and ice concentration, AREA, variables. The ice concentration value is then checked to contain it within the limits specified. Ice concentrations are not allowed to be larger than 1.0.

Output

The variables HEFF, AREA and GAREA contain output variables of GROWTH.

Interfaces

No other subroutines are called by GROWTH.

Tables and Items

No new variables are introduced by GROWTH.
2.4.3.12 Subroutine ADJUST

Subroutine ADJUST is used to set up the thickness and concentration at the outflow cells.

Input

All input variables, to ADJUST are passed from ICEMDL through the parameter list. No variables not previously defined are used.

Processing

Subroutine ADJUST is called at the end of each time step. ADJUST uses routine MEAN to calculate the ice in the open cells by taking an average of all grid cells adjacent to the open boundary. The ice thickness and concentration arrays are both modified in this manner.

Outputs

The arrays HEFF and AREA (ice thickness and ice concentration) are respectively modified.

Interfaces

Subroutine ADJUST calls subroutine MEAN to calculate the ice in open grid cells.

Tables and Items

No major new variables are defined.
2.4.3.13  

**Subroutine MEAN**

Subroutine MEAN is used to calculate the amount of ice in open cells.

**Input**

All data input to MEAN is passed by ADJUST through the parameter list. No new variables are introduced as input.

**Processing**

Subroutine MEAN calculates the amount of ice in a grid cell as the mean of ice in the adjacent cells. Array CUT, which is the boundary mask specifying the outflow cells, is used to control the calculations to output the mean ice held within the outflow cells.

**Output**

The variable, HMEAN, is output to ADJUST, containing the mean ice thickness and concentration on the open cells.

**Interfaces**

No other subroutines are called by MEAN.

**Tables and Items**

No new major variables are introduced by MEAN.
2.4.4 Output Module

2.4.4.1 Subroutine DIVERG

Subroutine DIVERG is used to output the divergence values to the CRANDIO output file.

Input

All required input variables are supplied to DIVERG from ICEMDL through the formal parameter list. No new variables are input.

Processing

Subroutine DIVERG operates like all output module subroutines. Major processing functions are performed for the sole purpose of setting up the data and appropriate record names for the CRANDIC data base.

The data is placed into the common block array DIVRG. The record label is set up using a data statement which specifies the catalog name, tau value and record length.

If an error occurs on the CWRITER function an error message is written, stating this fact and the program will terminate.

Output

Subroutine DIVERG places the divergence values on the CRANDIO output file.
Interfaces

Subroutine DIVERG interfaces with the standard FNOC CRANDIC software. DIVERG is called on every fourth time step by ICEMDL.

Tables and Items

The following common block is defined;

/DV/ - MDV - contains CRANDIC ID block;
DIVRG - contains data values;
FILLV - fills the small page (requirement of CRANDIC).

2.4.4.2 Subroutine UVPLCT

Subroutine UVPLCT is used to output the ice drift forecasts.

Input

All input to UVPLCT is provided by ICEMDL, through the parameter list. No new variables are introduced.

Processing

The first main function of UVPLCT is to convert the U, V ice drift components to ice drift direction and speed. This is accomplished through the FNOC routine UVDDFF. Ice drift speed is converted from m/sec to knots.
The final function is setting up of the CRANDIC record labels for CRANDIC. The record labels are set up using a data statement containing the catalog names, tau values and record lengths.

If an error occurs on the CWRITER function, a message is written and the program terminates.

**Output**

The ice drift direction and speed is output to the final CRANDIO file.

**Interfaces**

UVPLCT interfaces with the FNCC CRANDIC software. UVPLCT is called on every fourth time step by ICEMDL.

**Tables and Items**

No new major variables are introduced by UVPLCT.

2.4.4.3 **Subroutine BUOYD**

Subroutine BUOYD is used to calculate the drift of simulated buoys placed in the model.

**Input**

All input required for BUOYD is supplied by ICEMDL through the parameter list. No new variables are introduced by BUOYD.
The buoy drift positions are input through common block /BUOY/.

**Processing**

The initial processing of BUCYD checks the position of each buoy. This is done to determine if a buoy has moved out of the grid. Once a buoy reaches a boundary it is no longer tracked. The routine branches around the remainder of the processing and goes on to another buoy when one has moved to the boundary.

Subroutine INTRP, a FNOC utility routine, is used to interpolate the U, V ice drift components to the buoy position recorded during the previous time step. The u, v values are then used to calculate how far the buoy shall have moved during the current time step. This distance is then determined in terms of grid units on both the model grid and the FNOC hemispheric grid.

The final position of each buoy, in terms of both grid, is printed in a table.

**Output**

The positions of the buoys, calculated in common block /BUOY/ are output from BUCYD a printed table, outlining the position of each buoy is provided.

**Interface**

BUCYD utilizes the FNOC utility routine INTRP to provide u, v components of ice drift at buoy positions.
Tables and Interfaces

No new variables are introduced within BUOYD.

2.4.4.4 Subroutine HAPLOT

Subroutine HAPLOT is used to output the ice thickness and concentration values to the CRANDIC file.

Input

All required input to HAPLOT is provided by ICEMDL through the formal parameter list. No new input variables are defined.

Processing

Processing within HAPLOT performs two functions. The first is a unit conversion while the second creates the necessary data for CRANDIC. Ice thickness is output first. The thickness values are converted from meters to cm. The CRANDIC record labels and data are set up into their respective positions and CWRITE is used to write the data.

The ice concentration values are handled in the same manner. However no unit conversion is performed.

If an error results in the processing of any CWRITE an appropriate message is written and the program shall terminate processing.
Output

The ice thickness and compactness values are output to the CRANDIO file.

Interfaces

HAPLOT interfaces with the CRANDIO software.

Tables and Items

No new major variables are introduced.

2.4.4.5 Subroutine GROWDEC

Subroutine GROWDEC is used to output the open water growth and ice growth forecasts.

Input

All required input is passed to GROWDEC by ICEMDL.

Processing

The processing of GROWDEC proceeds exactly as other output module subroutines. The CRANDIO labels are defined and CWRITER is used to output the data.

Output

Forecasts of open water and total ice growth are written to the CRANDIO file.
Interfaces

GROWDEC interfaces with the CRANDIO software.

Tables and Items

No new variables are defined.

2.4.4.6 Subroutine PRNT

Subroutine PRNT is a small subroutine which is used to print model arrays. The processing of PRNT depends entirely upon the input data specifications.

Input

The formal parameters are defined as follows:

i) ARRAY - array to be printed;
ii) I,J,K - dimensions of ARRAY;
iii) MI,MZ - columns of ARRAY which are printed;
iv) N - number of rows of ARRAY to be printed.

Output

Printed output of a data array is provided.
Section 3.0  Environment

3.1  Equipment Environment

FNOC operates a multiprogramming/multiprogramming/mainframe computer system consisting of three CDC 6500's, with 131K each of central memory (CM), 9 CDC CYBER 170/175 with 196K CM and 1 million words of CDC 7030 extended core storage (ECS), related auxiliary equipment (7 and 9 track tape units, disk storage, etc.), a CYBER 203 with 2 million words of CM and its front-end processor CYBER 170/720, and a VARIAN plotter and its plotting software. The Sea Ice model is designed to run on the CYBER 203 computer, and its output is plotted on the VARIAN plotter.

3.2  Support Software

FNOC operates under the NOS/BE operating system. This system contains many local enhancements/modifications to facilitate ease of operation. Most of the enhancements are documented in either the FNOC subroutine/utility file or the FNWC Computer User Guide Edition 2. The Sea Ice Model was converted to CDC CYBER 200 FORTRAN Language, version 1.5 and utilizes various routines available on FNWCLIB.

3.3  Data Base

A number of data bases are maintained and needed by the sea ice model.
3.3.1 General Characteristics

Three separate input data bases are required for the sea ice model to properly execute. Two of these are created by the user. The third data base consists of the FNOC environmental data. Currently this data is not operationally available on the CY203, therefore, this data base is also set up by the user, from data available on other machines.

i) TAPE7

TAPE7 is the mnemonic for the main input file. This file is set up by a user and remains permanent for the desired configuration of the model runs.

The file is utilized by the input module and describes the execution configuration for the current run (e.g., grid size, grid location, grid configuration).

ii) TAPE8

TAPE8 is a more dynamic data file than TAPE7. TAPE8 contains the DTG's of the days during the current run. Therefore, this file will change for each run.

The file, TAPE8, needs to be defined by the user of the sea ice model.

iii) FNOC Data

FNOC data is kept on a CRANDIO file which, at the time of this writing, must be set up by the user. These data will change for a specific run. The data is obtained from another machine which has access to the FNOC master data base (MASFNWC).
iv) Output Data Base

A CRANDIC output data base is maintained by the sea ice model. This data base is filled with forecast variables computed by the sea ice model. Variables are written in specified time steps, defined in routine ICEMDL.

3.3.2 Organization and Detailed Description

i) TAPE7

The file labeled TAPE7 is a binary file accessed by a formatted READ. The following information is contained, with formats:

i) grid size specifications - 215;
ii) FNOC i grid points for MESH - 2F10.2;
iii) FNOC j grid points for MESH - 2F10.2;
iv) number of rows/columns in model grid - 15
v) uv boundary mask - 27F2.0, 27 rows
vi) thickness boundary mask - 28F2.0, 28 rows;
vii) outflow boundary mask - 28F2.0, 28 rows;
viii) ocean current direction - 25F3.0, 25 rows.

ii) TAPE8

The file labeled, TAPE8, is a binary
file accessed with a formatted READ. The following information is contained, with formats:

i) number of time steps to be run - I5;

ii) DTG, one row for each time step
    3(A8,2X).

The DTG rows are defined as described under ICEMDL.

iii) FNOC Input Data

The input FNOC data base is a CRANDIC data file on the CY203. CRANDIC is the operational data file format, specified by FNOC, on the CY203. Detailed characteristics of the CRANDIO files can be found in the CRANDIO software documentation distributed by FNOC.

The data on the CRANDIC file is created by transferring ZRANDIC data, from other FNCC machines to the CY203 with proper ZRANDIO to CRANDIO conversion software.

The following records are contained on the CRANDIO input file:

<table>
<thead>
<tr>
<th>Record</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>A20</td>
<td>u-wind component</td>
</tr>
<tr>
<td>A21</td>
<td>v wind component</td>
</tr>
<tr>
<td>A10</td>
<td>air temperature</td>
</tr>
<tr>
<td>A01</td>
<td>surface pressure</td>
</tr>
<tr>
<td>A12</td>
<td>surface vapor pressure</td>
</tr>
<tr>
<td>A11</td>
<td>short wave radiation incoming</td>
</tr>
<tr>
<td>A18</td>
<td>total heat flux</td>
</tr>
<tr>
<td>A16</td>
<td>sensible plus evaporative heat flux.</td>
</tr>
</tbody>
</table>
iv) CRANDIO Output File

The CRANDIO output file is created by the sea ice model. CRANDIO records are written consisting of forecast variables on certain time steps. The following records are written every four time steps:

<table>
<thead>
<tr>
<th>Record</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>i)</td>
<td>DIV ice divergence/convergence</td>
</tr>
<tr>
<td>ii)</td>
<td>FFF ice drift speed</td>
</tr>
<tr>
<td>iii)</td>
<td>DDD ice drift direction</td>
</tr>
<tr>
<td>iv)</td>
<td>THK ice thickness</td>
</tr>
<tr>
<td>v)</td>
<td>CCM ice concentration</td>
</tr>
<tr>
<td>vi)</td>
<td>PRS ice pressure</td>
</tr>
<tr>
<td>vii)</td>
<td>GAR open water growth</td>
</tr>
<tr>
<td>viii)</td>
<td>HDF ice growth</td>
</tr>
</tbody>
</table>
Section 4.0  Program Maintenance Procedures

4.1  Conventions

The Sea Ice Model system adheres to structured design and programming principles. Flowcharting and naming conventions adhere to the standard identified below.

b. CDC Programming Standards, CDC-STD 1.80.000, December, 1971.

4.2  Verification Procedures

The methods of verifying the sea ice model output through display of the output file on plotting display or printed output. Plotting of ice drift, ice growth/decay rate, ice thickness is a very efficient method used to check output of the model.

4.3  Error Conditions

This section describes the error conditions determined by the Sea Ice Model.

ICEMDL

Message - "OPEN ERROR"
Reason - error in opening the CRANDIC output file
64
RELAX

Message - "No convergence after 1300 iterations"
Reason - UICE and VICE are not convergent after 1300 iterations.

UVPLCT

Message - "STATUS is (value) ON WRITE OF filename"
Reason - ISTAT is not equal to 0.
Result - Output of UVPLCT is not written to CRANDIC file, program stops.

HAPLCT

Message - "STATUS IS (value) ON WRITE OF filename"
Reason - ISTAT is not going to 0.
Result - Output of HAPLCT is not written to CRANDIC file, program stops.

DIVERG

Message - "CWRITE STATUS IS (Value) ON WRITE OF filename"
Reason - ISTAT is not equal to 0, the CRANDIO output file is not opened.
Result - Output of DIVERG is not written to CRANDIO file, program stops.
65
INITIAL

Message - "CHECKNC no data initial"
Reason - No required data field for input
Result - Stop the program.

4.4 Special Maintenance Procedures

There are no special maintenance procedures for the Sea Ice Model program.

4.5 Special Maintenance Programs

There are no special maintenance programs for the Sea Ice Model.

4.6 Listings

Listings of the Sea Ice Model program and subroutines are to be found in Appendix C.
APPENDIX A

ICEMDL Flowcharts
ICEMDL

Read control variables

call MESH

call BNDRY

call OCEAN

call INITIAL

Read Restart data

Print Restart data
Start of new processing logs

Call FORM RELAX

Call FORM RELAX

Compute drift tolerance

Print drift forecast

Call DIVERG UVPLOT

Call BUOYD

Call ADJECT

Call GROWTH
3

Compute diagnostics

Print output

Call HAPLOT GROWDEC

Call INITIAL

new data

new time step

STOP
DIFFUS

set Constants

calculate diffusion term

end
XSUM

Sum vector

end
MEAN

Calculate grid cell means

end
OCEAN

Read current directions

Formulate data in grid

Call DDFFUV

end
FELLIP

Compute finite differences

end

Finite differences for RELAX as a function of current U.V.

FELLD1

Compute finite differences

end

Finite differences for main interaction process in RELAX
CALCULATE MATRIX COEFFICIENTS

CALL FELLIP

CALL FELLDI

CALCULATE VARIABLES OF PREVIOUS U.V.

CALCULATE NEW U.V.

SMALL ERROR

END
GROWTH

Calculate total growth

Add growth decay to thickness concentration

Truncate AREA

end
Heat

Calculate wind term

Call INITIAL surface pressure

Call INITIAL vapor pressure

Call INITIAL air temperature

Call INITIAL shortwave radiation

Call INITIAL total heat flux

Call INITIAL evaporative heat flux
Perform unit conversion

Calculate ice, mixed layer temp.

Call BUDGET open water

Call BUDGET ice

end
FORM

Calculate ice mass, coriolis force

Calculate drag terms

Calculate external forces

Calculate ice strength

Call PLAST

Finalize external forces

end
PLAST

Calculate strain rates

Calculate viscosities

Calculate ice stress

Calculate divergence

end
ADJUST

Call MEAN

Set outflow thickness

Call MEAN

Set outflow concentration

end
MESH

Read defining points

Fill in grid mesh

Calculate grid size map factor

Set buoy points

end
DIVERG
Set up data array
Set up CRANDIO label
Write data

Print error message
N

good write

STOP

end
Set up CRANDIO record label

Write error message

record present

N

STOP

Read data

Call INTRP

Fill grid with data

end
HAPLOT

Convert thickness from m to cm

Set up thickness label

Write thickness

Set up concentration label

Write concentration

Print error message

N

good writes

STOP

y

end
UV PLOT

Convert U.V. to D.F.

Convert m/ sec to knots

Set up CRANLIO record label

Write record

Print error message

good write

STOP

end
BUDYD

y
Buoy
off grid

N

Call
INTRP
Get UV
at buoy

Calculate
new buoy
position

N
All
buoys
moved

y

end
Appendix B

Grid Structure
The model grid is set up in a staggered manner. Momentum variables are defined at the grid points while thermodynamic variables are defined for grid cells.

Environmental variables accessed from the FNOC data base are computed to be valid at the grid cell locations. Subventive AVG is used to compute the grid cell averages for the thermodynamic variables.

The following example illustrates the definition of grid parameters for a small grid.

Say, we define NX,NY to be 7 (dimension of momentum variable grid). Figure B1 illustrates that NX1,NY1 (thermodynamic grid) will be 8 while the value of NUMBER will be 9.
Figure B1  Grid Configuration

NUMBER = 9

NX1, NY1 = 8

NX, NY = 7
APPENDIX C

ICE MDL Listings
1.5.1 CYCLE FT/1985 BUILT 03/19/85 29 27 SOURC LISTING

DATA COUNT, T1, T2, N1, N2, F1, F2, X1, X2, Y1, Y2, Z1, Z2, W1, W2, V1, V2, U1, U2

CALL SECF (TH3

PRINT 10000, counts, X1, Y1, Z1, W1

1000 FORMAT('T1 = ', T1, ' T2 = ', T2, ' N1 = ', N1, ' N2 = ', N2, ' F1 = ', F1, ' F2 = ', F2, ' X1 = ', X1, ' Y1 = ', Y1, ' Z1 = ', Z1, ' W1 = ', W1)

C DEFINE THE BASIC PARAMETERS

CALL SECF

PRINT 1000, T1, T2, N1, N2, F1, F2, X1, Y1, Z1, W1

10000 FORMAT(' T1 = ', T1, ' T2 = ', T2, ' N1 = ', N1, ' N2 = ', N2, ' F1 = ', F1, ' F2 = ', F2, ' X1 = ', X1, ' Y1 = ', Y1, ' Z1 = ', Z1, ' W1 = ', W1)

C OBTAIN THE INITIAL VALUES

LISTR = 0

CALL INITIAL (X1, Y1, Z1, W1, T1, T2, N1, N2, F1, F2, X1, Y1, Z1, W1)

CALL INITIAL (X1, Y1, Z1, W1, T1, T2, N1, N2, F1, F2, X1, Y1, Z1, W1)
THAN 1.5 .1 CYCLE FULLEST WALT F L//M27 SOURCE LISTING

00047  CE = CO + 2
00054  DO 12 T = 1.000
00061  DATA(I) = DATA(I) / 100.0
00064  DATA(I) = DATA(I) / 100.0
00067  CONTINUE

AND INITIAL SYSTEM

FIRST GIVE AT INITIAL HEAT AND AREA

00073  DO 12 I = 1,27
00076  DATA(I) = 1.000
00079  VICE(I) = 0.0
00082  CONTINUE
00087  DO 12 I = 1,27
00090  DATA(I) = 1.000
00097  VICE(I) = 0.0
00100  CONTINUE

Call: RCLY(CYCLE,TIME,HALF,1.0)

START WITH AN INITIAL VELOCITY FIELD OF ZERO

Call: PTLX(CYCLE,TIME,HALF,1.0)

SET RCLY TO 0 AND DEFINE THE VISCOITIES

00104  DO 12 I = 1,27
00107  DATA(I) = 0.0
00110  A = 0.0
00113  ETA(I) = 1.00
00116  ETA(I) = ETA(I) * (1.00 + 11)
00119  ETA(I) = ETA(I) / 4.0
00122  CONTINUE

Call: DELAY(100.0,0.0,1.0,13.0)

DATA(100) = DELAY(100.0,0.0,1.0,13.0)
SUBROUTINE ADJUST

THICKNESSES AND CONTACTS VALUES AT THE OUTFLOW CELLS ARE
ESTIMATED USING ADJUSTIVE METHOD.
ALL ICE FLOWING INTO THE GROUND THROUGH THE OPEN CELLS IS
ACCOUNTED FOR.
THESE CONTAIN THE AMOUNT OF ICE IN THE GROUND CELLS....
THIS IS USED IN THE CONVECTION CALCULATIONS.

CALL ADJUST(X(NHEX),X(NHEX+1),X(NHEX+2),X(NHEX+3))

START THE STANDARD PRECURSOR CORRECTIVE ITERATION SCHEME

CALL X(1,1,1)
PRINT 1222
10 P=RT(TP,1,T0,1,INITIATION TIME T0,1,1)
10 CONTINUE
CALL X(1,1,1)
CALL X(1,1,1)+X(1,1,1)
X(1,1,1)=X(1,1,1)+X(1,1,1)
10 CONTINUE

FIND IN THE PRECURSOR

DO 10 I=1,NH
DO 10 J=1,NK
ICF(I,J)=X(I,J)
10 CONTINUE
ICF(I,J)=ICF(I,J)
ICF(I,J)=ICF(I,J)
10 CONTINUE
CALL FLOY(1,ICF,ICF,ETAEZETA,X100A100B100C100D100E100F100G100H100I100J100K100L100M100N100O100P100Q100R100S100T100U100V100W100X100Y100Z1001002)
CALL RELAX(1,ICF,ICF,ETAEZETA,X100A100B100C100D100E100F100G100H100I100J100K100L100M100N100O100P100Q100R100S100T100U100V100W100X100Y100Z1001002)

CAN BE A SIMPLER TIME STEP

10 CONTINUE
THETA=1.0
DELTA=DELTA/D
10 CONTINUE
CALL FLOY(1,ICF,ICF,ETAEZETA,X100A100B100C100D100E100F100G100H100I100J100K100L100M100N100O100P100Q100R100S100T100U100V100W100X100Y100Z1001002)

SET U(N)=X(N) SAVE FOR U

DO 11 I=1,NH
DO 11 J=1,NK
11 CONTINUE
ICF(I,J)=ICF(I,J)
00140 \text{CONTINUE} \text{ CONTAINS THE NET OPEN WATER GROWTH}

00180 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00240 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00300 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00360 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00420 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00480 \text{CONTINUE CONTAINS THE NET OPEN WATER GROWTH}

00587 \text{CONTINUE} \\
00140 \text{CONTINUE}
CALL ADJUST TO ESTIMATE THE ICE IN THE UNFROZEN CELLS

CALL ADJUST(NETHEAT, NETHEAT*0.0, NETHEAT*0.1, NETHEAT*0.1, NETHEAT*0.1)

DETERMINE IF THE ITERATION PROCESS CONTINUES

CALL STATUS(T2)

T = T2 - T1

IF T < 0.0 THEN

ELSE IF T > 0.0 THEN

WHERE THE INDIVIDUAL IS PROJECTED

LISTED = (LISTED + 1)

IF LISTED < L (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) THEN

LISTED = 0

LISTED = (LISTED - 1) / TOTS(1) / TOTS(2) / TOTS(3)

SAVE TOTS(2), TOTS(3) FOR TOTS(1)

CALL PRINTOUT (TOTS(1), TOTS(2), TOTS(3), TOTS(4), TOTS(5), TOTS(6), TOTS(7), TOTS(8), TOTS(9))

T = 1.0

T = T2

T = (T3) / TOTS(1) + TOTS(2) / TOTS(2) + TOTS(3) / TOTS(3)

CONTINUE

VALUES OF...

GOTO 100

CONTINUE

END OF THE 514 TOTALS END OF INPUT

"FILE(1,2,3) SPEC.I, SPEC.I, SPEC.I, SPEC.I, SPEC.I"

FILE(1,2,3) SPEC.I, SPEC.I, SPEC.I, SPEC.I, SPEC.I"

CALL PRINTOUT

TOTS = TOTS(TOTS)

CALL STATUS(TOTS)

TOTS = 2.0

CALL STATUS(TOTS)

STOP 0.0 OF THE MODEL
1045 IF((LL.ROI.1) G.T. 100) 
1050 LL = LL - 2 
1055 LF = LT = FT = 2 
1060 SF = 0 
1065 IF = 0 
1070 CONTINUE 
1075 
1080 IF (FT.ROI.1) G.T. 100) 
1085 LF = LT = FT = 2 
1090 SF = 0 
1095 IF = 0 
1100 CONTINUE 
1105 
1110 IF (FT.ROI.1) G.T. 100) 
1115 LF = LT = FT = 2 
1120 SF = 0 
1125 IF = 0 
1130 CONTINUE 
1135 
1140 IF (FT.ROI.1) G.T. 100) 
1145 LF = LT = FT = 2 
1150 SF = 0 
1155 IF = 0 
1160 CONTINUE 
1165 
1170 IF (FT.ROI.1) G.T. 100) 
1175 LF = LT = FT = 2 
1180 SF = 0 
1185 IF = 0 
1190 CONTINUE 
1195 
1200 IF (FT.ROI.1) G.T. 100) 
1205 LF = LT = FT = 2 
1210 SF = 0 
1215 IF = 0 
1220 CONTINUE 
1225 
1230 IF (FT.ROI.1) G.T. 100) 
1235 LF = LT = FT = 2 
1240 SF = 0 
1245 IF = 0 
1250 CONTINUE 
1255 
1260 IF (FT.ROI.1) G.T. 100) 
1265 LF = LT = FT = 2 
1270 SF = 0 
1275 IF = 0 
1280 CONTINUE 
1285 
1290 IF (FT.ROI.1) G.T. 100) 
1295 LF = LT = FT = 2 
1300 SF = 0 
1305 IF = 0 
1310 CONTINUE 
1315 
1320 IF (FT.ROI.1) G.T. 100) 
1325 LF = LT = FT = 2 
1330 SF = 0 
1335 IF = 0 
1340 CONTINUE 
1345
TRAN 1.5.1 CYCLE RTW1234 BUILT 04/05/84 29 27 SOURCE LISTING

00034  4HFF(1,1,2) 4HFF(1,2,1)
00037  CONTINUE
00038  GO TO 142
00042  CONTINUE

00044  GO TO 142
00045  CONTINUE
00047  K4=3
00047  CALL RTM32
00050  CONTINUE

00052  CALL RTM32(4HFF(1,1,2),4HFF(1,2,1))
00058  G1) GO TO 142
00059  IF (G1.GT.24) GO TO 142
00060  CONTINUE

00062  CALL RTM32(4HFF(1,1,2),4HFF(1,2,1))
00063  CONTINUE
00065  CALL RTM32(4HFF(1,1,2),4HFF(1,2,1))
00067  CONTINUE
00069  CALL RTM32(4HFF(1,1,2),4HFF(1,2,1))
00071  CONTINUE
00073  CALL RTM32(4HFF(1,1,2),4HFF(1,2,1))
00075  CONTINUE
00077  CONTINUE

00078  CONTINUE
<table>
<thead>
<tr>
<th>1.51</th>
<th>CYCLE ( T )</th>
<th>( L )</th>
<th>( R )</th>
<th>( B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**SOURCE LISTING:**

```
output

def f(x):
    return x**2

result = f(3)
print(result)
```
1.3.1 CYCLE 01/12/68 12/07/68 12/27 12/27

100001

100002  IF (X <= 3) THEN
100003   I = I + 1
100004   J = J + 1
100005  END IF
100006  S1 = S1 + EFF(I,J)
100007  CONTINUE
100008  RETURN
100009  END
1. (5.1) CYCLE E11111 E11111 E111111 E11111 E111111 E111111

2. (7.5.1) CYCLE E111111 E111111 E111111 E111111 E1111111 E1111111

3. (7.5.1) CYCLE E1111111 E1111111 E1111111 E1111111 E1111111 E1111111

4. (7.5.1) CYCLE E11111111 E11111111 E11111111 E11111111 E11111111 E11111111

5. (7.5.1) CYCLE E111111111 E111111111 E111111111 E111111111 E111111111 E111111111
1.5.1 CYCLE EXECUTE BUILT IN DATA 29 27 SOURCE LISTING

1001 SUBROUTINE DFLAX (TICE, DFETA, TETA, TETA1, TETA2)
1003 7DFETA(RKI) = DFETA(RKI) - DFETA(RKI) * TETA + TETA1 + TETA2
1005 7FLAX(RKI) = FLAX(RKI) - FLAX(RKI) * TETA + TETA1 + TETA2
1007 7FETA(RKI) = FETA(RKI) - FETA(RKI) * TETA + TETA1 + TETA2
1009 7FFT(RKI) = FFT(RKI) - FFT(RKI) * TETA + TETA1 + TETA2
1011 7FTZ(RKI) = FTZ(RKI) - FTZ(RKI) * TETA + TETA1 + TETA2
1013 7FLX(RKI) = FLX(RKI) - FLX(RKI) * TETA + TETA1 + TETA2
1015 7CT(RKI) = CT(RKI) - CT(RKI) * TETA + TETA1 + TETA2
1017 7CTI(RKI) = CTI(RKI) - CTI(RKI) * TETA + TETA1 + TETA2
1019 7FZ(RKI) = FZ(RKI) - FZ(RKI) * TETA + TETA1 + TETA2
1021 CONTINUE

NOW SET THE COEFFICIENTS OF THE POLYNOMIAL COMPONENTS

1023 DO 10 J = 1, J
1025 DO 10 J = 1, J
1027 C = DFETA(J) * TETA + TETA1 + TETA2
1029 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1031 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1033 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1035 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1037 CONTINUE

NOW CALCULATE ALL FUNCTIONS OF PREVIOUSLY AN V VALUES

1039 TTFETA(J) = TTFETA(J) + (TETA - TETA)
1041 DO 10 J = 1, J
1043 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1045 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1047 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1049 CALL FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1051 CONTINUE

1053 FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1055 FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
1057 FFLX(TICE, TICE, DFETA, TETA, TETA1, TETA2)
ITP.N: I C Y C L E E T H I S A R U T L T 0 9/13/71 17:27 SOURCE LISTING

000040      DO 106 J = 1,44
000047      DO 106 J = 1,44
000053      VERC = VICE(I,J,I) = VICE(I,J,I)
000059      VERC = VICE(I,J,I) = VICE(I,J,I)
000065      S1 = SAVVE(I, ARG(VECC), S1 )
000071      S2 = SAVVE(I, ARG(VECC), S2 )
000077      CONTINUE
000083      S1 = SAVVE(I, S1+S2 )
000089      IF(S1.LT.0.000) GO TO 200
000095      T = T + 1
000101      201 CONTINUE
000107      PRINT 11
000113      11 FORMAT(3X,'ITERATIONS AFTER LAST ITERATION')
000119      C  CONTINUE
000125      C  PRINT 1+COUNT
000131      PRINT 1+5
000137      PRINT 1+COUNT
000143      12 FORMAT(3X,'ITERATION IN WHICH THE INSIDE OF THE 3X1.2 A')
000149      13 FORMAT(3X,'NUMBER OF ITERATIONS ARE 10.1P0)
000155      CALL SUB2(7)
000161      RELAX = RELAX + (T2 - T1)
000167      RETURN
000173      END
1.5.1 CYCLE: 

\[
\text{MOVE}(\text{S}, \text{D}) \text{, IF } (\text{D} \neq \text{S})
\]

\[
\text{D} \leftarrow \text{D} + \text{S}
\]

\[
\text{DO } \text{D} = 1 \text{ TO } \text{V}
\]

\[
\text{C} = (\text{C} \times \text{D}) + (\text{X} \times \text{M})
\]

\[
\text{C} \leftarrow (\text{C} + \text{X})
\]

\[
\text{C} = (\text{C} \times \text{D}) + (\text{X} \times \text{M})
\]

\[
\text{X} \leftarrow \text{X} + \text{D}
\]

\[
\text{CONTINUE}
\]

\[
\text{FINISH}
\]
154 CYCLE FLYING BUILT IN/03/21 22 21 SOURCE LISTING 4001

155 T=V((X,J)) = $21$°
156 TICE(T,J) = 073°
157 J

10

158 10 V

159 CALL 31007 (HEESE+F40+WERNER(1+J)) J3;TICE(T,J)*T1+T2+L0)
160 CALL 1004 6 (31007 J)
161 J1

162 CALL 31007 (HEESE+F40+WERNER(1+J)) J3;TICE(T,J)*T1+T2+L0)
163 J1

164 1 1047 T = $144^1$
165 1047 T = $144^1$
166 1047 T = $144^1$

167 1 1047 T = $144^1$

168 1047 T = $144^1$

169 1047 T = $144^1$

169 J1

1047 T

1047 T

1047 T
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTINUE</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SET</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PRINT</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>READ</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

Note: The code appears to be a listing for a programming language, possibly COBOL, given the syntax and structure.
1.3.1 CYCLE ENTRIES 29 JULY 79/03/11 22 27 SOURCE LISTING

CALL AREA(I+1) = AREA(I+1) + (HEFF(I+1) - M(I+1)) * N(I+1)

100 CONTINUE

RETURN
The solutions were used to calculate the first order of the matrix.

where the off-diagonal elements are calculated at each point and the matrix factors are used to calculate the mesh size at the node.
THAN 1.5,1 CYCLE FTVIS 1, RLI 7#207/41 22 27 SOURCE LISTING

00010 SET 640 INCREMENTS

00020 SET UP 1 POINTS

00021 DELTA = (J1 - J0) / FLOAT(V-1)
00022 J1 = J0
00023 J0 = J0 + 1
00024 J1 = J1 + 1
00025 (J1,J1) = J1
00026 CONTINUE
00027 J1 = J1 - DELTA
00028 CONTINUE

00029 DELTA = (J1) - J0) / FLOAT(V-1)
00030 J1 = J0
00031 J0 = J0 + 1
00032 J1 = J1 + 1
00033 (J1,J1) = J1
00034 CONTINUE
00035 J1 = J1 + DELTA
00036 CONTINUE

00037 DELTA = (DELTA + DELTA) * 0.5

100 CONTINUE

00038 J1 = J0
00039 J0 = J0 + 1
00040 J1 = J1 + 1
00041 D2 = ((2*J1) + 2) * ((2*J1) + 2)
00042 STVL = (12/1) * J1 * J1 / (1 + J1)
00043 XST = 1.234567890 / (1 + STVL)
00044 ST = J1 + XST

100 CONTINUE

00045 XSTVL = XSTVL / (J1 * J1)
00046 DELTAX = DELTA / (J1 * J1)
00047 DELTAY = DELTAY / (J1 * J1)
00048 DELTAX = DELTAX
00049 DELTAY = DELTAY
00050 PRINT 10000
00051 10000 FORMAT(1X,13F13.E9,13X,13F15.3)

00052 10000 FORMAT(1X,13F13.E9,13X,13F15.3)
00053 FORMAT(1X,13F13.E9,13X,13F15.3)
00054 FORMAT(1X,13F13.E9,13X,13F15.3)
00055 FORMAT(1X,13F13.E9,13X,13F15.3)
00056 FORMAT(1X,13F13.E9,13X,13F15.3)
00057 FORMAT(1X,13F13.E9,13X,13F15.3)
00058 FORMAT(1X,13F13.E9,13X,13F15.3)
00059 FORMAT(1X,13F13.E9,13X,13F15.3)
00060 PRINT 77
00061 PRINT 77, (AZY(LL), ZZ = 1, 20)
00062 77 FORMAT(1X,13F13.E9,13X,13F15.3)
TUN 1,2,3 CYCLE 4

1234567890 1234567890

CALL 14

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1

CALL 10

TNT 1
SUBROUTINE THICKEN

PURPOSE: TO AUTOMATE THE THICKNESS AND COMPACTNESS FOR THE DATA FILE FOR PLOTTING.

USAGE:

DEFF THE THICKNESS
DATA THE THICKNESS
PLOT THE PLOTTING OF THE DATA

COMMON
PLATE, DATA, PLOT

EXTERNAL

METHOD:
The thickness values are converted to their ASCII values and written to the output data file. The conversion values are written to the output file. The ASCII values are converted to their ASCII values. The output file is then converted to the ASCII values and written to the output data file.

DATA

CALL write(THICKNESS)

write the thickness and compactness data fields

K = 0
NK = NK - 1

CONTINUE

FILE = THICKNESS
CALL (NAME (1), NAME (1), NAME (1), NAME (1), NAME (1), NAME (1))

CONNECT (NAME (1), NAME (1), NAME (1), NAME (1), NAME (1), NAME (1))

NAME (1) = NAME (1) / NAME (1) + NAME (1)

NAME (1) CONTINUE

NAME (1)
NAME (1) = NAME (1) + NAME (1)

NAME (1) LABEL (1) = NAME (1) + NAME (1)

NAME (1) LABEL (2) = NAME (1) + NAME (1)

NAME (1) PRINT NAME (1) + NAME (1)

NAME (1) LABEL (1) = NAME (1) + NAME (1)

NAME (1) PRINT NAME (1) + NAME (1)

NAME (1) PRINT NAME (1) + NAME (1)

NAME (1) PRINT NAME (1) + NAME (1)
I.

DOCUMENTATION


to output the direction, speed vectors of the jet
for printed and plotted output.

PARAMETERS

ACCY

ACCX

AT

NS

FORECAST VALUE

EXTERNAL

NAME

METH

The wind components are converted to a meteorological
direction and speed for the 10 m AGL. The speed is converted to
km/s and multiplied by 100. These values (direction and speed)
are written to the printing file and the other parameters are deleted.
The direction and speed are then packed into the FILE with the
effective IN GAGE VALUE. This decision is then
written to the printing data base.

EXECUTIVE TABLE IS CALLED TO FORMAT THE DIRECTION AND
SPEED IN THE OUT PUT FILE.

CALL 3=TVACR3((375))

k = 3

k = k + 1

J = J + 1

CALL HIVERO( ) is used to convert real

OP A T R E N T WIND COMPONENTS (MPS) on POLAR COORDINATES

into direction (°) and magnitudes (mps) and the
| PLAN 1.2.1 CYCLE FUNCTION BUILT LAYOUT DIAGRAM | | |
|-----------------------------------------------|------------------|
| 1.11.1. | | |
| 1.11.2. | | |
| 1.11.3. | | |
| 1.11.4. | | |
| 1.11.5. | | |
| 1.11.6. | | |
| 1.11.7. | | |
| 1.11.8. | | |
| 1.11.9. | | |
| 1.11.10. | | |
| 1.11.11. | | |
| 1.11.12. | | |
| 1.11.13. | | |
| 1.11.14. | | |
| 1.11.15. | | |
| 1.11.16. | | |
| 1.11.17. | | |
| 1.11.18. | | |
| 1.11.19. | | |
| 1.11.20. | | |
| 1.11.21. | | |
LISTING 1.4.1 CYCLE THROUGH JOB 2 TO 27 SOURCE LISTING

00001010A(1) = '100
00002010A(1) = '110
00003010A(1) = '120
00004010A(1) = '130
000050CALL 10A(1) = 10A(1)
000060IF (T, N, N) TO 1000

000100K = 0
000110K = K + 1
000123n=2 3n+1 = 2n
0001320
000142010A(1) = 10A(1)
0001520CALL 10A(1) = 10A(1)
0001620IF (T, N, N) TO 1000

000170K = 0
000180K = K + 1
000192Kn = 2 3n+1 = 2n
0001920
000202010A(1) = 10A(1)
0002120CALL 10A(1) = 10A(1)
0002220IF (T, N, N) TO 1000

00023010000 = 10000 = 10000
000240100000 = 10000 = 10000
0002501000000 = 100000 = 100000
000260S A, A
000270S A, A
000280S A, A
000290S A, A
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00051</td>
<td>(QJ3 = Y13 - Y12)</td>
</tr>
<tr>
<td>00052</td>
<td>(QJ6 = Y16 - Z13)</td>
</tr>
<tr>
<td>00053</td>
<td>(QJ2 = Y12 - QJ3)</td>
</tr>
<tr>
<td>00054</td>
<td>(2J1 = 1.0 - QJ2)</td>
</tr>
<tr>
<td>00055</td>
<td>(D41 = (3.0<em>QJ1 + 0.0</em>QJ2 + 0.0*QJ3 + (P11 - P16)*QJ3 + (P12 - P12)*QJ6 + (P13 - P13)*QJ3 + (P14 - P14)*QJ3 + (P15 - P15)*QJ3))</td>
</tr>
<tr>
<td></td>
<td>(+ (3.0<em>QJ1 + 0.0</em>QJ2 + 0.0*QJ3 + (P11 - P16)*QJ3 + (P12 - P12)*QJ6 + (P13 - P13)*QJ3 + (P14 - P14)*QJ3 + (P15 - P15)*QJ3))</td>
</tr>
<tr>
<td></td>
<td>(+ (3.0<em>QJ1 + 0.0</em>QJ2 + 0.0*QJ3 + (P11 - P16)*QJ3 + (P12 - P12)*QJ6 + (P13 - P13)*QJ3 + (P14 - P14)*QJ3 + (P15 - P15)*QJ3))</td>
</tr>
</tbody>
</table>

00057 | \(\text{RETURN}\) |

```

**Special Cases:**

00058 | **Case 1:** \(Y1\) = 0
00059 | \(Q4 = 7(L+1)\)
0064 | \(IF ([Y1, Y1] = Y1)\) \(GO TO 51\)
0065 | \(S1 = P1 + P2 + P3\)
0066 | \(G0 TO 52\)
0067 | \(D3 = 7(-1)\)
0068 | \(IF ([Y1, Y1] = Y1)\) \(GO TO 62\)
0069 | \(S2 = P1 + P2 + P3\)
0070 | \(G0 TO 52\)
0071 | \(D3 = 7(-1)\)
0072 | \(IF ([Y1, Y1] = Y1)\) \(GO TO 63\)
0073 | \(S3 = P1 + P2 + P3\)
0074 | \(G0 TO 52\)
0075 | \(D3 = 7(-1)\)

**Special Formula:**

0076 | \(X12 = XT + XT\)
0077 | \(X13 = XT + X1\)
0078 | \(G13 = (X13 - X13)\)
0079 | \(G12 = (X12 - X12)\)
0080 | \(G11 = (X11 - X11)\)
0081 | \(D4 = (1.0 - P12)*P06 + (1.0 - P12)*P16 + (1.0 - P13)*P13 + (1.0 - P14)*P14 + (1.0 - P15)*P15)\)
0082 | \(\text{RETURN}\)

**Special Case:**

0083 | **Case 3:** \(X11\) = \(Y11\)
0084 | \(\text{RETURN}\)

**Point Z(x,y):**

0085 | \(P11 = (z)\)
0086 | \(\text{RETURN}\)

**End:**

0087 | \(\text{END}\)
LSTRD 1,5,1 CYCLE ENDP

DATA V(1),N, M,N

LSTRD 1,5,1 CYCLE ENDP
010 CONTINUE
015 50 CONTINUE
020 SET 100
025 END
**Program Maintenance Manual Polar Ice Forecast Subsystem - Arctic**

**P. A. Harr, T. C. Pham and J. P. Welsh**

**Control Data Corporation**
205 Montecito Ave.
Monterey, CA 93940

**Naval Air System Command**
Jefferson Davis Highway
Arlington, VA 22210

**Polar Oceanography Branch**
NORDA
NSTL Station, MS 39529

**Program Element, Project, Task Area & Work Unit Numbers**

93321D work unit

**Report Date**

October 1981

**Number of Pages**

146

**Unlimited**

**Unlimited**

**Unlimited**

**Numerical model**

**Sea Ice**

**Polar**

**Arctic**

**Ice Forecasting**

**Program maintenance**

This manual provides a complete and detailed description of the dynamic thermodynamic sea ice model (PIFS-N) including all the necessary information for maintenance programmer personnel required to maintain the system.