Not in DRA

OCEAN DATA SYSTEMS, INC.

alysis

luation of Program
r the SEASAT Program

Landsat Follow-on

Prediction Model

dels--Volume II

tive Equation Forecast
misphere--Volume III

ndustry Economic

, you have any commer

Prepared For

Econ
INCORPORATED

900 State Road
Princeton, NJ 08540

odsi

Contract No. NASW-2558

OCEAN DATA SYSTEMS, INC.
6000 EXECUTIVE BLVD., ROCKVILLE, MARYLAND 20852 • 301/881-3031

Prepared for
ECON, INC.
Princeton, New Jersey

Volume II

METEOROLOGICAL ANALYSIS MODELS

Final Report

Prepared under
Contract No. NASW-2558
Modification 5

July 1976

Prepared by
Rodger A. Langland
Darrell L. Stark
OCEAN DATA SYSTEMS, INC.
Monterey, California

ABSTRACT

As part of the SEASAT program of NASA, two sets of analysis programs were developed for ECON, Inc. under contract NASW-2558. One set of programs produce 63 x 63 horizontal mesh analyses on a polar stereographic grid. The other set produces 187 x 187 third mesh analyses. The parameters analyzed include sea surface temperature, sea level pressure and twelve levels of upper air temperature, height and wind analyses. Both sets use operational data provided by Fleet Numerical Weather Central. The analysis output is used to initialize the primitive equation forecast models also included as part of this contract.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

SECTION I. SCALAR ANALYSIS USING THE PATTERN-CONSERVING TECHNIQUE

A. Introduction

In meteorological analysis, one piece of information that should not be ignored is the most recent past analysis, or, if available, a good forecast valid at the current analysis time. A man doing a hand analysis usually uses such information. In particular, the analyst needs an estimate of the positions of highs and lows (curvature) and of the areas of strong and weak gradients. In referring to the past analysis or forecast, the man usually is looking not so much for absolute magnitudes as for the shape of the field. When he draws his new analysis, he will first attempt to fit the shapes in the past field to the new data. If conflict occurs, the new data takes precedence unless the analyst suspects that the data is in error. In regions where data is routinely sparse, many conflicts need to be resolved because of the accumulation of errors. This results from the cycle of a poor analysis initializing a forecast which is consequently poor which is used as a deficient first guess for the next analysis/prediction cycle.

The best objective analysis scheme is probably one that follows the same rules that a man follows. If such objective techniques can be worked out, the machine may do the job in a more consistent manner than a man is able to do.

The basic goal of this analysis is to fit the following information to varying degrees: the new data; the most recent past analysis or forecast value (the first guess); the gradients of the first guess in eight directions from each grid point; and the Laplacian of the first guess. The degree of fit desired for each piece of information is specified by an array of weights. These variables and weights are named in Table I-1.

The desired fit is realized by minimizing the sum of the deviations of the various characteristics of the analysis from their counterparts in the first guess. The minimization is accomplished with an elementary application of the calculus of variations.

Information is spread through space by the gradient and Laplacian terms. In a surface analysis, there are sometimes natural obstacles (mountain ridges, coastlines, etc.) beyond which an analyst would not allow a new observation to influence the analysis. This kind of constraint can be simulated in the objective analysis by reducing the weights of the gradients and Laplacian along the demarcation zone.

The decision on the magnitudes of the various weights is less arbitrary if we view each weight as the inverse of the variance associated with the parameter it multiplies. This viewpoint is also useful in re-evaluating the weights of the data.

An analysis cycle consists of three basic steps:

1. Assemble the data at grid points.
2. Solve the minimization equation.
3. Re-evaluate the weight of each report.

In order to adequately evaluate the weight of each report, at least two cycles are required. It is desirable to include one additional cycle to allow initially suppressed data to enter the analysis with a high weight if supporting data some distance away causes the analysis to conform more closely to the report after the second cycle. The basic steps are detailed individually in the following sections.

B. Assembly

We shall refer to the guess field as $P_{i,j}$ with weight $A_{i,j}$. On the first cycle, it is the first guess, and $A_{i,j}$ has a low and probably uniform value. On subsequent cycles, $P_{i,j}$ is the result of the previous cycle, but $A_{i,j}$ reverts back to its original value. During assembly, $A_{i,j}$ will be modified to incorporate the data weights, as re-evaluated at the end of the previous cycle.

Each report is moved to the nearest grid point along the gradient of the guess field; that is, the guess field is interpolated to the report location and the difference between the interpolated value and the nearest grid point value is added to produce a pseudo reported value. A weighted average of the guess value and all reports affecting the grid point is done.

$$\tilde{P}_{i,j} = \frac{A_{i,j}P_{i,j} + A_1P_1 + A_2P_2 + \ldots + A_nP_n}{A_{i,j} + A_1 + A_2 + \ldots + A_n}$$

$$\tilde{A}_{i,j} = A_{i,j} + A_1 + A_2 + \ldots + A_n.$$

The superscript (∿) designates assembled values. A_n is the weight of the n^{th} report and P_n is the n^{th} report. In the following sections, the superscript (∿) will not be used. The assembled values will be referred to as $P_{i,j}$.

C. Minimizing the Deviations

TABLE I-1: PCT SCALAR CONSTRAINTS

Constraint	Weight
$P_{i,j}$ = Variable being analyzed (assembled value)	$A_{i,j}$
$\mu_{i,j}$ = y axis gradient = $P_{i,j+1} - P_{i,j}$	$B_{i,j}$
$\nu_{i,j}$ = x axis gradient = $P_{i+1,j} - P_{i,j}$	$C_{i,j}$
$\alpha_{i,j}$ = x-1,y+1 gradient = $P_{i-i,j+1} - P_{i,j}$	$E_{i,j}$
$\beta_{i,j}$ = x+1,y+1 gradient = $P_{i+1,j+1} - P_{i,j}$	$F_{i,j}$
$L_{i,j}$ = Laplacian = $P_{i+1,j} + P_{i-1,j} + P_{i,j+1} + P_{i,j-1} - 4P_{i,j}$	$D_{i,j}$

The first guess shapes μ, ν, α, β and L and their respective weights B, C, E, F and D have a constant value during the entire analysis. Within limits specified by the weights, we require the final analysis to have similar values of the constraints as the first guess field.

To effect this matching, we shall minimize the following integral:

$$I \equiv \iint [$$

(a) $A_{i,j} \; (P^*_{i,j} - P_{i,j})^2 +$ [I.1]

(b) $B_{i,j} \; (P^*_{i,j+1} - P^*_{i,j} - \mu_{i,j})^2 +$

(c) $B_{i,j-1} \; (P^*_{i,j} - P^*_{i,j-1} - \mu_{i,j-1})^2 +$

(d) $C_{i,j} \; (P^*_{i+1,j} -P^*_{i,j} - \nu_{i,j})^2 +$

(e) $C_{i-1,j} \; (P^*_{i,j} - P^*_{i-1,j} - \nu_{i-1,j})^2 +$

(f) $E_{i,j} \; (P^*_{i-1,j+1} - P^*_{i,j} - \alpha_{i,j})^2 +$

(g) $E_{i+1,j-1} \; (P^*_{i,j} - P^*_{i+1,j-1} - \alpha_{i+1,j-1})^2 +$

(h) $F_{i,j} \; (P^*_{i+1,j+1} - P^*_{i,j} - \beta_{i,j})^2 +$

(i) $F_{i-1,j-1} \; (P^*_{i,j} - P^*_{i-1,j-1} - \beta_{i-1,j-1})^2 +$

(j) $D_{i,j} \; (P^*_{i+1,j} + P^*_{i-1,j} + P^*_{i,j+1} + P^*_{i,j-1} - 4P^*_{i,j} - L_{i,j})^2 +$

(k) $D_{i-1,j} \; (P^*_{i,j} + P^*_{i-2,j} + P^*_{i-1,j+1} + P^*_{i-1,j-1} - 4P^*_{i-1,j} - L_{i-1,j})^2 +$

(l) $D_{i+1,j} \; (P^*_{i+2,j} + P^*_{i,j} + P^*_{i+1,j+1} + P^*_{i+1,j-1} - 4P^*_{i+1,j} - L_{i+1,j})^2 +$

(m) $D_{i,j-1} \; (P^*_{i+1,j-1} + P^*_{i-1,j-1} + P^*_{i,j} + P^*_{i,j-2} - 4P^*_{i,j-1} - L_{i,j-1})^2 +$

(n) $D_{i,j+1} \; (P^*_{i+1,j+1} + P^*_{i-1,j+1} + P^*_{i,j+2} + P^*_{i,j} - 4P^*_{i,j+1} - L_{i,j+1})^2$

$$]\,dxdy$$

In the above, the starred quantities are the analysis values we are seeking. Each term is a departure from the desired matching of differential properties. Extra terms have been added to account for the effect of a changing $P^*_{i,j}$ on the differential properties computed at surrounding points. Their effect is to more closely couple neighboring grid points. See Figure I-1 for a depiction of the minimization stencil as it relates to the terms of equation [I.1].

To minimize the integral, we simply take the first variation with respect to $P^*_{i,j}$, and set it to zero (see equation [I.2]). The solution of the resulting equation will be the $P^*_{i,j}$ that will cause the integral to be minimized. The fact that each term is squared insures a minimum as opposed to a maximum value.

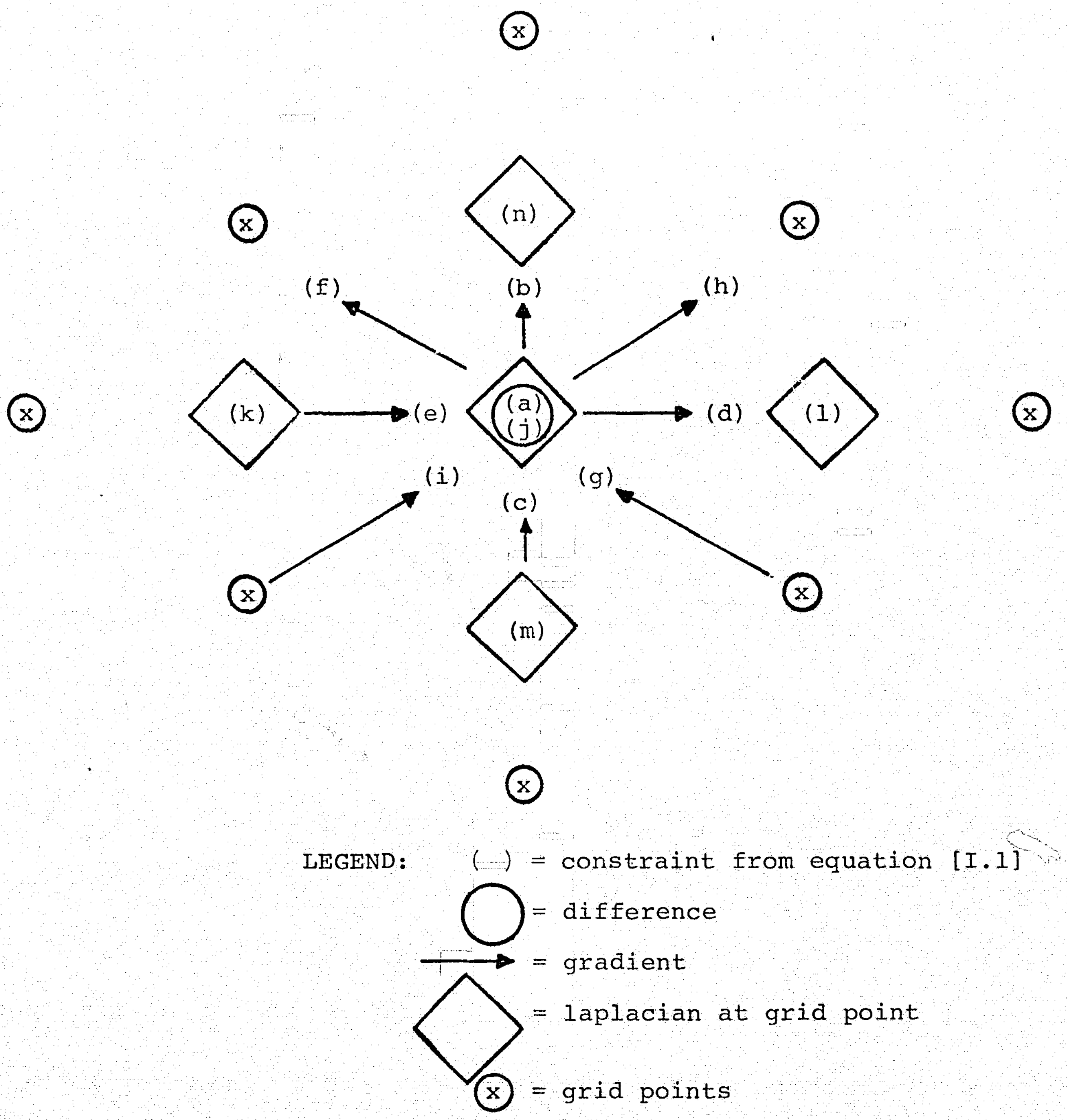


FIGURE I-1: SCALAR MINIMIZATION STENCIL

$$
\begin{aligned}
\frac{\delta I}{\delta P^*} = \iint [\ & 2A_{i,j}\ (P^*_{i,j} - P_{i,j}) \\
& -2\ B_{i,j}\ (P^*_{i,j+1} - P^*_{i,j} - \mu_{i,j}) \\
& +2\ B_{i,j-1}\ (P^*_{i,j} - P^*_{i,j-1} - \mu_{i,j-1}) \\
& -2\ C_{i,j}\ (P^*_{i+1,j} - P^*_{i,j} - \nu_{i,j}) \\
& +2\ C_{i-1,j}\ (P^*_{i,j} - P^*_{i-1,j} - \nu_{i-1,j}) \\
& -2\ E_{i,j}\ (P^*_{i-1,j+1} - P^*_{i,j} - \alpha_{i,j}) \\
& +2\ E_{i+1,j-1}\ (P^*_{i,j} - P^*_{i+1,j-1} - \alpha_{i+1,j-1}) \\
& -2\ F_{i,j}\ (P^*_{i+1,j+1} - P^*_{i,j} - \beta_{i,j}) \\
& +2\ F_{i-1,j-1}\ (P^*_{i,j} - P^*_{i-1,j-1} - \beta_{i-1,j-1}) \\
& -8\ D_{i,j}\ (P^*_{i+1,j} + P^*_{i-1,j} + P^*_{i,j+1} + P^*_{i,j-1} - 4P^*_{i,j} - L_{i,j}) \\
& +2\ D_{i-1,j}\ (P^*_{i,j} + P^*_{i-2,j} + P^*_{i-1,j+1} + P^*_{i-1,j-1} - 4P^*_{i-1,j} - L_{i-1,j}) \\
& +2\ D_{i+1,j}\ (P^*_{i+2,j} + P^*_{i,j} + P^*_{i+1,j+1} + P^*_{i+1,j-1} - 4P^*_{i+1,j} - L_{i+1,j}) \\
& +2\ D_{i,j-1}\ (P^*_{i+1,j-1} + P^*_{i-1,j-1} + P^*_{i,j} + P^*_{i,j-2} - 4P^*_{i,j-1} - L_{i,j-1}) \\
& +2\ D_{i,j+1}\ (P^*_{i+1,j+1} + P^*_{i-1,j+1} + P^*_{i,j+2} + P^*_{i,j} - 4P^*_{i,j+1} - L_{i,j+1}) \\
&]\ dxdy \overset{set}{=} 0
\end{aligned}
\tag{I.2}
$$

The terms in $\frac{\delta I}{\delta P*}$ can be grouped into three categories:

1. Those involving $P*_{i,j}$.
2. Those involving P* at surrounding points.
3. Those not involving P*.

$$S_{i,j}P^*_{i,j} \left\{ \begin{aligned} &[\, A_{i,j} + B_{i,j} + B_{i,j-1} + C_{i,j} + C_{i-1,j} + E_{i,j} \\ &+ E_{i+1,j-1} + F_{i,j} + F_{i-1,j-1} + 16\, D_{i,j} + D_{i-1,j} \\ &+ D_{i+1,j} + D_{i,j-1} + D_{i,j+1}\,] \quad P^*_{i,j} \end{aligned} \right.$$

$$-H_{i,j} \left\{ \begin{aligned} &- B_{i,j}\, P^*_{i,j+1} - B_{i,j-1}\, P^*_{i,j-1} - C_{i,j}\, P^*_{i+1,j} - C_{i-1,j}\, P^*_{i-1,j} \\ &- E_{i,j}\, P^*_{i-1,j+1} - E_{i+1,j-1}\, P^*_{i+1,j-1} - F_{i,j}\, P^*_{i+1,j+1} \\ &- F_{i-1,j-1}\, P^*_{i-1,j-1} - 4\, D_{i,j}\, P^*_{i+1,j} - 4\, D_{i,j}\, P^*_{i-1,j} \\ &- 4\, D_{i,j}\, P^*_{i,j+1} - 4\, D_{i,j}\, P^*_{i,j-1} + D_{i-1,j}\, P^*_{i-2,j} \\ &+ D_{i-1,j}\, P^*_{i-1,j+1} + D_{i-1,j}\, P^*_{i-1,j-1} - 4\, D_{i-1,j}\, P^*_{i-1,j} \\ &+ D_{i+1,j}\, P^*_{i+2,j} + D_{i+1,j} P^*_{i+1,j+1} + D_{i+1,j}\, P^*_{i+1,j-1} \\ &- 4\, D_{i+1,j}\, P^*_{i+1,j} + D_{i,j-1}\, P^*_{i+1,j-1} + D_{i,j-1}\, P^*_{i-1,j-1} \\ &+ D_{i,j-1}\, P^*_{i,j-2} - 4\, D_{i,j-1}\, P^*_{i,j-1} + D_{i,j+1}\, P^*_{i+1,j+1} \\ &+ D_{i,j+1}\, P^*_{i-1,j+1} + D_{i,j+1}\, P^*_{i,j+2} - 4\, D_{i,j+1}\, P^*_{i,j+1} \end{aligned} \right.$$

$$-G_{i,j} \left\{ \begin{aligned} &- A_{i,j}\, P_{i,j} + B_{i,j}\, \mu_{i,j} - B_{i,j-1}\, \mu_{i,j-1} + C_{i,j}\, \nu_{i,j} \\ &- C_{i-1,j}\, \nu_{i-1,j} + E_{i,j}\, \alpha_{i,j} - E_{i+1,j-1}\, \alpha_{i+1,j-1} \\ &+ F_{i,j}\, \beta_{i,j} - F_{i-1,j-1}\, \beta_{i-1,j-1} + 4\, D_{i,j}\, L_{i,j} \\ &- D_{i-1,j}\, L_{i-1,j} - D_{i+1,j}\, L_{i+1,j} - D_{i,j-1}\, L_{i,j-1} \\ &- D_{i,j+1}\, L_{i,j+1} \end{aligned} \right.$$

Note that all terms in S and G except $A_{i,j}$ in $S_{i,j}$ and $-A_{i,j}\ P_{i,j}$ in $G_{i,j}$ involve first guess pattern information which is constant during the analysis.

The minimization may be written as

$$S_{i,j} \; P^*_{i,j} - (G_{i,j} + H_{i,j}) = 0 \qquad [I.3]$$

In $H_{i,j}$, let us group together the coefficients of P^* at each point.

$$\begin{aligned}
- H_{i,j} = {} & D_{i-1,j} \; P^*_{i-2,j} + (-C_{i-1,j} - 4\; D_{i,j} - 4\; D_{i-1,j}) \; P^*_{i-1,j} \\
& + (-C_{i,j} - 4\; D_{i,j} - 4\; D_{i+1,j}) \; P^*_{i+1,j} \\
& + D_{i+1,j} \; P^*_{i+2,j} + (-E_{i,j} + D_{i-1,j} + D_{i,j+1}) \; P^*_{i-1,j+1} \\
& + (-B_{i,j} - 4\; D_{i,j} - 4\; D_{i,j+1}) \; P^*_{i,j+1} \\
& + (-F_{i,j} + D_{i+1,j} + D_{i,j+1}) \; P^*_{i+1,j+1} \\
& + (-F_{i-1,j-1} + D_{i-1,j} + D_{i,j-1}) \; P^*_{i-1,j-1} \\
& + (-B_{i,j-1} - 4\; D_{i,j} - 4\; D_{i,j-1}) \; P^*_{i,j-1} \\
& + (-E_{i+1,j-1} + D_{i+1,j} + D_{i,j-1}) \; P^*_{i+1,j-1} \\
& + D_{i,j-1} \; P^*_{i,j-2} + D_{i,j+1} \; P^*_{i,j+2}
\end{aligned}$$

Define:

$$\begin{aligned}
X_{i,j} & \equiv C_{i,j} + 4\; (D_{i,j} + D_{i+1,j}) \\
Y_{i,j} & \equiv B_{i,j} + 4\; (D_{i,j} + D_{i,j+1}) \\
Z_{i,j} & \equiv - F_{i,j} + D_{i+1,j} + D_{i,j+1} \\
R_{i,j} & \equiv - E_{i+1,j} + D_{i,j} + D_{i+1,j+1}
\end{aligned}$$

Note that X, Y, Z and R have a constant value during the analysis.

Then

$$\begin{aligned} -H_{i,j} = {} & D_{i-1,j} P^*_{i-2,j} - X_{i-1,j} P^*_{i-1,j} - X_{i,j} P^*_{i+1,j} \\ & + D_{i+1,j} P^*_{i+2,j} + R_{i-1,j} P^*_{i-1,j+1} - Y_{i,j} P^*_{i,j+1} \\ & + Z_{i,j} P^*_{i+1,j+1} + Z_{i-1,j-1} P^*_{i-1,j-1} - Y_{i,j-1} P^*_{i,j-1} \\ & + R_{i,j-1} P^*_{i+1,j-1} + D_{i,j-1} P^*_{i,j-2} - D_{i,j+1} P^*_{i,j+2} \end{aligned} \quad [I-4]$$

The minimization equation [I.3] is solved by simultaneous over-relaxation. The matrices $S_{i,j}$ and $G_{i,j}$ may be computed initially except for the $A_{i,j}$ term and will not change throughout the analysis. Matrix $H_{i,j}$ must be recomputed for every iteration of the relaxation.

The relaxation proceeds as follows: At Point (i,j) the terms of the minimization equation are evaluated using the assembled P field for P*. In general, the equation is not satisfied and a residual is defined as

$$S_{i,j} \, P^{*\,\tau}_{i,j} - (G_{i,j} + H_{i,j}) \equiv R \qquad [I.5]$$

The superscript τ is an iteration counter. The value of $P^*_{i,j}$ is to be altered so that on the next iteration, the residual will be zero, provided $H_{i,j}$ does not change. Of course, $H_{i,j}$ will change, but if the equation is fairly well behaved, repetition of the procedure should lead to convergence on the correct solution.

$$S_{i,j} \, P^{*\,\tau+1}_{i,j} - (G_{i,j} + H_{i,j}) = 0 \qquad [I.6]$$

Subtracting [I.6] from [I.5],

$$S_{i,j} \, (P^{*\,\tau}_{i,j} - P^{*\,\tau+1}_{i,j}) = R \qquad [I.7]$$

and

$$P^{*\,\tau+1}_{i,j} = P^{*\,\tau}_{i,j} - \frac{R}{S_{i,j}}$$

Convergence can be hastened by increasing the correction term in [I.7]. The factor by which it is increased is called the over-relaxation coefficient. Equation [I.7] becomes

$$P^{*\,\tau+1}_{i,j} = P^{*\,\tau}_{i,j} - \text{ALFA}\,\frac{R}{S_{i,j}} \qquad [I.8]$$

ALFA must be between 1 and 2.

One iteration consists of making the correction [I.8] at every grid point. Iterations are repeated until the maximum residual is less than a specified convergence criterion. The resulting P* field is the solution of equation [I.3].

D. Calculating The Resultant Weight

In solving the minimization equation, we started with an assembled guess $P_{i,j}$ with weight $A_{i,j}$. After the equation is solved, the quantity $A_{i,j}P_{i,j}$ has been altered by adding a quantity, $A^a_{i,j}P^a_{i,j}$, which we shall call ambient information. The resultant value is

$$A^*_{i,j}P^*_{i,j} = A_{i,j}P_{i,j} + A^a_{i,j}P^a_{i,j} \qquad [I.9]$$

The resultant weight $A^*_{i,j}$ will be useful in re-evaluating the weights of the data. Holl and Mendenhall (1972) show that $A^*_{i,j}$ could be obtained by inverting a large matrix. For large grids, it is not computationally feasible to do so, and Holl suggests a practical alternative. According to [I.9], a small change in the value of $P_{i,j}$ would have the following effect on the outcome of solving the minimization equation:

$$A^*_{i,j}\ \delta P^*_{i,j} = A_{i,j}\ \delta P_{i,j} \qquad [I.10]$$

By perturbing the assembled $P_{i,j}$ at grid points which had data and re-solving the minimization equation, the response in $P^*_{i,j}$ is found. From [I.10]

$$A^*_{i,j} = A_{i,j}\,\frac{\delta P_{i,j}}{\delta P^*_{i,j}}$$

The re-solution of [I.3] is done only in the vicinity of the point in question to obtain a value for $\delta P^*_{i,j}$.

E. Re-evaluating The Data Weights

The quality of each observation is judged according to the effect that its removal has on the analysis. If an observation causes a change that is contrary to the guess field, to the differential properties of the guess field and to any other available observations, then its removal will cause a substantial change in the analysis. We have little confidence in such a report and would like to reduce its weight.

At the end of each cycle, the weight of each report is re-evaluated. For a particular report, the analysis value with the report removed is called the background analysis value, P_B, and the resultant analysis weight with the report removed is called the background analysis weight, A_B. To determine the impact of removing the report, A_B and P_B are computed:

$$A_B = A^*_{i,j} - A_n$$
$$P_B = (A^*_{i,j}P^*_{i,j} - A_nP_n)/A_B \qquad [I.11]$$

The starred values are the result of the solution of [I.3] on the present cycle. A_n is the weight which the report in question had during the present cycle.

It was mentioned on page I-2 that each weight is viewed as the inverse of a variance. Accordingly, the variance of the report is $\frac{1}{A_o}$, where A_o is the weight the report had originally. The variance of the background analysis value for this report is $\frac{1}{A_B}$. Therefore, the expected difference between the report and its background value is $(\frac{1}{A_o} + \frac{1}{A_B})^{1/2}$. The actual difference is $P_n - P_B$. If the actual difference exceeds the expected difference, we shall reduce the weight of the report proportionately.

Defining $$\lambda^2 \equiv \frac{(P_n - P_B)^2}{\frac{1}{A_o} + \frac{1}{A_B}}$$

which through manipulation and substitution of equations [I.11] converts to

$$\lambda^2 = \frac{A_o(A^*_{i,j})^2(P_n - P^*_{i,j})}{(A^*_{i,j}-A_n+A_o)(A^*_{i,j}-A_n)}$$

The re-evaluated weight is defined as $A_{nR} = \frac{2A_o}{1+\lambda^2}$

This results in:

$\lambda^2 > 1$ implies actual error > expected error $\therefore A_{nR} < A_o$

$\lambda^2 = 1$ implies actual error = expected error $\therefore A_{nR} = A_o$

$\lambda^2 < 1$ implies actual error < expected error $\therefore A_{nR} > A_o$

Notice that on any cycle, every data point may have its original weight restored, even if it had been reduced previously. In this way, a report that causes a large change in the analysis may have full effect if it is supported by data nearby.

F. Program Description - 63 x 63 Grid Version

1. PCT

The calling arguments are described in detail in the comments of the program. All the arrays are variably dimensioned, using the dimensions M and N provided in the calling arguments. the date-time group is provided through common block /DTG/, a title for plotting, a contour starting point and a contour interval are in common block /INFO/ and sense switch variables are passed in /ISW/.

A random-access file TAPE9 is used for temporary storage and must be declared on the PROGRAM card of the calling program. The writing of some arrays on the random-access file for later retrieval allows their use as work arrays. Immediately after opening the file, the data list, the I and J data location lists, the initial data weight list and the initial weight field of the first guess are all written on the random-access file so that these arrays can be used in subroutine BKGRND. Since the same array is frequently used to hold two different fields, two names, separated by a space, make up the names of these arrays. Of course, the space is ignored by Fortran. The two names are interpreted as one, but this convention helps in reading the listing.

After BKGRND computes matrices S and G (see page I-10), they are written on TAPE9 and the data-related arrays are read back in. The weights of the Laplacian field (D) are also written on TAPE9 so that the array can be used later as a work array.

DO loop 100 is the main loop controlling the number of cycles to be made through the program. A cycle consists of assembly (Section I- B), solving Equation [I.3] (Section I-C), computing the resultant weight field (Section I-D) and re-evaluating the data weights (Section I-E). Before doing the assembly, the original weights of the first-guess field are read from TAPE9. During assembly, this field is altered by adding the data weights. The latest solution field of P, the field being analyzed, is also read from TAPE9 and used as the guess field for this assembly. Except for the first cycle, the root-mean-square difference between the observations that were accepted on that cycle and the resulting analysis of that cycle is computed.

After calling ASSMBL on the last analysis cycle only, subroutine PLTDAT (see Appendix) writes the data list on the plot file. The data list is rewritten on TAPE9 because, in ASSMBL, gross errors were flagged by setting the last bit of the data word. The last bit of all good data is cleared. The assembled weight field A is written on TAPE9 as a record named OLDA, so that A can be used as a work array. This field is called "OLDA" only to distinguish it from the resultant weight which will be computed in REVALWT. Matrices

G and S (see page I-10) are read from TAPE9 and subroutine UPDATE adds A to S and A*P to G. Next, the current guess field P is written on TAPE9 and named OLDP. Weight field D is read from TAPE9 and subroutine BLEND solves Equation [I.3], resulting in a new analysis field which is written on TAPE9 and called NEWP. On the last analysis pass only, the analyzed field is passed through an optional short wave filter and subroutine PLOT (see Appendix) writes the analyzed field on the plot file and PRT makes a printer map of it. Depending upon external sense switch settings the analysis field may also be written to the disk using the Fleet Numerical Weather Central (FNWC) random access routine ZRANDIO (see Appendix).

The current data weight is packed into the left half of each word in DWT and the original data weight in the right half. The original weight is needed for the weight re-evaluation. Next, the guess field that was used in the call to BLEND on this cycle (OLDP) is packed with the latest analysis field (NEWP). Array DATA is used to hold these fields and to pass them to subroutine REVALWT. The assembled weight field OLDA is read from TAPE9 and passed to REVALWT. After REVALWT computes new values of DWT, arrays DATA, AI and AJ are read from TAPE9 for the next cycle. At the end of PCT, file TAPE9 is closed.

2. BKGRND

Matrices S, G, X, Y, Z and R (see pages I-10 and I-11) are computed and returned. There is no problem in interpreting the code with the aid of the comments.

3. ASSMBL

The data is moved to the nearest grid point and the weighted average of the data and the guess value computed. Array SUM is first initialized to A*P, where A is the weight of the guess field and P is the guess field. Then, the data list is scanned and the guess field interpolated to each report location. If the interpolated value differs from the report by more than GROS, the report is rejected by setting the last bit of the report word. Otherwise, this bit is cleared. If the report is accepted, the difference between the interpolated guess and the report is added to the guess value at the nearest grid point and the result stored in array DATAM. The I and J grid point coordinates to which each report was moved are packed in the lower nineteen bits of each word of DATAM with the last bit corresponding to the last bit of DATA.

The product of the data weight times the data value, after being moved to the nearest grid point, is added to SUM(I,J) and the data weight is added to A(I,J). After all data have been scanned, SUM is divided by A to obtain the weighted average.

4. UPDATE

The current assembled weight field A is added to matrix S (see page I-10 and the BKGRND listing) and A*P is added to matrix G, where P is the current guess field.

5. BLEND

The minimization equation [I.3] is solved by simultaneous over-relaxation. The method is described in detail on pages I-13 and I-14. No further discussion is warranted.

6. REVALWT

The resultant weight at grid points which have data is computed as described in Section I-D. The weight of each datum is re-evaluated as discussed in Section I-E. A single pass through array DATAM is made. If the datum was not rejected in ASSMBL on the current cycle, the I,J coordinates of the point to which it was moved are unpacked from the lower nineteen bits of the word.

The re-solution of Equation [I.3] is done on a circle of three-grid-interval radius surrounding the point. The J limits of this area are put into JB (bottom of the area) and JT (top of the area). The I limits for each row (seven rows total) go into the seven-element arrays IL (left of area) and IR (right of area). The solution can be done only up to the third interior row.

A perturbation of five percent of the assembled value P is added to P and the resulting change to matrix G is computed. The data weight before re-evaluation is saved to be printed out later. The re-solution over the limited area is the same as in BLEND. Once convergence is obtained, G is restored to its correct value and the resultant weight computed from the equation at the bottom of page I-15. Then, the new data weight is calculated as specified on page I-17 and the datum, its old weight and its new weight are printed out. Finally, the guess field P is restored to the values it had over the limited area before the re-solution.

G. Program Description - 187 x 187 Grid Version

1. Design Philosophy

A major consideration in the program design is the grid size. Program specifications call for a 187 x 187 grid, which means 34,969 computer words for each required array. Even if some data packing scheme is implemented, the resulting arrays are too large and too numerous for the computer central memory capacity. Therefore, the program has been designed to process the 187 x 187 grid by partitions. Sixteen partitions are defined as follows:

I Direction	J Direction	Partition
3 - 48	3 - 48	1
3 - 48	49 - 94	2
3 - 48	95 - 140	3
3 - 48	141 - 186	4
49 - 94	3 - 48	5
49 - 94	49 - 94	6
49 - 94	95 - 140	7
49 - 94	141 - 186	8
95 - 140	3 - 48	9
95 - 140	49 - 94	10
95 - 140	95 - 140	11
95 - 140	141 - 186	12
141 - 186	3 - 48	13
141 - 186	49 - 94	14
141 - 186	95 - 140	15
141 - 186	141 - 186	16

The border points 1, 2 and 187 are used to process neighboring points, but are not, themselves, processed. To assure continuity along the borders of partitions, sixteen 50 x 50 arrays are defined as follows:

Partition array	I Direction	J Direction
1	1 - 50	1 - 50
2	1 - 50	47 - 96
3	1 - 50	93 - 142
4	1 - 50	139 - 188
5	47 - 96	1 - 50
6	47 - 96	47 - 96
7	47 - 96	93 - 142
8	47 - 96	139 - 188
9	93 - 142	1 - 50
10	93 - 142	47 - 96
11	93 - 142	93 - 142
12	93 - 142	139 - 188
13	139 - 188	1 - 50
14	139 - 188	47 - 96
15	139 - 188	93 - 142
16	139 - 188	139 - 188

The points 188 are a repeat of points 187. This design allows a 46 x 46 partition to be processed by itself without loss of continuity. Note that each 50 x 50 array contains a 46 x 46 partition. The 50 x 50 array will be referred to as a partition in this document. The term "array" is usually used to refer to 187 x 187 arrays.

For an example of processing by partition, consider the program SSTHEM. This program uses nine 187 x 187 arrays having the names A, B, C, D, E, F, P, G and S. Central memory has storage providing for one partition at a time of each of these nine arrays. The remaining partitions must be stored outside of central memory. The program has been designed to store up to sixteen partitions of nine arrays in either extended core storage or on a mass storage file. A partition is brought into central memory when it is needed. During the processing of the partition the logic may require that the partition values be permanently altered. In this case, the partition is returned to storage with its new values in it. The program logic requires that old values of the array P be saved along with newly computed values. The old valued array is stored by partition on a separate mass storage file.

The re-evaluation of data weights requires that a block of points from each of the arrays A, B, C, D, E, F, P, G, S and old P be in core for each data point to be processed, A block must contain points plus five and minus five from the data point in both the I and H directions. The 16 partitions of an array are not adequate for this requirement. To accomplish the re-evaluation of data weights, an additional 33 partitions have been defined. Each additional partition is a composite of the first 16 partitions. Just prior to executing the PCT routine (described in the next numbered paragraph), the subroutine ASSIGN is called to have a partition number assigned to each data report. Partition numbers are in the range 1 through 9 and are stored in the table KPART which has an entry for each data report. For convenience, the relative coordinates of the data point in the partition are determined and also stored in the KPART entry. For greater efficiency a table of the partitions that actually get assigned is built. This table is called IWHAT and contains an entry for each of the 49 partitions. The entry contains the number of reports that the partition is assigned to. A zero entry means that the partition is not assigned to any reports.

The subroutine REVALWT is called on each pass from the PCT routine. REVALWT computes data weights, a partition at a time. Any partition with a zero entry in IWHAT is not processed. For each partition processed, REVALWT computes a new data weight for each of the reports having the partition assigned to it. REVALWT calls the subroutine EXTRCT to have array data brought into core. EXTRCT retrieves array data by partition number. For partitions 1 through 16, EXTRCT retrieves a single 50 x 50 block of each of the required arrays. For higher partition numbers, EXTRCT retrieves the appropriate 50 x 50 blocks and builds the required composites. A composite set consists of a 50 x 50 block of each of the required arrays.

2. PCT

A subroutine interaction chart (Figure I-2) for the program SSTHEM show the relationship of the PCT routine to a program package.

When the mass storage option is selected, a random-access file is required for partition storage and the number of this file is the variable JF in the common block /MEM/. The value of JF must be set by the calling program. The TAPE9 file is opened by PCT, but the JF file is opened on a first call to the subroutine INDAT (see appendix) and, only when the mass storage option has been selected.

The PCT logic flow consists, first, of calling the subroutine BKGRND to have the matrices S and G computed (see page I-10), the second, of executing one or more cycles of calls to ASSMBL, UPDATE, BLEND and REVALWT. DO loop 100 is the main loop controlling the number of cycles to be made through the program.

By calling INFIELD the first guess field and all necessary constant fields are generated, partitioned and stored away for subsequent processing.

To implement partition processing the subroutine BKGRND is called sixteen times, once for each partition. Prior to each call, the subroutine INDAT is called to bring in a partition's worth of data for each of the arrays required by BKGRND. After each call to BKGRND a call to OUTDAT (see appendix) stores a partition's worth of data

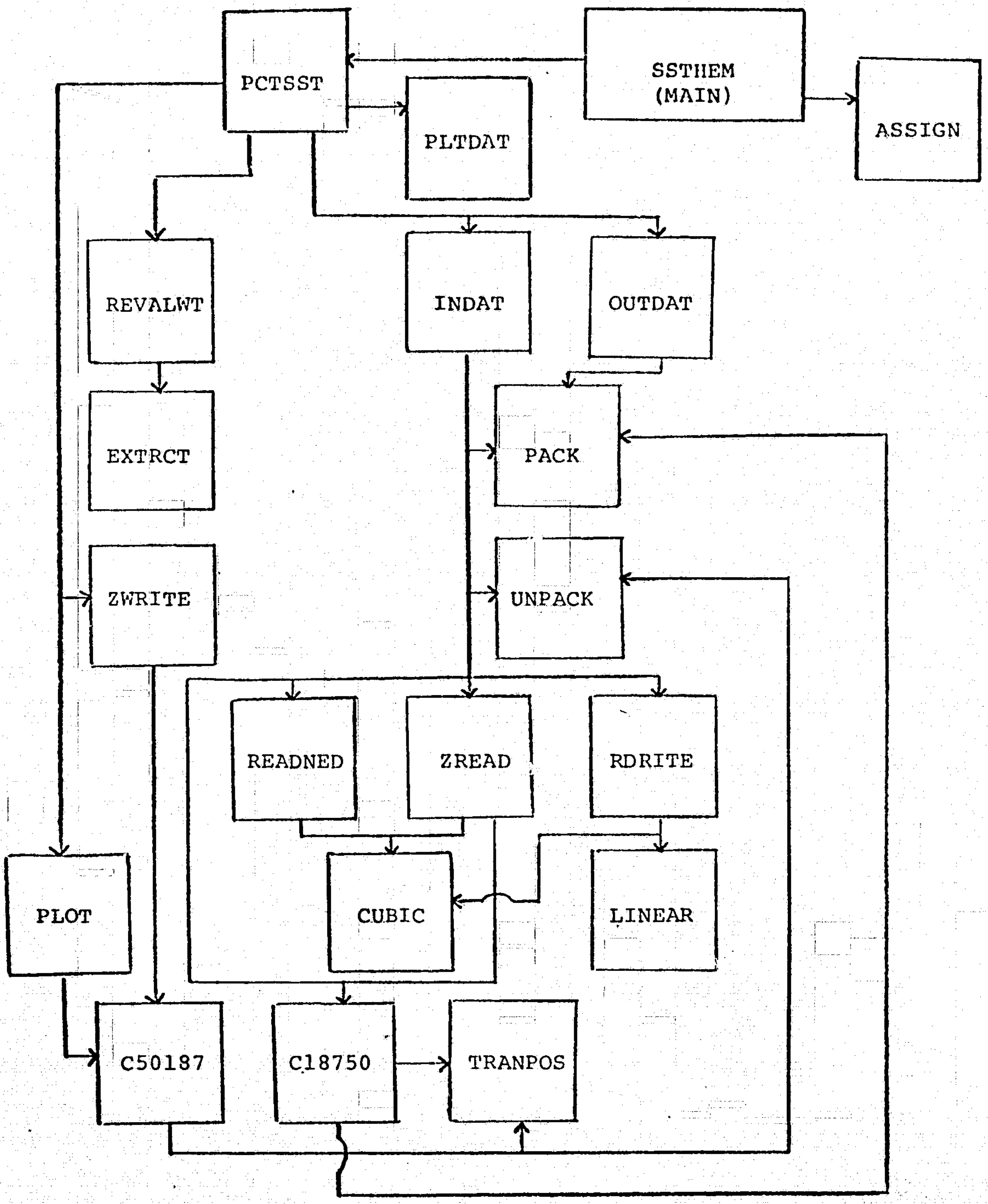


FIGURE I-2 : SUBROUTINE INTERACTION CHART

for each array computed or updated by BKGRND. In each cycle, the subroutines ASSMBL and UPDATE are called sixteen times using the routines INDAT and OUTDAT in the same manner. The BLEND routine is called sixteen times per iteration and the routines INDAT and OUTDAT, again, are used in the same manner. Iterations are executed until convergence is obtained, or until 150 iterations have been executed without convergence. When convergence is not obtained, PCT terminates the program run. In each cycle it is necessary to call REVALWT just once. REVALWT determines for itself what partitions are required and retrieves these through the subroutine EXTRCT (see appendix).

3. BKGRND

This routine is identical to the 63 x 63 version.

4. ASSMBL

The current partition number is passed to ASSMBL in the calling sequence, and only reports having coordinates within the partition are processed. By the time ASSMBL has been called for all partitions, all reports will have been processed.

5. UPDATE

This routine is identical to the 63 x 63 version.

6. BLEND

The minimization equation [I-3] is solved by simultaneous over-relaxation for each partition with the maximum residual being passed back to PCT in common block /BLSTUF/.

7. REVALWT

To implement partition processing, REVALWT makes use of the tables KPART and IWHAT built by the routine ASSIGN (see appendix) and located in the common block /REVAL/. In a main processing loop 49 partitions are candidates for processing. When a partition's IWHAT entry is non-zero, all reports having the partition assigned are given an updated data weight on a single pass through array DATAM. If the datum was not rejected in ASSMBL on the current cycle, the I, J coordinates of the point to which it was moved are unpacked from the lower nineteen bits of the word and may be used for printout purposes. The report coordinates relative to its assigned partition are unpacked form the leftmost 40 bits of the report's KPART entry. (the right-most 20 bits of the entry contain the assigned partition number.)

The re-solution of Equation [I-3] is done in the same manner as in the 63 x 63 version.

8. Common Storage Areas - Peculiar to the 187 x 187 Grid Version

Blank Common

Storage blocks in blank common have been arranged to accomodate a partition's worth of data from each of nine 187 x 187 data arrays in addition to certain other working arrays. For example, blank common used by the SSTHEM program is defined as follows:

```
COMMON A(50,50), B(50,50), C(50,50), D(50,50),
E(50,50), F(50,50), P(50,50), DATA(3969), DI(3969),
DJ(3969), DWT(3969), DATAM(3969), OLDP(50,50),
SAVP(50,50), SAVG(50,50)
```

The arrays DI and DJ also hold data for the S and G matrices, respectively. This arrangement may not be disturbed without corresponding logic changes being made in the various routines which reference the blank common area.

MEM Common

The labeled common MEM contains cells and arrays defined as follows:

Name	Description
MEMTYP	This cell is set from a data card field. 0 means use the extended core storage option. 1 means use the mass storage option.

NPART This cell contains the number of partitions into which 187 x 187 arrays are divided. It is set to the value 16.

JF This cell contains the file number of the storage file used in mass storage mode.

ICODE This cell contains flags indicating the arrays to be brought in from storage. The right-most nine octal digits are used. Arrays 1 through 9 are assigned in order from left to right. An octal "1" means bring in a partition of the array; an octal "0" means don't.

JCODE This cell contains flags indicating the arrays to be written to storage. The flags are assigned similar to those of ICODE.

SC(2500) This is a scratch array large enough to hold a partition's worth of data from any 187 x 187 array.

OPTION Common

The labeled common OPTION contains cells defined as follows:

Name Description

INTTYP This cell is set from a data card field to indicate that input field data are in a 63 x 63 grid or a 187 x 187 grid. "0" means 187 x 187; "1" means 63 x 63.

KKODE This cell contains flags indicating the partitions of the field arrays to be displayed by the routine PRT. The right-most sixteen octal digits of the word are used. Partitions 1 through sixteen are assigned in order from right to left. An octal "1" means display; an octal "0" means don't. (Set from card input.)

IVEC This cell is set to "0" by programs working with scalar fields, and to "1" by programs working with vector fields.

BLSTUF Common

The labeled common BLSTUF is used by PCT to pass parameters to BLEND as follows:

Name — Description

MM2 and NM2 — These cells contain variables used by BLEND to limit the number of points processed in a partition.

ALPHA — This is the alpha variable used by by BLEND in the over-relaxation process.

RMAX — This cell contains the current value of the maximum residual obtained in the BLEND convergence process.

REVAL Common

The labeled commmon REVAL contains the following arrays:

Name — Description

KPART(500) — This array contains an entry for each data report. Each entry contains a partition number and a set of coordinates for the report. Starting from the left, the first 20 bits contain the coordinate I, the second 20 bits contain the coordinate J and the last 20 bits contain the partition number of the partition assigned to the report. The coordinates are relative to the partition area.

IWHAT — This array contains an entry for each of 49 partitions. Each entry contains the number of reports to which the partition is assigned.

SECTION II. VECTOR WIND ANALYSIS USING THE PATTERN-CONSERVING TECHNIQUE

A. Introduction

The pattern-conserving technique described in Section I is used to analyze a scalar variable. In this section, we will concentrate on those aspects which are peculiar to the wind problem.

The most essential feature of the pattern-conserving technique is that, while fitting new data, it tends to retain certain differential properties of the first-guess field. For scalar analysis, we were only concerned with gradients and the Laplacian. The wind, being a vector, complicates the problem slightly. Some of the properties we would like to conserve; e.g., vorticity and divergence, involve both scalar components. We must analyze both components simultaneously.

The differential properties that we choose to conserve are the gradients of each wind component in eight directions from each grid point, the vorticity and the divergence. The same method is used here as in the scalar analysis, the main difference being that two minimization equations rather than one must be solved simultaneously.

The equations are considerably simplified by using the staggered grid illustrated by Figure II-1 and defining the divergence, vorticity and gradients as in Table II-1 and Figure II-2. This arrangement causes certain matrices to be tridiagonal.

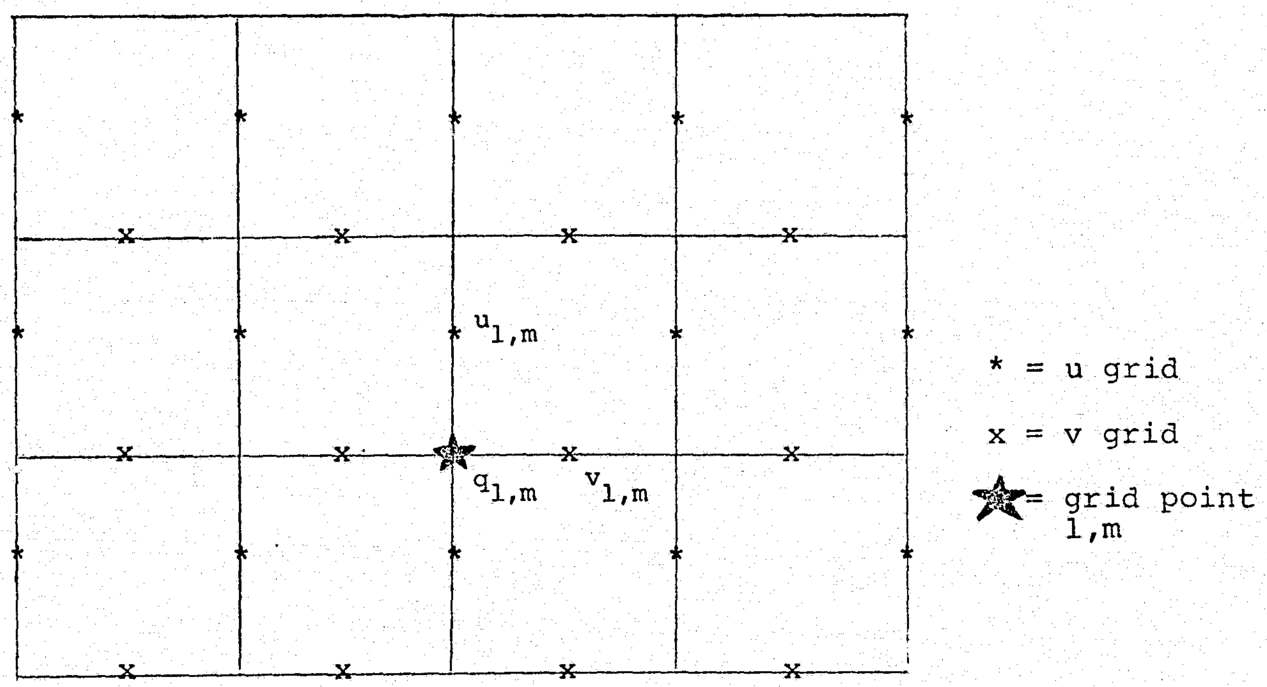


FIGURE II-1: STAGGERED u,v GRID

The re-evaluation of the data weights has been simplified for the wind analysis because of the increased complexity of the program and because wind reports have more inherent variance than most scalar variables.

B. Assembly

The assembly of data to grid points is done in the same way as in the scalar analysis. The wind components are moved to the nearest grid point along the gradient of the guess field, and a weighted average is computed at each grid point for each component. Since the grid is staggered (Figure II-1), the components of an observation *may be moved to different grid points*; i.e., to grid points with different indices.

For the wind analysis, we need a field of assembly weights (A) for each component (AU and AV). Both AU and AV are initialized as the class weight of the first guess. When a U component is assembled to a grid point (I,J), the data weight of the report is added to AU (I,J). The same data weight applies to both components of a report. Similarly, the V component is assembled to a grid point (I,J) and the data weight is added to AV(I,J).

C. Minimizing the Deviations

In the scalar analysis, we wanted to conserve the gradients and the Laplacian. With the regular grid, the finite difference expressions for the gradients and Laplacian did not provide good horizontal coupling among the grid points, so we included in the integral to be minimized the gradients and Laplacian at surrounding grid points. In the case of the wind analysis, the more complex differential properties and the staggered grid extend the influence of the computations at a grid point further than in the scalar analysis, and it is not necessary to add the contributions at the surrounding points.

TABLE II-1: PCT VECTOR CONSTRAINTS

Constraint	Weight
$u_{l,m}$ = Variable being analyzed (assembled value)	$A_{l,m}$
$v_{l,m}$ = Variable being analyzed (assembled value)	$\hat{A}_{l,m}$
$d_{l,m}$ = divergence = $\partial u/_{\partial x} + \partial v/_{\partial y}$	
$= u_{l+1,m} - u_{l,m} + v_{l,m+1} - v_{l,m}$	$D_{l,m}$
$q_{l,m}$ = vorticity = $\partial v/_{\partial x} - \partial u/_{\partial y}$	$Q_{l,m}$
$= v_{l,m} - v_{l-1,m} - u_{l,m} + u_{l,m-1}$	
$e_{l,m}$ = x-1,y+1 u gradient = $u_{l-1,m+1} - u_{l,m}$	$E_{l,m}$
$\hat{e}_{l,m}$ = x-1,y+1 v gradient = $v_{l-1,m+1} - v_{l,m}$	$\hat{E}_{l,m}$
$f_{l,m}$ = y axis u gradient = $u_{l,m+1} - u_{l,m}$	$F_{l,m}$
$\hat{f}_{l,m}$ = y axis v gradient = $v_{l,m+1} - v_{l,m}$	$\hat{F}_{l,m}$
$g_{l,m}$ = x+1,y+1 u gradient = $u_{l+1,m+1} - u_{l,m}$	$G_{l,m}$
$\hat{g}_{l,m}$ = x+1,y+1 v gradient = $v_{l+1,m+1} - v_{l,m}$	$\hat{G}_{l,m}$
$h_{l,m}$ = x axis u gradient = $u_{l+1,m} - u_{l,m}$	$H_{l,m}$
$\hat{h}_{l,m}$ = x axis v gradient = $v_{l+1,m} - u_{l,m}$	$\hat{H}_{l,m}$

We shall minimize the following integral:

$I \equiv \iint \;[$

(a) $A_{l,m}(u^*_{l,m} - u_{l,m})^2$ [II.1]

(b) $+ \hat{A}_{l,m}\,(v^*_{l,m} - v_{l,m})^2$

(c) $+ D_{l,m}(u^*_{l+1,m} - u^*_{l,m} + v^*_{l,m+1} - v^*_{l,m} - d_{l,m})^2$

(d) $+ Q_{l,m}(v^*_{l,m} - v^*_{l-1,m} - u^*_{l,m} + u^*_{l,m-1} - q_{l,m})^2$

(e) $+ E_{l,m}(u^*_{l-1,m+1} - u^*_{l,m} - e_{l,m})^2$

(f) $+ \hat{E}_{l,m}(v^*_{l-1,m+1} - v^*_{l,m} - \hat{e}_{l,m})^2$

(g) $+ F_{l,m}(u^*_{l,m+1} - u^*_{l,m} - f_{l,m})^2$

(h) $+ \hat{F}_{l,m}(v^*_{l,m+1} - v^*_{l,m} - \hat{f}_{l,m})^2$

(i) $+ G_{l,m}(u^*_{l+1,m+1} - u^*_{l,m} - g_{l,m})^2$

(j) $+ \hat{G}_{l,m}(v^*_{l+1,m+1} - v^*_{l,m} - \hat{g}_{l,m})^2$

(k) $+ H_{l,m}(u^*_{l+1,m} - u^*_{l,m} - h_{l,m})^2$

(l) $+ \hat{H}_{l,m}(v^*_{l+1,m} - v^*_{l,m} - \hat{h}_{l,m})^2$

$]\; dxdy$

The superscript (*) indicates the values we seek. The differential properties of the first guess field are defined in Table II-1, and a depiction of the u component minimization stencil as it relates to the u terms of equation [II.1] is given in Figure II-2.

To minimize the integral we take the first variation with respect to $u^*_{l,m}$ and with respect to $v^*_{l,m}$, yielding the following two equations:

$$\begin{aligned}
\frac{\delta I}{\delta u^*_{l,m}} = \iint [\, & A_{l,m}(u^*_{l,m} - u_{l,m}) \\
& - D_{l,m}\,(u^*_{l+1,m} - u^*_{l,m} + v^*_{l,m+1} - v^*_{l,m} - d_{l,m}) \\
& - Q_{l,m}\,(v^*_{l,m} - v^*_{l-1,m} - u^*_{l,m} + u^*_{l,m-1} - q_{l,m}) \\
& - E_{l,m}\,(u^*_{l-1,m+1} - u^*_{l,m} - e_{l,m}) - F_{l,m}\,(u^*_{l,m+1} - u^*_{l,m} - f_{l,m}) \\
& - G_{l,m}\,(u^*_{l+1,m+1} - u^*_{l,m} - g_{l,m}) - H_{l,m}\,(u^*_{l+1,m} - u^*_{l,m} - h_{l,m})\,]\,dxdy \overset{set}{=} 0
\end{aligned} \qquad [II.2]$$

$$\begin{aligned}
\frac{\delta I}{\delta v^*_{l,m}} = \iint [\, & \hat{A}_{l,m}\,(v^*_{l,m} - v_{l,m}) \\
& - D_{l,m}\,(u^*_{l+1,m} - u^*_{l,m} + v^*_{l,m+1} - v^*_{l,m} - d_{l,m}) \\
& + Q_{l,m}\,(v^*_{l,m} - v^*_{l-1,m} - u^*_{l,m} + u^*_{l,m-1} - q_{l,m}) \\
& - \hat{E}_{l,m}\,(v^*_{l-1,m+1} - v^*_{l,m} - \hat{e}_{l,m}) - \hat{F}_{l,m}\,(v^*_{l,m+1} - V^*_{l,m} - \hat{f}_{l,m}) \\
& - \hat{G}_{l,m}\,(v^*_{l+1,m+1} - v^*_{l,m} - \hat{g}_{l,m}) - \hat{H}_{l,m}\,(v^*_{l+1,m} - v^*_{l,m} - \hat{h}_{l,m})\,]\,dxdy \overset{set}{=} 0
\end{aligned} \qquad [II.3]$$

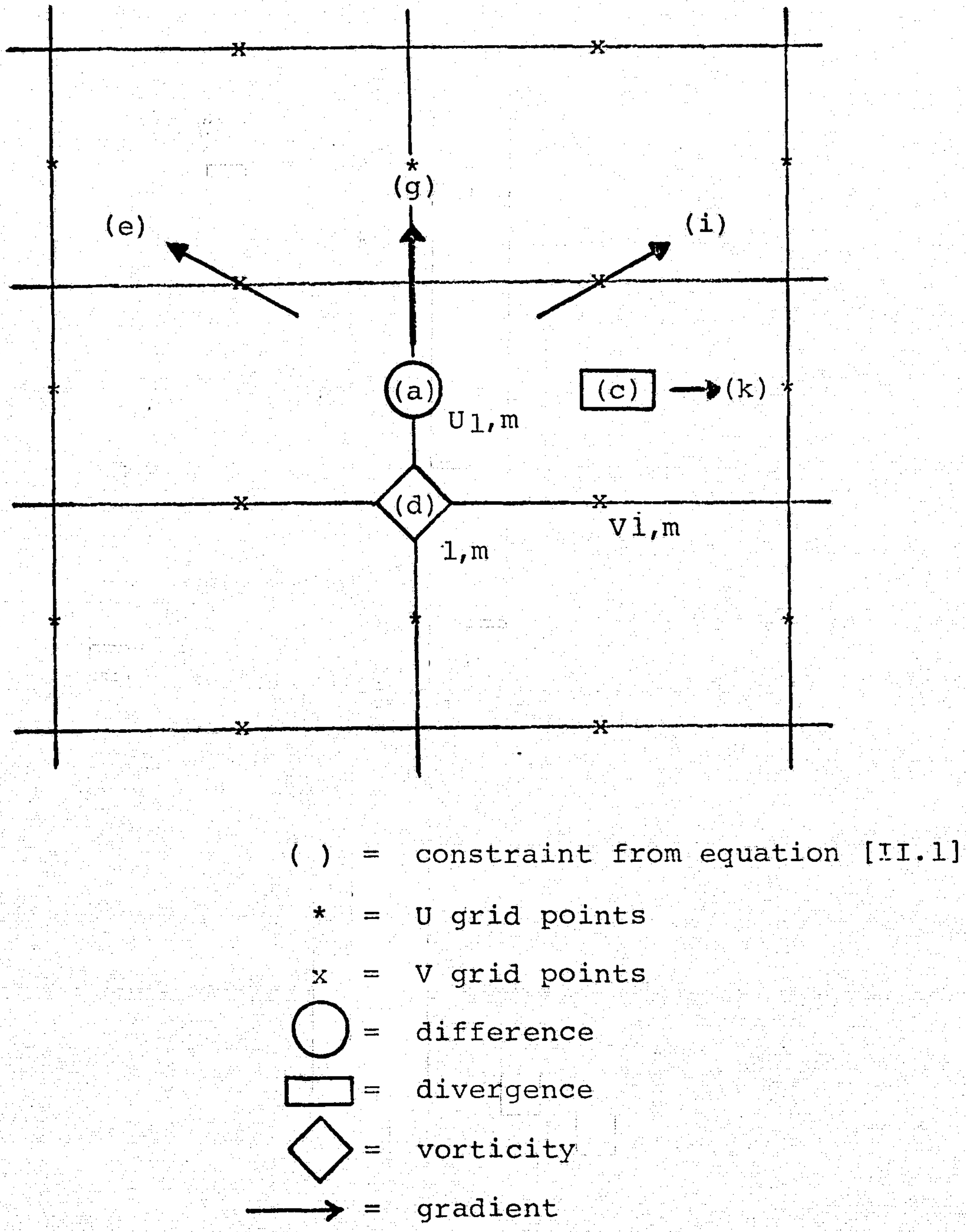


FIGURE II-2: U COMPONENT MINIMIZATION STENCIL

In equation [II.2] group terms involving 1) $u^*_{l,m}$; 2) u^* at surrounding points; 3) v^* and 4) everything else.

$$\overbrace{\iint [(A_{l,m} + D_{l,m} + Q_{l,m} + E_{l,m} + F_{l,m} + G_{l,m} + H_{l,m})}^{S_{l,m}}\ u^*_{l,m} \qquad [II.4]$$

$$X_{l,m}\left\{\begin{aligned}&+ (- D_{l,m} - H_{l,m})\ u^*_{l+1,m} + (-Q_{l,m})\ u^*_{l,m-1}\\ &+ (-E_{l,m})\ u^*_{l-1,m+1} + (-F_{l,m})\ u^*_{l,m+1} + (-G_{l,m})\ u^*_{l+1,m+1}\end{aligned}\right.$$

$$Y_{l,m}\ \{+ (D_{l,m} - Q_{l,m})\ v^*_{l,m} + (-D_{l,m})\ v^*_{l,m+1} + Q_{l,m}\ v^*_{l-1,m}$$

$$Z_{l,m}\left\{\begin{aligned}&- A_{l,m}\ u_{l,m} + D_{l,m}\ d_{l,m} + Q_{l,m}\ q_{l,m} + E_{l,m}\ e_{l,m} + F_{l,m}\ f_{l,m}\\ &+ G_{l,m}\ g_{l,m} + H_{l,m}\ h_{l,m}]\ dxdy = 0\end{aligned}\right.$$

Group [II.3] similarly:

$$\overbrace{\iint [(\hat{A}_{l,m} + D_{l,m} + Q_{l,m} + \hat{E}_{l,m} + \hat{F}_{l,m} + \hat{G}_{l,m} + \hat{H}_{l,m})}^{\hat{S}_{l,m}}\ V^*_{l,m} \qquad [II.5]$$

$$\hat{X}_{l,m}\left\{\begin{aligned}&+ (-\hat{D}_{l,m} - \hat{F}_{l,m})\ V^*_{l,m+1} + (- Q_{l,m}) V^*_{l-1,m} + (-\hat{E}_{l,m})\ V^*_{l-1,m+1}\\ &+ (-\hat{G}_{l,m}) V^*_{l+1,m+1} + (-\hat{H}_{l,m})\ V^*_{l+1,m}\end{aligned}\right.$$

$$\hat{Y}_{l,m}\ \{+ (D_{l,m} - Q_{l,m})\ u^*_{l,m} - D_{l,m}\ u^*_{l+1,m} + Q_{l,m}\ u^*_{l,m-1}$$

$$\hat{Z}_{l,m}\left\{\begin{aligned}&- \hat{A}_{l,m}\ v_{l,m} + D_{l,m}\ d_{l,m} - Q_{l,m}\ q_{l,m} + \hat{E}_{l,m}\hat{e}_{l,m} + \hat{F}_{l,m}\ \hat{f}_{l,m}\\ &+ \hat{G}_{l,m}\ \hat{g}_{l,m} + \hat{H}_{l,m}\ \hat{h}_{l,m}]\ dxdy = 0\end{aligned}\right.$$

Note that all terms in S and Z except $A_{l,m}$ in $S_{l,m}$ and $- A_{l,m}u_{l,m}$ in $Z_{l,n}$ involve first guess information which is constant during the analysis. Similar conditions hold for $\hat{S}$ and $\hat{Z}$.

Equations [II.4] and [II.5] can be written in matrix form:

$$\underline{\underline{S}}_{l,m}\ \underline{u}^* + \underline{\underline{X}}_{l,m} + \underline{\underline{Y}}_{l,m} + \underline{\underline{Z}}_{l,m} = 0 \qquad [II.6]$$

$$\hat{\underline{\underline{S}}}_{l,m}\ \underline{v}^* + \hat{\underline{\underline{X}}}_{l,m} + \hat{\underline{\underline{Y}}}_{l,m} + \hat{\underline{\underline{Z}}}_{l,m} = 0 \qquad [II.7]$$

These equations must be solved simultaneously. The method of solution used is Liebmann over-relaxation. Using a first-guess for u* and v*, equation [II.6] is, in general, not satisfied. A residual is defined by:

$$\underline{\underline{S}}_{l,m}\ \underline{u}^{*\tau} + \underline{\underline{X}}_{l,m} + \underline{\underline{Y}}_{l,m} + \underline{\underline{Z}}_{l,m} = R \qquad [II.8]$$

The superscript τ is the iteration counter. We wish to find a next guess at $\underline{u}^*$ such that the residual is zero, if the values at surrounding points do not change.

$$\underline{\underline{S}}_{l,m}\underline{u}^{*\tau+1} + \underline{\underline{X}}_{l,m} + \underline{\underline{Y}}_{l,m} + \underline{\underline{Z}}_{l,m} = 0 \qquad [II.9]$$

Subtracting [II.9] from [II.8],

$$\underline{\underline{S}}_{l,m}\ (\underline{u}^{*\tau} - \underline{u}^{*\tau+1}) = R$$

and $\underline{u}^{*\tau+1} = \underline{u}^{*\tau} - \dfrac{R}{\underline{\underline{S}}_{l,m}}$. [II-10]

Convergence is more rapid if the correction in [II-10] is exaggerated:

$$\underline{u}^{*\tau+1} = \underline{u}^{*\tau} - \text{ALFA} \frac{R}{\underline{\underline{S}}_{l,m}} . \tag{II-11}$$

The over-relaxation coefficient ALFA must be between 1 and 2.

At a particular grid point, u* is corrected by equation [II.11] and v* is then corrected in an analogous way. In computing R from equation [II.8] or from the analogous equation in v*, the latest estimate of both u* and v* at surrounding points is used. Some of them have been changed on the current iteration and some have not.

During each iteration through the grid, the maximum residual is checked. When it becomes less than a prescribed convergence criterion, the equations are considered solved.

D. Re-evaluating the Data Weights

In the scalar analysis, the resultant weight of the analysis was found by perturbing each guess value and re-solving the minimization equation in the neighborhood of the grid point. The resultant weight was used to re-evaluate the data weights. This procedure presents a considerable computational burden. It accounts for the majority of the computation in the scalar analysis.

In the case of the wind analysis, such a procedure would be considerably more time-comsuming since it would have to be done for the u and v components. Also, agreement with individual wind observations is less crucial than for most scalar observations, since wind reports are so sensitive to local effects, ship motions, elevation effects and many other problems. For these reasons, it is not considered worthwhile to solve for the resultant weights.

The validity of wind reports is judged according to the vector difference between the reported wind and the analyzed wind. The analyzed wind is obtained by interpolation from the analysis fields. The root-mean-square difference is computed and averaged for all the observations that were accepted on the current scan as a diagnostic output. If the report differs in vector magnitude from the analysis by more than the expected difference, its weight is re-evaluated. The expected difference is defined as the square root of the class variance assigned to the report initially, which is the inverse of the original data weight. Define:

$$\lambda^2 = \frac{|\mathbf{V}_n - \mathbf{V}_a|}{\frac{1}{A_o}}$$

where $\mathbf{V}_n$ is the nth report, $\mathbf{V}_a$ is the interpolated analyzed wind, and A_o is the original report weight.

If λ^2 is greater than 1, which implies actual error is greater than expected error, the report weight is computed as:

$$A_n = \frac{2A_o}{1 + \lambda^2} .$$

If λ^2 is less than 1, the report is assigned the weight A_o even if its weight was previously reduced.

E. Program Description - 63 x 63 Grid Version

1. PCTWND

All the arrays have variable dimensions. Common block /DTG/ provides the date-time group, /INFO/ the ident information and /ISW/ the switch settings. Random-access file TAPE9 must be declared on the PROGRAM card of the calling program and is used for temporary storage space within PCTWND. After TAPE9 is opened, the data location lists, the data weight list and the initial weight of the first-guess field are written on it. These arrays are used in the call to subroutine BKGRND, which computes matrices S, ZU and ZV. In the notation of Section II-C, a (^) referred to the v wind component. Since S is the same as $\hat{S}$, no distinction needs to be made. ZU is used in the program for Z and ZV for $\hat{Z}$.

The convention of assigning two names to an array and separating them by a space is used here as it was in PCT. Array AI S holds the I coordinate data location list initially, but returns matrix S from BKGRND. The matrices listed above are written on TAPE9 and their arrays are refilled with their original fields. Weight fields E and F are stored on TAPE9 so they can be used as work arrays later. DO loop 100 is the main loop controlling the number of cycles to be made through the program. At the beginning of each cycle, the original field of first-guess weights is read from TAPE9.

Subroutine ASSMBL adds the data weights to the guess weights, moves the data to grid points and computes the weighted average at each grid point. After calling ASSMBL on the last analysis cycle only, subroutine PLTWIND (see Appendix) writes the data list on the plot file.

Next, the matrices which BKGRND computed are read from TAPE9. Subroutine UPDATE adds A to S, -A*U to ZU and -A*V to ZV. Arrays E and F were used as work arrays in ASSMBL, so their fields are refilled from TAPE9. Subroutine BLEND solves Equations [II.6] and [II.7], resulting in the analysis fields U and V. On the last pass only, U and V are written on the plot file by subroutine PLOT and a printer map is made by PRT. Depending on external sense switch settings the analysis fields may also be written to the disk using the FNWC random access routine ZRANDIO. Finally, the data weights are re-evaluated by REVALWT as described in Section II-D, and a vector root-mean-square difference between the observations that were accepted on that cycle and the resulting analysis of that cycle is computed.

2. BKGRND

Matrices S, ZU and ZV are computed as indicated on page II-9. The comments adequately describe each step.

3. ASSMBL

All data is moved to the nearest grid point and the weighted average of all data and the guess field are computed for each grid point. The assembled weight at each grid point is the guess weight plus the weight of each datum which was moved to that grid point.

Arrays SUMU and SUMV are used to add up the product of weights times the data for each grid point. They are initialized to the guess value times its weight. Then, all the data is scanned and moved to the nearest grid point. Since the grid is staggered (See Figure II-1), the J coordinate of the u component report and the I coordinate of the v component report are decreased by .5. Then, the guess u and v are interpolated to the adjusted report location and if a gross error has not occurred, the difference between the report and the interpolated guess is added to the guess value at the nearest grid point. Because of the staggered grid, a u report may be moved to a grid point which is labeled differently from the point to which the v report is moved.

The assembled weight field is the single array A, but two arrays are needed: one for u and one for v. These two fields are packed into array A.

Gross errors are rejected by setting the last bit of the DATA word. For good data, this bit is cleared.

Finally, the weighted average of u and v are computed for each grid point.

4. UPDATE

Matrix S applies to both the u and v equations, but the terms A*U and A*V have been left out. UPDATE adds them in and two matrices result. These two are packed into array S. Also, A*U is subtracted from ZU and A*V from ZV.

5. BLEND

The two minimization equations, [II.6] and [II.7], are solved as described in detail on page II-10. No further discussion is needed.

6. REVALWT

Data weight re-evaluation is much simpler for wind analysis than for scalar analysis. The explanation in Section II-D is followed closely and the comments in the listing suffice to explain the code.

F. Program Description - 187 x 187 Grid Version

1. Design Philosophy

The vector wind analysis program has been designed to process the 187 x 187 grid by partitions in a fashion similar to that of the scalar case described under Section I-G-1. However, in the wind analysis case, the 16 partitions of an array are adequate for the re-evaluation of data weights. As a result, the subroutines ASSIGN and EXTRCT (described in the Appendix) are not required.

2. PCTWND

Except for partition processing and a different arrangement of storage arrays, the description of PCTWND under Section II-E-1, also describes the 187 x 187 grid version of PCTWND. The partition processing implemented in PCTWND is similar to that implemented in PCT described under Section I-G-2.

Except for blank common, the common descriptions under Section I-G-8 also apply to the vector wind analysis case.

For the vector case blank common has been arranged as follows:

```
COMMON A(50,50), DQ(50,50), U(50,50), V(50,50),
ZU(50,50), ZV(50,50), S(50,50), DATA(63,63),
AI(63,63), AJ(63,63), DWT(63,63)
```

In the 187 x 187 grid case, the matrices E, F, G and H have been reduced to scalars, and these scalar values are set by a DATA statement in PCTWND.

3. BKGRND

The most significant difference between the 187 x 187 grid and the 63 x 63 grid cases is that in the 187 x 187 case BKGRND does not call the subroutine DIVERG. DIVERG is called from the main routine, WINDHEM.

4. ASSMBL

Except for partition processing, the description of ASSMBL under Section II-E-4 also describes the 187 x 187 grid version of ASSMBL.

The current partition number is passed to ASSMBL in the calling sequence, and only those reports having coordinates within the partition are processed. By the time ASSMB has been called for all partitions, all reports will have been processed.

5. UPDATE

This routine is identical to the 63 x 63 version.

6. BLEND

The two minimization equations, [II.6] and [II.7], are solved for each partition with the maximum residual being passed back to PCTWND in common block /BLSTUF/.

7. REVALWT

To implement partition processing, REVALWT calls INDATV (see Appendix) to have a partition's worth of each required data array brought into central memory. This is done for each data report processed.

SECTION III: MASS STRUCTURE LINEAR TRANSFORMATIONS

A. Introduction

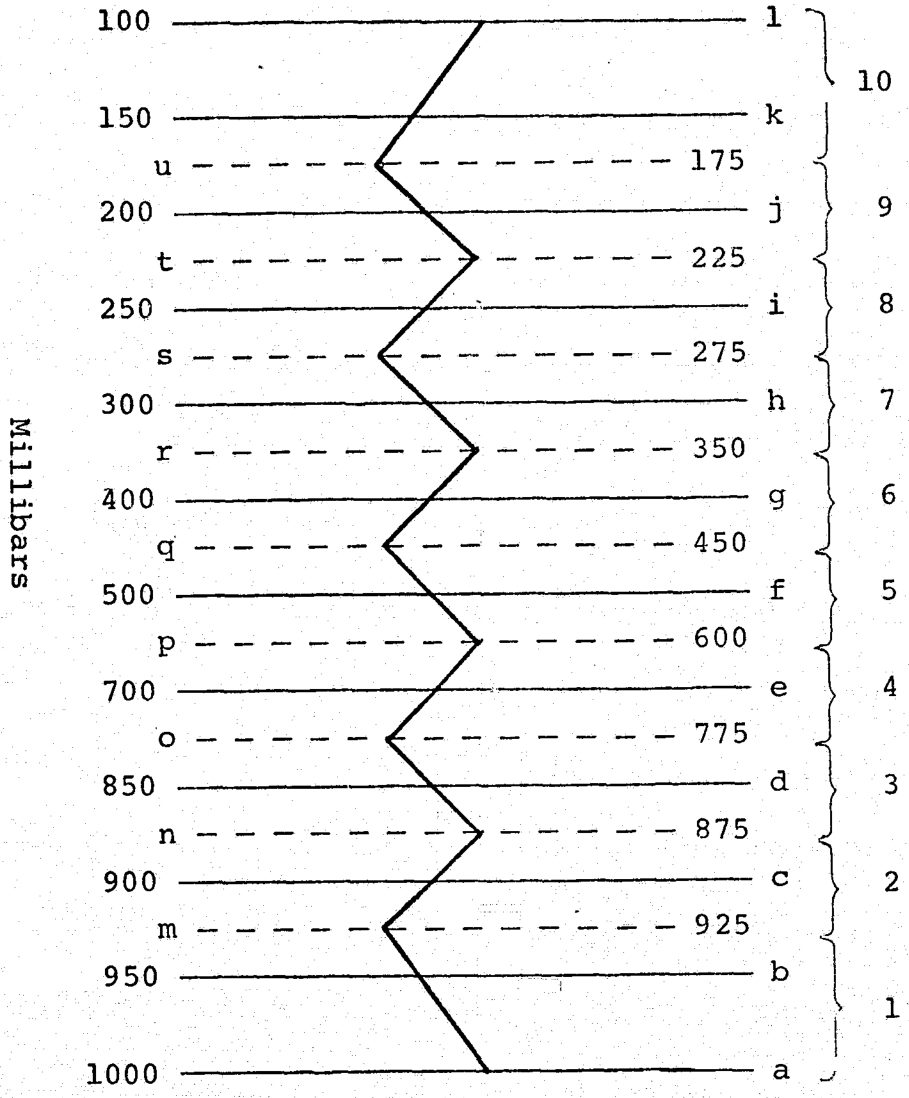
If an analysis of upper-air pressure heights and temperatures is to be used in initializing a forecast model, it is desirable for the heights and temperatures at grid points to be consistent with the hydrostatic equation. It will be shown in Section III-B that the heights, temperatures and layer stabilities can be interrelated through various linear transforms. It turns out that to close the equation set it is necessary to also specify a single height parameter and a single temperature parameter.

The vertical organization of height and temperature levels and stability layers to be used in the mass structure analysis is shown in Figure 4. The stability parameter used here is defined as:

$$\sigma \equiv - \frac{p^2}{\rho} \frac{\partial \ln \theta}{\partial p} \qquad [III.1]$$

Other definitions are possible, as discussed by Holl et al (1963). This definition makes θ linear in $p^{-\kappa}$ ($\kappa \equiv R/Cp$), which is consistent with pseudo-adiabatic diagrams.

A limitation of this technique is that σ must be assumed to be constant in each of the layers labeled 1 - 10 in Figure III-1. If a serious departure from this condition occurs in a layer, the temperature above the layer will depart from the reported temperature, but will agree hydrostatically with the analyzed heights.



(12 levels, 10 layers)

FIGURE III-1: VERTICAL ORGANIZATION OF THE BASIC MASS STRUCTURE MODEL (DOES NOT SHOW THE STRATOSPHERIC EXTENSION).

B. Derivation of the Transformations

The hydrostatic equation states

$$\frac{dz}{dp} = - \frac{1}{g\rho} = - \frac{RT}{pg} \qquad [III.2]$$

or

$$T = - \frac{pg}{R} \frac{dz}{dp}. \qquad [III.3]$$

Potential temperature is defined as

$$\Theta \equiv T \left(\frac{P}{P_o}\right)^{-\kappa} \quad \text{where } \kappa = R/_{Cp} \qquad [III.4]$$

Therefore,

$$\Theta = - \frac{g}{R} \frac{dz}{dp} p^{1-\kappa} (P_o)^{\kappa}. \qquad [III.5]$$

Defining $\eta \equiv 1-\kappa$,

$$\ln \Theta = \ln \frac{g}{R} + \ln \left(-\frac{dz}{dp}\right) + \eta \ln p + \kappa \ln P_o \qquad [III.6]$$

and

$$\frac{d\ln\Theta}{dp} = \frac{d}{dp}\left[\ln\left(-\frac{dz}{dp}\right)\right] + \frac{\eta}{p} \qquad [III.7]$$

and

$$- \frac{dz}{dp} \frac{d\ln\Theta}{dp} = - \frac{d^2z}{dp^2} - \frac{\eta}{p} \frac{dz}{dp} . \qquad [III.8]$$

Substituting for ρ from the hydrostatic equation into [III.1]

$$\sigma = g \frac{dz}{dp} p^2 \frac{d\ln\Theta}{dp} \qquad [III.9]$$

and

$$\frac{d\ln\Theta}{dp} = \frac{\sigma}{g\,\frac{dz}{dp}\,p^2} \,. \qquad [III.10]$$

Substituting [III.10] into [III.8]

$$\frac{d^2z}{dp^2} + \frac{\eta}{p}\frac{dz}{dp} = \frac{\sigma}{gp^2} \,. \qquad [III.11]$$

Multiplying by p^η

$$p^\eta \frac{d^2z}{dp^2} + \eta p^{\eta-1}\frac{dz}{dp} = \frac{\sigma}{g}p^{\eta-2}. \qquad [III.12]$$

Let us define

$$\chi \equiv - p^\eta \frac{dz}{dp} \,. \qquad [III.13]$$

Then

$$\frac{d\chi}{dp} = - \eta p^{\eta-1}\frac{dz}{dp} - p^\eta \frac{d^2z}{dp^2} \,. \qquad [III.14]$$

From [III.12] and [III.14]

$$- \frac{d\chi}{dp} = \frac{\sigma}{g}p^{\eta-2}. \qquad [III.15]$$

Integrating <u>within the layer</u> characterized by constant stability

$$- \chi + C_1 = \frac{\sigma}{g}\frac{1}{\eta-1}p^{\eta-1} + C_2 \,. \qquad [III.16]$$

Defining $M \equiv C_2 - C_1$

$$\chi = \frac{-\sigma}{g(\eta-1)} p^{\eta-1} + M. \qquad [III.17]$$

Integrating [III.13]

$$Z = C_3 - \int \frac{\chi}{p^{\eta}} dp. \qquad [III.18]$$

Substituting [III..17] into [III.18]

$$Z = C_3 + \int [\frac{\sigma}{g(\eta-1)} \quad p^{-1} \quad - Mp^{-\eta}] \; dp \qquad [III.19]$$

or

$$Z = C_3 + \frac{\sigma}{g(\eta-1)} \ln p \quad - M \quad \frac{1}{1-\eta} p^{1-\eta} + C_4 \qquad [III.20]$$

and finally

$$Z = N^* + M^* \, p^{\kappa} - \frac{\sigma}{g\kappa} \ln p \qquad [III.21]$$

where

$$N^* = C_3 + C_4$$

and

$$M^* = -\frac{M}{1-\eta} \, .$$

Equation [III.21] is the basic equation of the method relating pressure height to stability. We also need a relationship between the temperature and the stability. Taking the first derivative of equation [III.21] with respect to pressure,

$$\frac{dz}{dp} = M^* \, \kappa \, p^{\kappa-1} - \frac{\sigma}{g\kappa p}. \qquad [III.22]$$

Substituting [III.22] into [III.3]

$$T = - \frac{pg}{R} (M^* \kappa\ p^{\kappa -1} - \frac{\sigma}{g\kappa p}), \qquad [III.23]$$

or

$$T = \frac{\sigma}{R\kappa} - M^* \frac{g}{C_p} p^{\kappa}. \qquad [III.24]$$

Equation [III.24] is the basic equation of the method relating temperature to stability.

Repeating equation [III.21]

$$Z = N^* + M^* p^{\kappa} - \frac{\sigma}{g\kappa} \ln p. \qquad [III.21]$$

Equations [III.21] and [III.24] are the two model equations we need. They apply to each of the ten layers in Figure III-1. The N*, M* and σ in the ten layers make a total of 30 unknowns. Applying equation [III.21] to each mandatory level gives us twelve equations:

$$\begin{array}{l} N_1 + M_1\ P_a^{\kappa} - \sigma_1 \beta_a = Z_a \\ N_1 + M_1\ P_b^{\kappa} - \sigma_1 \beta_b = Z_b \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ N_{10} + M_{10}\ P_k^{\kappa} - \sigma_{10} \beta_k = Z_k \\ N_{10} + M_{10}\ P_1^{\kappa} - \sigma_{10} \beta_1 = Z_1 \end{array} \qquad [III.26]$$

where

$\beta_\eta \equiv \frac{1}{g\kappa} \ln p_\eta$ and the subscript (*) has been omitted.

Requiring continuity of height at the interface levels between each layer leads to nine more equations:

$$N_1 - N_2 + P_m^{\kappa} (M_1 - M_2) - \beta_m (\sigma_1 - \sigma_2) = 0 \qquad [III.27]$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$N_9 - N_{10} + P_u^{\kappa} (M_9 - M_{10}) - \beta_u (\sigma_9 - \sigma_{10}) = 0$$

Requiring continuity of temperature at the interface levels gives, from equation [III.24], the remaining nine equations:

$$\alpha_m (M_1 - M_2) - (\sigma_1 - \sigma_2) = 0 \qquad [III.28]$$

$$\vdots \qquad\qquad \vdots$$

$$\alpha_u (M_9 - M_{10}) - (\sigma_9 - \sigma_{10}) = 0$$

where

$$\alpha_\eta \equiv g\kappa^2 \, p_\eta^{\kappa}.$$

The 30 equations in 30 unknowns may be written as a single matrix equation.

$$\underline{\underline{B}} \; \underline{C} = \underline{Z}'. \qquad [III.29]$$

The vector $\underline{Z}'$ is composed of the twelve mandatory level heights and 18 zeroes. The vector $\underline{C}$ is the 30-element column vector $\begin{pmatrix}\underline{N}\\ \underline{M}\\ \underline{\sigma}\end{pmatrix}$, where the ten elements of $\underline{N}$, and $\underline{M}$ and $\underline{\sigma}$ correspond to the ten layers. Equation [III.29] is written out in Figure III-2. This can be represented in a partitioned form as

$$\begin{vmatrix} I_1 & I_2 & I_3 \\ H_1 & H_2 & H_3 \\ G_1 & G_2 & G_3 \end{vmatrix} \begin{vmatrix} \hat{N} \\ \hat{M} \\ \hat{\sigma} \end{vmatrix} = \begin{vmatrix} \hat{Z} \\ 0 \\ 0 \end{vmatrix} \qquad [III.30]$$

It can be solved using Gauss elimination, giving

$$\underline{C} = \underline{\underline{B}}^{-1} \underline{Z}'.$$

Which in partitioned form can be represented as

$$\begin{vmatrix} \hat{N} \\ \hat{M} \\ \hat{\sigma} \end{vmatrix} = \begin{vmatrix} F_1 & F_2 & F_3 \\ E_1 & E_2 & E_3 \\ D_1 & D_2 & D_3 \end{vmatrix} \begin{vmatrix} \hat{Z} \\ 0 \\ 0 \end{vmatrix} \qquad [III.31]$$

In the analysis, we need a transformation to get stabilities from heights. That transformation is part of matrix $\underline{\underline{B}}^{-1}$, namely $\underline{\hat{\sigma}} = \underline{\underline{D}}_1 \underline{\hat{Z}}$.

We will also need a transform back to heights. For that problem, our set of 30 equations contains 32 unknowns (10 Ns, 10 Ms and 12 Zs). Two of the unknowns will have to be given in order to close the set.

The obvious choice for one of them is the 1000 mb height, since more data is available at the surface than in the upper air. Choosing the second parameter is more difficult. Since the temperature will be computed from the heights, the second parameter might be chosen as a reference for the temperature profile. We chose the thickness of the 1000 - 300 mb layer.

$$
\left|\begin{array}{cccccccccc|cccccccccc|cccccccccc}
1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{a}&0&0&0&0&0&0&0&0&0&-\beta_{a}&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{b}&0&0&0&0&0&0&0&0&0&-\beta_{b}&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{c}&0&0&0&0&0&0&0&0&0&-\beta_{c}&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{d}&0&0&0&0&0&0&0&0&0&-\beta_{d}&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{e}&0&0&0&0&0&0&0&0&0&-\beta_{e}&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{f}&0&0&0&0&0&0&0&0&0&-\beta_{f}&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{g}&0&0&0&0&0&0&0&0&0&-\beta_{g}&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{h}&0&0&0&0&0&0&0&0&0&-\beta_{h}&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{i}&0&0&0&0&0&0&0&0&0&-\beta_{i}&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{j}&0&0&0&0&0&0&0&0&0&-\beta_{j}&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{k}&0&0&0&0&0&0&0&0&0&-\beta_{k}\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&P^{\kappa}_{l}&0&0&0&0&0&0&0&0&0&-\beta_{l}\\
\hline
1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{m}&-P^{\kappa}_{m}&0&0&0&0&0&0&0&0&-\beta_{m}&\beta_{m}&0&0&0&0&0&0&0&0\\
0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{n}&-P^{\kappa}_{n}&0&0&0&0&0&0&0&0&-\beta_{n}&\beta_{n}&0&0&0&0&0&0&0\\
0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{o}&-P^{\kappa}_{c}&0&0&0&0&0&0&0&0&-\beta_{o}&\beta_{o}&0&0&0&0&0&0\\
0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{p}&-P^{\kappa}_{p}&0&0&0&0&0&0&0&0&-\beta_{p}&\beta_{p}&0&0&0&0&0\\
0&0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{q}&-P^{\kappa}_{q}&0&0&0&0&0&0&0&0&-\beta_{q}&\beta_{q}&0&0&0&0\\
0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{r}&-P^{\kappa}_{r}&0&0&0&0&0&0&0&0&-\beta_{r}&\beta_{r}&0&0&0\\
0&0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{s}&-P^{\kappa}_{s}&0&0&0&0&0&0&0&0&-\beta_{s}&\beta_{s}&0&0\\
0&0&0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{t}&-P^{\kappa}_{t}&0&0&0&0&0&0&0&0&-\beta_{t}&\beta_{t}&0\\
0&0&0&0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0&P^{\kappa}_{u}&-P^{\kappa}_{u}&0&0&0&0&0&0&0&0&-\beta_{u}&\beta_{u}\\
\hline
0&0&0&0&0&0&0&0&0&0&\alpha_{m}&-\alpha_{m}&0&0&0&0&0&0&0&0&-1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&\alpha_{n}&-\alpha_{n}&0&0&0&0&0&0&0&0&-1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{o}&-\alpha_{o}&0&0&0&0&0&0&0&0&-1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{p}&-\alpha_{p}&0&0&0&0&0&0&0&0&-1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{q}&-\alpha_{q}&0&0&0&0&0&0&0&0&-1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{r}&-\alpha_{r}&0&0&0&0&0&0&0&0&-1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{s}&-\alpha_{s}&0&0&0&0&0&0&0&0&-1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{t}&-\alpha_{t}&0&0&0&0&0&0&0&0&-1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\alpha_{u}&-\alpha_{u}&0&0&0&0&0&0&0&0&-1&1
\end{array}\right|
\left|\begin{array}{c}
N_{1}\\ N_{2}\\ N_{3}\\ N_{4}\\ N_{5}\\ N_{6}\\ N_{7}\\ N_{8}\\ N_{9}\\ N_{10}\\
M_{1}\\ M_{2}\\ M_{3}\\ M_{4}\\ M_{5}\\ M_{6}\\ M_{7}\\ M_{8}\\ M_{9}\\ M_{10}\\
\sigma_{1}\\ \sigma_{2}\\ \sigma_{3}\\ \sigma_{4}\\ \sigma_{5}\\ \sigma_{6}\\ \sigma_{7}\\ \sigma_{8}\\ \sigma_{9}\\ \sigma_{10}
\end{array}\right|
=
\left|\begin{array}{c}
Z_{a}\\ Z_{b}\\ Z_{c}\\ Z_{d}\\ Z_{e}\\ Z_{f}\\ Z_{g}\\ Z_{h}\\ Z_{i}\\ Z_{j}\\ Z_{k}\\ Z_{l}\\
0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0
\end{array}\right|
$$

FIGURE III-2: $\underline{\underline{B}}\ \underline{C} = \underline{Z}'$

Define the 12-element column vector $\underline{\Sigma} \equiv \begin{pmatrix} Z_a \\ H \\ \underline{\sigma} \end{pmatrix}$,
where Z_a is the 1000 mb height and H is the 1000 - 300 mb thickness. We need the transformation $\underline{\Sigma} = \underline{\underline{D}}\ \underline{Z}$.

The last ten rows of matrix $\underline{\underline{D}}$ are the first twelve columns of the last ten rows of $\underline{\underline{B}}^{-1}$ (D_1 of [III.31]). The first two rows of $\underline{\underline{D}}$ are

100000000000

-100000010000.

Matrix $\underline{\underline{D}}^{-1}$ may be obtained by Gauss elimination, and the heights can be recovered using $\underline{Z} = \underline{\underline{D}}^{-1}\ \underline{\Sigma}$.

Let us repeat the sequence of operations. At grid points, matrix $\underline{\underline{D}}_1$ is used to convert twelve mandatory level heights to ten layer stabilities. The stabilities are limited to greater than zero and less than a maximum value if necessary. Then matrix $\underline{\underline{D}}^{-1}$ is used to compute the mandatory level heights at the grid points.

The temperatures at the grid points can now be computed by simply substituting σ and M*, which are submatrices of $\underline{\underline{B}}^{-1}$, into [III.24]. In matrix form, $\underline{T} = \underline{\underline{Q}}\ \hat{\underline{Z}}$ where $\underline{T}$ comprises the twelve mandatory level temperatures, $\hat{\underline{Z}}$ the twelve mandatory level heights and $\underline{\underline{Q}}$ the matrix of equation [III.24], namely

$$\underline{\underline{D}}_1/R\kappa - \frac{g}{C_p} p^{\kappa}\ \underline{\underline{E}}_1. \qquad \text{[III.32]}$$

The temperatures and heights at the layer interface levels can be obtained by simply changing the pressures in the matrix coefficients. By proper substitution, the following matrix equation can be formulated:

$$\underline{F} = \underline{\underline{S}}\ \hat{\underline{Z}}$$

where

$$\underline{F} \equiv \begin{pmatrix} \underline{T}_i \\ \underline{Z}_i \end{pmatrix}$$

Vectors $\underline{T}_i$ and $\underline{Z}_i$ are the nine temperatures and heights at the layer interfaces.

SECTION IV: THE SEA-SURFACE TEMPERATURE ANALYSIS

A. Introduction

The Pattern Conservation Technique (PCT) is applied to analyze the sea-surface temperature. For Details of the PCT, see Section I.

Since reports of the sea temperature tend to be less reliable than most atmospheric data, they are given a relatively low initial weight. Actually, the effect of lowering the data weight is achieved by increasing the weights of the guess field shape parameters. The gradients in four directions from each point and the Laplacian of the guess field may have weights as high as 0.1, compared to the initial data weight of 1.0. The shape parameter weights are proportional to the gradients of the terrain field, which are on file TAPE3. Reducing these weights where terrain gradients occur has the effect of inhibiting the spreading of the influence of data across land.

Since the first guess at the sea temperature (the previous analysis) is usually fairly accurate, and since the data quality is not so good, the PCT is very well suited for this analysis.

A brief description of the major program elements not covered in Section I follows.

B. SSTHEM

The arrays necessary to do the analysis are in blank common. Those arrays which pertain to data must be dimensioned at least as big as the number of grid points being used (3,969 for the 63 x 63 grid), because they are used in the PCT routine to hold gridded fields.

Common block /DTG/ holds the date-time group in two forms: one for identifying the data and one for display purposes. Common block /INFO/ contains a title for display and a contour origin and contour interval for plotting.

The guess field is normally the previous analysis, but if sense switch 1 is "on", the guess field is read from a history tape. System routine SSWTCH checks the sense switch. If the sense switch is "on", subroutine READNED searches the history tape for the desired field, which is described by array IPT. READNED is described in the Appendix. If switch 3 is on, the 12 hour old guess field is read using ZRANDIO. If switch 3 is off, the guess field is read from the last analysis output tape (TAPE5) using RDRITE. If switch 4 is on, the output fields are written using ZRANDIO. If switch 5 is on, the output fields are written to TAPE8 using the subroutine PLOT. If switch 6 is on, data lists, analysis statistics and a contour map of the final analysis are produced.

Subroutine READATA or ADPSST extracts the sea temperature.

reports from the data files. Next, bogus cards are read from the input file until an end-of-file is encountered. Each card is one report. Columns 1-10 of each card should contain the bogus Centigrade temperature report in normal floating point format. If the decimal point is punched, the numbers may appear anywhere within the ten columns. If not, columns 9 and 10 will be interpreted as the tenths and hundredths digits. The same rule applies to the location and weight parameters. Columns 11-30 should contain the grid point location of the bogus report. The I coordinate goes in Columns 11-20; the J coordinate in Columns 21-30. Columns 31-40 should hold the weight which the user wishes the bogus data to have initially.

Next, the terrain gradients in four directions from each point are read from TAPE3 and statement function WT computes the weights of the corresponding gradients of the guess field. The weight of the Laplacian is the average of the four gradient weights. The weight of the guess value is .01.

The number of passes to be made through the PCT program is specified by the variable NOPAS. The convergence criterion to be used in the PCT program is CONV. If a report differs from the guess value interpolated to the report location by more than GROS, the report is summarily rejected for that pass. All reports are rechecked at the beginning of each pass.

All that remains is to call the PCT subroutine. See Section I and the comments in the program for the details of the calling arguments.

C. READATA/ADPSST

The routine READATA searches the file containing all types of reports (referred to as the "spot data") for ship reports of sea-surface temperature. The temperatures are put into array DATA; and X and Y grid point coordinates of the report location go into arrays DI and DJ, respectively; the initial data weights go into array DWT; and the total number of reports found is put into the variable NOREP.

The routine ADPSST searches the FNWC disk resident ADPFILE for ship records containing sea surface temperature observations. Knowing the FNWC data format the observations are decoded and put into the arrays described above.

The formats for both methods of accessing data are peculiar to the FNWC data base and further elaboration and explanation are not deemed necessary.

SECTION V. TEMPERATURE AND SEA-LEVEL PRESSURE GUESS

A. Introduction

Two different methods can be used to generate the temperature and sea-level pressure guess fields. The method used is dependent upon whether an analysis/forecast cycle is available or whether one is producing the first analysis in a cycle. If no primitive equation forecasts are available, the first-guess fields for analyzing the sea-level pressure and upper-air temperature fields are computed by advecting the previous analyses with half of the smoothed 500 mb geostrophic wind. Free-slip boundary conditions are used in the advection computation. Program MAKGS generates this type of first-guess field.

It has been found by operational experience that if a 12-hour forecast verifying at the analysis time is available, these produce better first guess fields than advection of the last analysis, especially in the upper air. This method is used after the initial analysis is produced in the analysis/prediction/analysis cycle. Program GSFCST generates this type of first-guess fields.

B. Advective First-Guess - MAKGS

If sense switch 1 has been turned on prior to executing MAKGS, it is assumed that no previous analyses have been made and that file TAPE1 is the history file of the fields to be advected. If switch 2 is on, current date-time-group fields are read from the disc using ZRANDIO and no advection is performed. This option is used to initialize the 12-level analysis from the 10-levels available at FNWC. The two new levels (950 and 900 mb) are obtained from a linear interpolation in ln(p) using the 1000 and 850 mb levels. If switch 3 is on the 12-hour old input fields are read using ZRANDIO. If switch 3 is off the 12-hour old fields are read from the last analysis output tape (TAPE5) using RDRITE. If switch 4 is on the output fields are written to the disc using ZRANDIO. If switch 5 is on the output fields are written to TAPE6 using the subroutine PLOT. If switch 6 is on the output fields are contour mapped.

First, the 12-hour-old 500 mb height field is read and smoothed very heavily by subroutine FILTR. (See Section III-4-C of the report on the Primitive Equation Forecast Model.) The geostrophic wind is then computed from the smoothed field.

The 12-hour-old sea-level pressure field is then read. Subroutine ADVECT advects the old pressure field forward for twelve hours, using one-hour time steps. The resulting guess field for the sea-level pressure analysis is smoothed lightly and output in a manner determined by the sense switches.

Finally, the 12-hour-old upper-air temperature fields are read and advected for twelve hours with the same wind field used before. The resulting guess fields are smoothed and output in a similar manner.

C. ADVECT

Field X is advected for the number of hours specified in the variable ISTEP, using the wind field supplied in arrays U and V. The winds are modified along the boundary by subroutine BCWND (see below). The field being advected is first modified along the boundary to remove any gradient normal to the boundary. Array DX2 is filled with the distance in meters of two grid intervals, taking proper account of the map factor.

The advection is begun with a forward time step, followed by successive centered time steps, until the specified number of one-hour steps has been made. Simple centered space differencing is used in the advection. After each time step, subroutine BC is called to ensure that no gradient normal to the lateral boundaries can build up.

D. BC

The lateral boundary points of field X are made equal to the second interior point, removing any gradient normal to the boundary across the first interior grid row. The corner points are filled by averaging the three surrounding points.

E. BCWND

Fields U and V are changed so there is no wind normal to the boundary on the first interior grid row and no gradient of the component tangent to the boundary in the direction perpendicular to the boundary.

F. Prognostic First-Guess GSFCST

The program GSFCST extracts from the PE forecast output tape (TAPE10) the appropriate 12-hour forecast surface pressure and upper-air temperature fields. If sense switch 4 is on the output fields are written to the disc using ZRANDIO. Is switch 5 is on the output fields are written to TAPE 6 using PLOT. If switch 6 is on the output fields are contour mapped.

SECTION VI. THE HEMISPHERIC SEA-LEVEL PRESSURE ANALYSIS

A. Introduction

The same techniques used in SSTHEM are also used in the sea level pressure analysis (PSHEM). The guess field is prepared by program MAKGS or GSFCST prior to the running of PSHEM.

If switch 3 is on, this guess field and data are read using ZRANDIO; if off, the field is read from TAPE6 using RDRITE. If switch 1 is on, the input data is read using READATA from a "spot data" TAPE1. Similar to SSTHEM, if switch 4 is on the output field is written using ZRANDIO; with switch 5 on the output is written to TAPE8 using PLOT. If switch 6 is on, data·sample and contour maps are printed.

The pattern conservation technique is used to preserve the shape of the guess field according to weight fields A through F. The data is fitted moreclosely in SPHEM than in SSTHEM by lowering the weights of the shape parameters from a maximum of .1 to .01. The terrain gradients in file TAPE3 are used to decrease the weights of the corresponding shape parameters where the terrain is uneven, just as in SSTHEM. In nature, mountain ridges act as obstacles to the migration of surface pressure patterns. An observation on one side of such a mountain ridge should not materially affect the analysis at grid points on the opposite side. The reduction of the weights simulates this effect.

Bogus data is read from data cards in the same way as in SSTHEM.

B. PSHEM

All the necessary arrays are dimensioned in blank common. The arrays for holding information about the data are dimensioned larger than those holding gridded fields because of the large number of surface pressure observations that are available.

Common block /DTG/ holds the date-time group for identification of data (IDT) and for display (JDT). Common block /INFO/ contains a title and a contour origin and interval for plotting. Common block /ISW/ contains the sense switch indicators.

The input field and data are read in a manner depending upon how the sense switches are set. Then the bogus reports are read from data cards until no more cards remain. Each card contains one report. Columns 1-10 hold the reported pressure in mb; 11-20 hold the I coordinate of the report location; 21-30 the J coordinate; 31-40 have the weight to be assigned to the report. All four parameters should be in F10.2 format. If no decimal point is punched, it is assumed to be just left of the second column from the right side of each ten-column field. A punched decimal point overrides this rule.

Statement function WT computes the weights of the shape parameters based on the gradients of the terrain field in four directions from each point, which are read from file TAPE3.

The PCT program is called to perform the analysis. The method of output is determined by the sense switch settings.

C. READATA/ADPPS

Similar to the description in SSTHEM, READATA searches a "spot data" tape for sea level pressure reports from ship and land stations. ADPPS extracts the same type of reports from the appropriate FNWC disc resident operational ADPFILE records. Whichever way the data is accessed, the report value is put into the array DATA, the locations into DI and DJ and the weight into DWT.

The data is assigned a weight value of 1.0 for land reports and 0.7 for ship reports. There is a problem in decoding pressure reports, since the hundreds digit is left off. A report more than 50 cannot be assumed to be less than 1,000 mb. Pressure greater than 1,050 mb does occur; likewise, a report of 49 could be 949 mb. To resolve this problem, the guess pressure field, which is passed to the data extraction routine as an argument, is interpolated to the report location whenever the report is between 40 and 65. If the guess is less than 1,000 mb, then the report is assumed to be less than 1,000 mb, also.

Notice that only reports at the exact synoptic time are accepted because the pressure field can change quite rapidly.

SECTION VII. UPPER-AIR GEOPOTENTIAL HEIGHT AND TEMPERATURE ANALYSIS

A. Introduction

The mass structure analysis procedure will be described in four main parts: the virtual temperature analysis, the computation of the height guess fields, the height-temperature analysis and statospheric field extrapolation.

The virtual temperature analysis uses the PCT and guess fields at each pressure level, prepared by MAKGS or GSFCST (See Section V). Then the sea-level pressure analysis (already done by PSHEM) is hydrostatically converted to a 1,000 mb height field using the 1,000 mb virtual temperature analysis. The analyzed virtual temperatures are then used to compute the other heights.

These height fields are used as the guess fields in the height analysis program. The PCT is again used, but this time, in addition to using the height data from radiosondes, all available wind data (aircraft reports, pibals, radiosonde reports) and satellite wind reports are used to compute geostrophic height gradients. The gradient data replaces the gradients of the guess field at those points where wind data is available. These gradients are given much higher weights than those computed from the guess field.

After the height analysis is made, matrix $\underline{\underline{D}}$ (see page

III-10) is multiplied by the twelve-element column vector $\underline{Z}$ (the twelve mandatory level heights) to obtain the stabilities in each of the ten layers. (See Figure 4.) If the heights imply a hydrostatically unstable layer, the stability of that layer is made slightly positive and matrix $\underline{\underline{D}}^{-1}$ (see Page III-10) is used to compute the corrected heights. Finally, matrix $\underline{\underline{Q}}$ (see page III-10) yields the final virtual temperature analysis.

This technique can be described as a mass structure technique in that the final temperatures and heights are hydrostatically consistent and stable.

There are at least three advantages of this method over the analysis of stabilities as done by FNWC:

1. Incomplete soundings can be used
2. Wind data is used directly
3. Satellite-observed temperature data can be used without a geopotential reference

A potential problem area in this analysis is the matter of a geopotential reference. At present, sea level is the basic reference level. The 1,000 mb height is computed hydrostatically from the sea-level pressure and the 1,000 mb temperature, assuming that the layer between sea level and 1,000 mb has a mean temperature equal to that of the 1,000 mb level. Then the 1,000 mb height is used as the reference level for all the other heights. The advantage is that the many sea-level pressure reports are utilized. The disadvantage is that many of these reports are from high-altitude stations and the sea-level pressures have little meaning. If a great deal of satellite temperature data becomes available, it is worthwhile to consider using an upper level, where satellite-observed temperatures are most reliable, as the reference level.

Finally, 50, 30 and 10 mb height and temperature fields are produced using an empirical extrapolation method which builds upon the 100 mb analyzed height and temperature fields.

B. Temperature Analysis - TMPANAL

The PCT is used to analyze radiosonde virtual temperature observations at the twelve mandatory pressure levels. The guess field has been prepared by program MAKGS or GSFCST. (See Section V)

All the necessary arrays are in blank common and are passed as arguments to the PCT program. Common block /DTG/ holds the date-time group for data identification (IDT) and for display (JDT). Common block /INFO/ contains a display-coded title and a contour origin and contour interval for plotting. Common block /ISW/ contains sense switch setting information.

If sense switch 1 is on, the input data is read from a "spot data" tape. If switch 3 is on, the first guess fields and data are read using ZRANDIO. If switch 3 is off, the input fields are read from TAPE6 using RDRITE. If switch 4 is on, the output fields are written using ZRANDIO; with switch 5 on, the output is written to TAPE6 using PLOT. If switch 6 is on, data samples and contour maps are printed.

The weights of the shape parameters (gradients and Laplacian) are set to a constant value so the terrain does not affect the analysis.

For each of the twelve mandatory levels, the first guess field and data are read in a manner specified by the switch settings and PCT does the analysis. For each level, the INPUT file is checked for bogus data. Bogus reports should be ordered by pressure level, one report per card, with the levels separated by blank cards. Any levels for which no bogus data is provided should be represented by a blank card.

The bogus data should occupy the first ten columns of each card in F10.2 format. The I and J coordinate location and initial data weight go into columns 11 through 40 in F10.2 format.

The temperature analyses are output in a manner specified by the switch settings.

C. READATA/ADPUAT

The data file is searched for radiosonde temperature observations for date-time IDT and level LVL. The source of the input data is determined by switch settings.

If a dewpoint temperature is reported, the virtual temperature is computed by subroutine VIRT. If not, the temperature report is accepted as the virtual temperature. The entire analysis is done in degrees Centigrade.

D. Height First-Guess - HTGUES

The program HTGUES calculates the first-guess height field. If sense switch 3 is on, the sea-level pressure and 1,000 mb temperature fields are read from the disc using ZRANDIO. If switch 3 is off, the input fields are read from TAPE8 using RDRITE. The height of the 1,000 mb surface above sea level is computed by integrating the hydrostatic equation, assuming that the 1,000 mb temperature represents the average temperature for the layer between sea level and 1,000 mb. If switch 4 is on, the resulting 1,000 mb height field is written to the disc using ZRANDIO. If switch 5 is on, the fields are output to TAPE7 using PLOT. Then the other mandatory level temperatures are read and the other heights computed similarly. If switch 6 is on, the resulting output height fields are contour mapped.

E. Height-Temperature Analysis - HTANAL

The heights of the mandatory pressure levels are analyzed with the PCT program, using guess fields computed by HTGUES and radiosonde and satellite retrieved height reports. Wind data from radiosonde reports, pibals, aircraft and satellite derived reports are used to compute height gradients which are treated as data in the PCT program.

After the analysis, the stabilities of the ten layers shown in Figure 4 of Section III are computed using matrix $\underline{\underline{D}}$. (See page III-10.) Any instabilities are removed and the virtual temperatures at the twelve mandatory levels are computed using matrix $\underline{\underline{Q}}$. (See page III-10.)

All the necessary arrays are in blank common and are passed as arguments to the subroutines. Common block /DTG/ holds the date-time group for data identification (IDT) and for display (JDT). Common block /INFO/ has a title, a contour origin and a contour interval for display. Common block /LVL/ contains the number (1-12) of the mandatory level being analyzed and /ISW/ contains the sense switch indicators.

If sense switch 1 is on, the data is read from a "spot data" tape (TAPE1). If switch 3 is on, the input fields and data are read using ZRANDIO. If switch 3 is off, the input fields are read from TAPE7 using PLOT. Switch 4 on prompts the output fields to be written to the disc using

ORIGINAL PAGE IS
OF POOR QUALITY

ZRANDIO. If switch 5 is on, the output height fields are written to TAPE8 and the temperature fields to TAPE4 using PLOT. If switch 6 is on, data samples, analysis statistics and contour maps are produced.

For each of the twelve levels, the guess field and the height data are read in a manner specified by the sense switch settings. Then the input file is checked for bogus data in the same manner as described in Section IV-B. Bogus reports should be ordered by pressure level, one report per card, with the levels separated by blank cards. Any levels for which no bogus data is provided must be represented by a blank card. The bogus height should occupy the first ten columns of each card in F10.2 format. The I and J coordinate location and initial data weight go into columns 11 through 40 in F10.2 format.

The PCT subroutine does the analysis and writes the output in a manner specified by the switch settings.

F. READADATA/ADPUAH

These subroutines are exactly like those described in Section VII-C, except that heights are read instead of temperatures. Reports are given an initial data weight of .0025.

G. RDWNDS

Subroutine BKGRND (see Section I), which is part of the PCT, has been slightly altered to call RDWNDS after computing the gradients of the guess field. RDWNDS calls subroutines AIREP, PIBAL, RASONDE and ADPSAT, which read wind data from aircraft, pilot balloon radiosonde and satellite wind reports, respectively, and compute corresponding height gradients. The gradients of the guess field are in arrays XMU (positive J direction) and XNU (positive I direction) when RDWNDS is called. Subroutines AIREP, PIBAL, RASONDE and ADPSAT put the height gradients into the corresponding arrays YNU and YMU at the points corresponding to the lower left corners of the grid squares within which the wind reports lie. A look at the gradient definitions in Table 1-1 on page I-5 will clarify this point. If more than one report is available for a grid point, the one implying a height gradient closest to the gradient of the guess field is accepted. The weights on the gradient terms (E and C) are increased at points where wind data is used. To determine whether data has been accepted at a grid point, the weight value B at the point is compared to B(1,1). If it is different, then data was used. This assumes that data will not be available at point (1,1), a safe assumption, since data south of the Equator is rejected.

H. AIREP/ADPAIRP

Aircraft-reported wind data are read in a manner specified by the switch settings. Reports within six hours of the time of the analysis are accepted, but their initial weights are reduced in proportion to their age.

The flight level of the reporting aircraft is used to find which mandatory level the report is nearest. The standard heights of the pressure levels are used for this comparison. The report is accepted unchanged at the closest level.

Once the wind direction and speed are unpacked, the u and v components on the polar stereographic grid are computed by subroutine DDFF2UV. The X and Y gradients are computed geostrophically as:

$$\mathrm{XGRAD} = \frac{f}{g} \cdot V \cdot \Delta X$$

$$\mathrm{YGRAD} = \frac{f}{g} \cdot U \cdot \Delta Y$$

Here, f is the Coriolis parameter; g the acceleration of gravity; ΔX and ΔY the grid distances in the X and Y directions. If the gradients differ from those of the guess field by more than a specified amount, the report is rejected.

The value of the weight of the gradient in the Y direction is checked against the value at point (1,1). If they are not equal, then data must have already been utilized at the point in question. In this case, the gradients computed from the previous report (already in YNU and YMU) are compared to the gradients of the guess field (XNU and XMU). The differences are called OLDIFI and OLDIFJ. A similar comparison is made between the gradients just computed from the report being considered and the guess gradients. The differences are called NEWDIFI and NEWDIFJ. If the sum of the new differences is less than that of the old ones, the new report is accepted.

The weights on the gradient data are computed from the age of the report. A report at the time of the analysis gets a weight of .002; a six-hour-old report has a weight of .0015. The guess gradients were assigned a weight of .001.

Array IPLT was passed as an argument from RDWNDS. The u and v components of the wind report and its I and J coordinates are packed into a single word of IPLT.

I. PIBAL/ADPPIB

Winds from pilot balloon observations are read and treated as those from aircraft reports were treated in AIREP with the source being specified by a switch setting. The only significant differences between AIREP and PIBAL, except those resulting from the slightly different data format, are that only reports for the exact time of the analysis are accepted and the gradient weights are all .002.

J. RASONDE/ADPUAW

Winds from radiosonde observations are read and treated exactly as in PIBAL. The only differences are due to the data formats.

K. ADPSAT

Winds derived from satellite cloud vectors are read. The reports include a pressure level estimate and are included in the analysis at the closest analysis level to this estimate. The computed gradients are given a value of .0015.

L. PLTWND

This subroutine is called by RDWNDS with array IPLT as an argument. Array IPLT contains the components and the I,J location of all the wind reports which were used in RDWNDS. A twenty-word identification package is added as detailed in the description of subroutine PLOT in the Appendix and an unformatted record containing IPLT and the identification array is written on the plot file. The fifth word of the identification array is set to 7HWNDATA to identify the record as packed wind data.

M. TEMP12

After the PCT program finishes the analysis on all twelve pressure levels, TEMP12 is called to check for hydrostatic instability and to compute the final temperature analysis from the heights. As discussed on page III-10, the 12 x 12 matrix $\underline{\underline{D}}$, when multiplied by the twelve-element column vector $\underline{Z}$ (the twelve mandatory level heights at a point) yields the stabilities of the ten layers shown in Figure 3-1. If a negative stability is computed, a message is printed and the stability is changed to a slightly positive value. Excessively high stabilities are reduced. Then matrix $\underline{\underline{DINV}}$ (the inverse of $\underline{\underline{D}}$) is multiplied by the twelve element vector consisting of the 1,000 mb height, the 1,000 to 300 mb thickness and the ten stabilities to give the corrected heights. See page III-10 for details.

Finally, matrix $\underline{\underline{Q}}$ is multiplied by the corrected heights to produce the final temperature analyses. The explanation of this transformation is on page III-10.

N. Stratospheric Height-Temperature Extrapolation - STRATO

The method of obtaining stratospheric height and temperature fields is by vertical extrapolation as presented by Lea (1961). The extrapolated values are given by equations of the form

$$Z=A_o+A_1Z_{level-1}+A_2T_{level-1} \quad \text{and}$$

$$T=A_3+A_4Z_{level-1}+A_5T_{level-1}.$$

In this manner the 50 mb fields are extrapolated from the 100 mb height and temperature, 30 mb from the 50 mb and 10 mb from the 30 mb. The coefficients which are a function of latitude and month were obtained from empirical studies of selected rawinsonde stations. The empirical constants are defined in the block data routine STRTCON. After the fields have been produced they are filtered once by a long wave filter.

SECTION VIII. UPPER-AIR WIND ANALYSIS

A. Introduction

The application of the Pattern Conservation Technique (PCT) to wind analysis is discussed in Section II. The discussion in this section is limited to the peculiarities of the upper-air wind analysis program. The technique is used to analyze the winds at the twelve standard pressure levels.

Recall that the PCT wind analysis fits the wind data, while conserving the gradients in four directions, the vorticity and the divergence of the guess field. Each of these differential properties has an associated field of weights. In the upper-air wind analysis, the surface terrain is not used to compute these weights.

Since the analyzed wind fields are used to initialize the Primitive Equation Forecast Model (PEHEM), special attention should be given to controlling the divergence of the final winds. Excessive initial divergence, which is not reflected in the initial temperature and geopotential fields, causes large-amplitude, high-frequency oscillations to develop early in the forecast. This phenomenon is usually referred to as geostrophic adjustment and can seriously degrade the usefulness of the forecast for twelve or more hours, until the mass and motion fields become balanced.

Because the divergence is a very sensitive differential property of the wind field, unrealistically large values will usually result from any analysis scheme unless the divergence is explicitly controlled. The PCT is an effective and versatile method of controlling the divergence.

A common philosophy on this subject is that zero is the best available estimate of the initial divergence. This reasoning led to the use of the balance equation to derive a wind field containing no divergence from the geopotential field. While zero is certainly a better estimate than the large values which arise from a scalar analysis of each wind component, the twelve-hour forecast divergence from a previous run of the PEHEM is likely to have some skill.

The PCT can be used to apply either philosophy. A slight modification is made to the basic PCT wind program to read the wind field whose divergence is to be conserved from a file, instead of computing the divergence from the guess field. If no divergence is desired in the analyzed wind, switch 1 is left off. If the twelve-hour forecast divergence is desired, switch 1 is set and the PE forecast winds are read from TAPE10 using RDRITE.

The first-guess wind field at each of the twelve standard pressure levels is the geostrophic wind computed from the analyzed geopotential field for that level.

B. WINDHEM

The PCT technique is used to analyze the winds at each of the twelve pressure levels. If sense switch 2 is on, the input data is obtained from a "spot data" tape, TAPE1. If switch 3 is on, the height analysis and input data are read from the disc using ZRANDIO. If switch 3 is off, the input fields are read from TAPE8 using PLOT. If switch 4 is on, the output fields are written using ZRANDIO; if switch 5 is on, they are written to TAPE11 using PLOT. If switch 6 is on, partial data lists, analysis statistics and contour maps are produced.

Subroutine WIND computes the geostrophic wind to be used as the first guess. Subroutines AIREP, PIBAL, RASONDE and ADPSAT read wind data from aircraft reports, pilot balloons, radiosondes and satellite wind reports at the proper level indicated by variable LVL, which is in a common block.

Bogus data is read from the INPUT file until a zero data weight is found. Bogus reports for different levels should be separated by a blank card. When the blank card is read, the data weight will be zero and the reading is stopped. Any levels for which no bogus reports are to be inserted must be represented by a blank card. If no bogus reports are desired, twelve blank cards should be in the

INPUT file.

One card is required for each bogus report. The direction from which the wind blows relative to north goes in the first ten columns, the speed in meters per second in the next ten columns, the I and J coordinates of the report position in columns 21 through 40 and the initial data weight in columns 41 through 50. Each ten-column field is read with an F10.2 format.

Subroutine DDFF2UV converts the bogus wind data to u and v components. The components are packed into the single array DATA with u in the left half of each word and v in the right. The entire data list is written on the plot file by subroutine PLTWIND.

Finally, the PCTWND subroutine does the analysis.

C. WIND

The geostrophic wind components are computed from the geopotential height analysis. In the computation of the grid distance and the Coriolis parameter, the latitude is not allowed to be less than twenty-five degrees. The geostrophic wind law is inadequate in tropical latitudes and has a singularity at the Equator.

D. AIREP/ADPAIRP, PIBAL/ADPPIB, RASONDE/ADPUAW, ADPSAT

The first three subroutines are identical to those by the same names described in Sections VII-H, I and J, except that the computation of the geopotential gradients has been deleted and the u and v wind components are packed into array DATA. The locations and initial data weights are put into arrays AI, AJ and DWT.

ADPSAT reads cloud vector derived satellite wind reports if switch 2 is on. These reports contain a pressure level estimate in centibars at which the observation is believed valid, an age, and a direction and speed. The observation is included in the data list of the pressure level nearest the observation. The observation is given a weight of 0.5 in comparison to 0.1 for a radiosonde.

E. BKGRND

In the PCTWND program (see Section II) subroutine BKGRND has been changed to call DIVERG. This subroutine reads the twelve-hour forecast wind field and computes its divergence. In BKGRND, array ZU is used temporarily to hold the computed divergence. Normally, both the divergence and vorticity fields have the weight DQ. We now wish to give the divergence a higher weight. To accomplish this, the divergence weight is packed into the left half of each word in DQ and the vorticity weight into the right half. A minor change was necessary in subroutine BLEND of PCTWND to allow the divergence and vorticity to have different weights.

F. DIVERG

A wind field for level LVL (in common block /LVL/) is read from TAPE10 if switch 1 is on. The output for the initial time is skipped. The u component field is read and the contribution of the gradient of u to the divergence is computed.

Then the v component is read and the contribution of the v gradient is added to the divergence field. The final divergence field is smoothed heavily.

The divergence field is assigned the same weight as wind reports have. This weight field is packed with the vorticity weight field in array DQ. The divergence weights are in the left half of each word and the vorticity weights in the right.

If switch 1 is off, the divergence value is set to zero.

SECTION IX. SUPPORT PROGRAMS

A. CKRAOB

Radiosonde data and satellite retrieved radiosonde profiles are known to occasionally contain errors due to garbling in transmission or errors in the actual workup of the sounding. As mentioned earlier, the upper air analysis is constrained to hydrostatic equilibrium. The hydrostatic equation can be used as a check of whether reports at successive levels in the sounding are reasonable. Due to radiosonde malfunction, levels may be missing in the sounding and the hydrostatic equation can be used to approximate a reasonable value for the missing levels. The mass structure analysis is required to analyze heights and temperatures at the 950 and 900 mb levels. These are not mandatory levels for radiosonde reports so the observation must be interpolated from the profile that is available and written in a format the analysis programs can read.

If sense switch 1 is off, the input data is read using ZRANDIO; if on, the data is read from a "spot data" tape. First, significant level observations are read and duplicates removed. Next mandatory level observations are read and duplicates are removed. Next the mandatory and significant levels of observations reporting both are merged into an array in which pressure monotonically decreases (height increases). Tropopause and maximum wind data are also inserted at the appropriate level if reported.

The next steps are checking for hydrostatic consistency, assigning heights to significant level reports, and approximating missing levels if possible.

First the temperature lapse rate is checked using 3 degrees C/100 meters as a gross check. If the lapse rate exceeds this, the level is flagged as missing and the check continues until the top of the sounding is reached. Next an attempt is made to find the station level report. This is supposed to be the first level reported in a significant level report. Knowing the station elevation, one can calculate the standard station pressure. If the two values differ by greater than 60 mb, this is probably not the station level report. If not, the first reported mandatory level is used for the base height for the hydrostatic workup. Other miscellaneous checks are performed to try to further verify the station level report. The checks are adequately described by comment cards so one can understand this logic. Given a surface pressure, height and temperature, one can calculate, using the hydrostatic equation, the next pressure level height knowing the pressure and temperature at that level. These heights are included in the significant level reports and changed in mandatory levels if the report does not appear to be consistent. Below 250 mb, the average height change required to make the sounding hydrostatically consistent is on the order of 10 meters in a sample size of approximately 450 soundings. Above 250 mb the change runs from about 20 meters to 50 meters. Next, missing temperatures and dewpoint levels are interpolated in ln(p) if possible. Wind reports from both mandatory and significant levels are merged with tropopause and maximum wind reports. Missing levels are interpolated in ln(p) if possible.

Finally, the levels analyzed by the mass structure analysis are extracted and written to the disc in a format compatible with the analysis programs input. The reports are extracted as reported, if possible, or interpolated in ln(p) if no reports are available at the needed level. This applies to heights, temperatures, dewpoint depression, wind direction and wind speed.

B. INITIAL

The purpose of the program INITIAL is to generate the input file for the primitive equation model initialization. The routine BREAD is used to read from the appropriate tape files the needed initialization fields in the correct order. The order is as follows:

Fields	Description	Units
1-13	Heights (1000 - 50 mb)	meters
14-26	Temperatures (1000 - 50 mb)	degrees Celsius
27	Land-sea-ice table	integer
28	Sea level pressure	millibars
29	Terrain height	meters
30	Sea surface temperature	degrees Celsius
31	Albedo	percentage
32-55	u/v wind field pairs (1000 - 100 mb)	meters/second

The fields are written to the output file using BWRITE.

The heights and temperatures are those produced by HTANAL. The land-sea-ice table is an integer field with 1 denoting land, 2 meaning sea and 3 meaning ice. This field was obtained from the FNWC operational system and is representative of the day-time-group of the test data. The sea level pressure was analyzed by PSHEM. The terrain height was obtained by analyzing one degree terrain data obtained from Scripps Institute of Oceanography onto the grid and limiting the normalized terrain gradient to less than 2000 meters per standard mesh length. This terrain is similar to that used by the FNWC operational primitive equation model. The sea surface temperature field was analyzed by SSTHEM. The albedo field was obtained by interpolating climatological fields produced by Posey and Clapp to the polar stereographic grid. These fields are also similar to those used by the FNWC operational model. The wind fields were analyzed by WINDHEM.

C. IDENT

The program IDENT is used to produce inventories of any of the output files generated by the analysis programs that have been written by BUFFER OUT. This includes analyzed fields and data records. The twenty word idents which are part of these field are decoded and pertinent information printed.

D. PLTDWA

This program is executed to produce output that can be processed for display on the Versatec plotter. The input file is TAPE2, which can be any of the output tapes produced by the analysis programs. By decoding the ident of the program determines which type of field it has read and processes it for the plotter in the appropriate manner.

E. CREATE

An option has been included in the analysis programs to allow access to data and initial fields independent of the FNWC operating system, specifically ZRANDIO. This option is exercised by setting IREAD in common block/MS/ equal to one (1) vice zero (0) and having the sense switches set as if to read the data with ZRANDIO.

The program CREATE is executed at the beginning of the program execution sequence (see Section X-B). The input tape (TAPE1) contains in BUFFER OUT format one tape per date-time-group. The tape contains an index record which specifies which data records and fields follow, then these records. An index record specifying the names of the data index records is next, followed by the indices. The index records contain the names of the individual records of data, one index for each specific type of data. The execution of CREATE transfers the input tape to disc in a READMS random access format on TAPE16. When read by the analysis programs, the data is transferred to central memory in exactly the same format as a ZRANDIO read.

SECTION X: PRODUCTION ORGANIZATION

A. Introduction

The execution of the analyses is performed in a sequence of programs executed within a single computer job. Each successive program writes its output to the disk with ZRANDIO or to a TAPE file with BUFFER OUT so the fields are available to succeeding programs. When the sequence is complete, one file has been produced which contains all the analysis output fields and another file has been generated in the format needed to initialize the prediction model (see IX-B).

All the programs that produce upper air fields have an option for producing a 10 or 12 level output. The ten level model does not output the 950 and 900 mb levels. The number of levels is specified by the number 10 or 12 in parenthesis on the execution card. Coding is included to gain access to this number and use it as an indicator within the program.

B. Program Execution Sequence

The actual sequence of programs that are executed depend upon whether it is the first analysis execution in the day-time-group series or whether output from previous analyses or forecasts are available.

In the first analysis in the series the current FNWC analyses are used as the first guess for the normal analysis levels. Since 950 and 900 mb analyses are not produced at FNWC, these levels are interpolated in ln(p) from the available 1000 and 850 mb levels. Since the current analysis is available, it is used as the first guess instead of an inferior twelve hour advected field.

Therefore the program sequence is as follows:

Step	Program	Output
0.	CREATE	Random access data and initial fields (optional).
1.	CKRAOB	Checked radiosonde reports.
2.	SSTHEM	SST anal.
3.	MKGUES	Produce surface pressure and upper air temperature first guess fields from current FNWC fields, interpolate 950 and 900 mb fields.
4.	PSHEM	Analyze surface pressure.
5.	TMPANAL	Analyze 1000 to 100 mb temperature fields.
6.	HTGUES	Generate upper air height analysis first guess from temperature anals (1000 to 100 mb).
7.	HTANAL	Analyze height fields and retrieve consistent temperature fields (1000 - 100 mb). Extrapolate 50, 30 and 10 mb strato heights and temperatures.
8.	WINDHEM	Analyze upper air wind fields (1000 - 100 mb).
9.	INITIAL	Generate forecast initialization file.

Prior to the execution of the sequence the required fields and data must be made available. Upon completion, the output files are copied to tape for later use.

In subsequent analysis cycles, the output from the previous analysis or prognostic fields from a prediction initialized from the previous analysis are available. In these studies twelve hour prognoses are used as first guess fields for surface pressure and upper air temperature analyses. In this second type of execution sequence Step 3 of the above sequence is replaced with:

3.	GSFCST	Read prediction model output tape and write surface pressure and upper air temperature first guess fields.

This implies that for each analysis sequence after the initial day-time-group a twelve hour forecast must be calculated to use as first guess for the next analysis sequence. The first guess for the sea surface temperature analysis is the twelve hour old analysis field. Therefore the output from the last analysis cycle must also be available.

The TAPE numbers and content involved in the analysis sequence are given below:

Tape	Content	Program
TAPE1	NEDN "spot data" (not used) or data/field tape (optional)	All anals read CREATE read (optional)
TAPE2	NEDN field tape (not used)	SSTHEM read
TAPE3	Terrain gradients	SSTHEM read PSHEM read
TAPE4	Retrieved temperature anals and strato temperature extrapolations (temporary)	HTANAL write
TAPE5	Last analysis sequence output	SSTHEM read MKGUES read
TAPE6	Surface pressure and upper air temperature first guess (temporary)	MKGUES write GSFCST write PSHEM read TMPANAL read
TAPE7	Height anal first guess (temporary)	HTGUES write HTANAL read
TAPE8	Analysis sequence output tape	All write INITIAL read
TAPE9	WRITMS/READMS Analysis scratch file	All anals read and write
TAPE10	12 hour old forecast output tape	GSFCST read WINDHEM read
TAPE11	Wind anal output (temporary)	WINDHEM write
TAPE12	Land/sea/ice field	INITIAL read
TAPE13	Terrain height field	INITIAL read
TAPE14	Albedo field	INITIAL read
TAPE15	Prediction model initialization fields	INITIAL write
TAPE16	READMS data/initial fields (optional)	All anals read (optional)
TAPE17	187x187 model READMS/WRITMS partition scratch file	All 187x187 anals

As soon as HTANAL and WINDHEM have finished executing, TAPE4 and TAPE11 are rewound and copied to TAPE8. This results in the following fields and data records being written to TAPE8, the analysis sequence output tape, for the twelve level analysis output.

Type Field	Number Levels	Record Types/Level	Total Number Records
Sea surface temp.	1	3 data/1 field	4
Surface pressure	1	3 data/1 field	4
Upper air temp. anal	12	3 data/1 filed	48
Height anals	12	4 data/1 field	60
Strato height extrapolation	3	1 field	3
Temperature retrievals	12	1 field	12
Strato temp. extrapolation	3	1 field	3
Upper air wind anal	12	3 data/2 fields	60
		Total =	194

C. Production Series

The data sets which were saved from FNWC operational fields and data included eight successive twelve hour date-time groups (DTG). The data sets included appropriate analysis and prediction fields for verification and all possible types of raw data that might be used.

The first DTG will be used to initialize the twelve layer analysis model. This means using the first execution sequence as described in Section X-B. From this analysis cycle a twelve hour forecast will be run. This forecast will be used as the first guess for the next DTG. This analysis should have the influence of the FNWC analysis fields removed and forecast divergence will be available to input to WINDHEM. The execution sequence will be the second type in Section X-B. This DTG will be considered the base date-time-group for the prediction model. Seventy-two hour forecasts of the various prediction models will be run from this DTG. Analyses to use for verification will be produced using the second type of execution sequence, namely running twelve hour forecasts to produce first guess fields for the next analysis DTG. This will be done for the remaining six date-time-groups until an analysis set is produced to verify the seventy-two hour forecast.

APPENDIX

LATLNIJ

The latitude/longitude location of a point is transformed to I and J grid coordinates. Latitudes and Longitudes are input in degrees. The trigonometrics of the problem are pictured in the diagrams below. Triangles ABC and AFE are similar, so

$$\frac{r}{d} = \frac{R}{D}$$

The unknown value is R, a distance to be measured in grid lengths. Angle ϕ is the latitude; R_e is the earth's radius.

$$r = R_e \cos\phi$$

$$d = R_e \ (1+\sin\phi)$$

Since θ is 45°. D = RED, which is the number of grid intervals from the Pole to the Equator. (The standard value of RED for the hemispheric 63 x 63 grid is 31.205.) Therefore,

$$R = \frac{RED \cdot \cos\phi}{1 + \sin\phi} . \qquad [A.1]$$

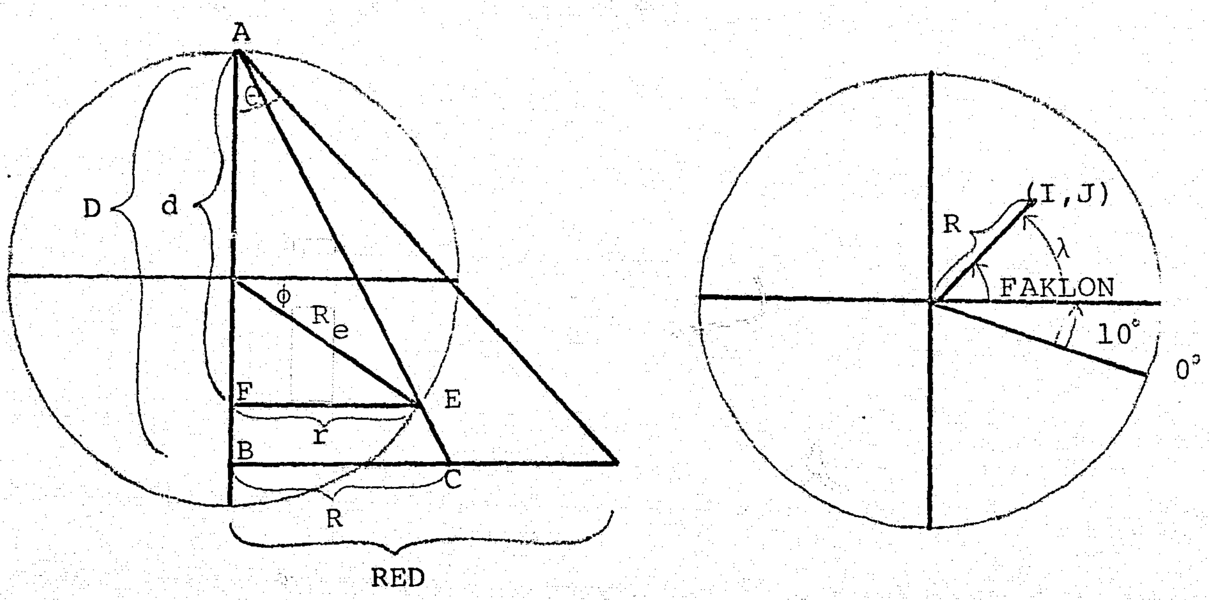
R is the radial distance on the map from the Pole to the point in question, measured in grid lengths. The longitude must be used to calculate I and J.

The longitude is input as a positive angle in the Eastern Hemisphere and negative in the Western. It is converted to a "mathematical style" angle, increasing counterclockwise from the positive I direction by subtracting 10° from it. Designating this angle as FAKLON and R as FACTOR, it is easily seen that

$$I = 32 + \text{FACTOR} \cdot \cos(\text{FAKLON})$$

$$J = 32 + \text{FACTOR} \cdot \sin(\text{FAKLON})$$

for the 63 x 63 grid.



CROSS-SECTION EARTH

POLAR VIEW EARTH

FIGURE A-1:

IJLATLN

The I,J grid coordinate location of a point on the 63 x 63 polar stereographic grid is transformed to latitude and longitude. The latitude can be solved directly from Equation [A.1] in the previous paragraph.

$$\phi = \sin^{-1} \left(\frac{RED^2 - R^2}{RED^2 + R^2}\right)$$

The longitude is easily computed by subtracting the I and J coordinates of the Pole from the coordinates of the point and using the ATAN function. A look at Figure A-1 will clarify this point.

PLOT

All the analysis programs use PLOT to write results to a disk file. A twenty-word identification is added in front of the array, giving the information necessary to direct the plot program to plot the field. Subroutine RDPLT can be used to read the fields written by PLOT and to remove the twenty-word identification group.

The calling arguments include the array to be written (DATA), a four-word title (ITL), a single-word display coded date-time-group for labeling the plot (IDTG), a starting point for computing contours (CO), the interval between contours (CI), a number for computing labels on contours (CL - not presently used), the forecast time in hours, an integer

(ITAU), the unit number of the file to which the field is to be written, an integer (IUNIT), the dimension of the field in the X direction, an integer (M) and the dimension of the field in the Y direction (N). All this information is put into the twenty-word identification group as shown in Table A-1.

PLOT always writes a single unformatted binary record of 3,989 words. Common block /PLOT/ holds the array that is used for this purpose. Subroutine RDPLT also uses this common block. If the array being written out is smaller than 3,969 words, there are some unused words at the end of each record. Since the dimensions of the field are in words seven and eight of the record, the proper field is easily retrieved.

TABLE A-1: TWENTY-WORD IDENTIFICATION GROUP

Word	Content	Word	Content
1	IDTG (Display Code)	11	Title (Display Code)
2	ITAU (Integer)	12	Ttile (Display Code)
3	CO (Real)	13	Title (Display Code)
4	CI (Real)	14	
5	Blank for Fields, 4HDATA for Data 7HWNDATA for wind gradient data	15	
6	NOREP (Integer)	16	
7	M (Integer)	17	
8	N (Integer)	18	
9		19	CL (Real)
10	Title (Display Code)	20	

PLTDAT

The scalar analysis programs use PLTDAT to write data lists to a disk file to be plotted. The same file that holds the analysis fields is used for data. A data list is distinguished from a field by the fifth word of the twenty-word identification group described in the previous paragraph on PLOT and in Table A-1: For a data list, the fifth word is the word DATA in display code. The first record of a data list is 3,989 words, the first twenty of which are the identification group followed by the data list. The number of data words in the record is put into word six of the identification group. The 3,989-word record is followed by two records whose length is that of the data list with no identification group. These two records contain the I and J locations of each datum in the list. There is a serial correspondence among the data list, the I location list and the J location list.

The calling arguments are the array containing the I locations, a real array (AI), the real array containing the J locations (AJ), the data list, a real array (DAT), the scale factor (explained below [SCALE]), the length of the data list, an integer (N), the unit number of the disk file to write on (IUNIT), the I dimension of the field being analyzed, an integer used to interpret the I and J locations

(II) and the J dimension of the field being analyzed (JJ).

Before writing the data list on the file, the real numbers in DAT are scaled, by multiplying them by the real number SCALE, and converted to display code. The plot program is designed to plot only three digits of each datum. SCALE should be specified so that the digits one wishes to plot are in the units, tens and hundreds positions after the multiplication. For example, if a height of 5580. is to be plotted, one would wish to select the first three digits. In this case, SCALE should be 0.1.

PLTWIND

The wind analysis program uses PLTWIND to write the wind data list to a disk file. The arguments are the same as for PLTDAT, except that SCALE is left out. The description of PLTDAT in the previous paragraph applies also to PLTWIND, except the data list is not scaled or converted to display code. Each word of the data list (DAT) contains a u and a v wind component, packed as real numbers. The data list, with the identification group before it, is written on the file in that form.

PLTWND (In Program HTANAL)

Program HTANAL also writes a wind data list on its result file, but in order to avoid using excessive amounts of central memory, the u and v components and the I and J location are converted to integers and packed into a single word. The identification group is added and a single 3,989-word record is written. The fifth word of the identification group is the display-coded word WNDATA.

RDPLT/RDRITE

All the analysis programs use RDPLT or RDRITE to read the results of previously run programs from disk files. In RDPLT the data records are skipped, the identification group is removed and the desired array is returned to the calling program. Common block/PLOT/ is used to read the 3,989-word records. In RDRITE no distinction is made between types of records. One record is read for each call of the routine and the identification group is removed.

ZRANDIO

With the proper switch settings, all programs have the capability of reading fields and data using ZRANDIO which is FNWC's random disk input/output routine. ZREAD, ZWRITE and ADPREAD are general purpose interfaces to ZRANDIO. ZREAD and ADPREAD are used for reading fields and observational data respectively while ZWRITE is used for writing fields. Unique labels are assigned to each record using a formula which combines a three character catalog number (i.e., A01), the

the date-time-group, and the forecast increment. Since the development and production runs of the analysis models were done on FNWC's computers, this option of disk access was used whenever possible due to its greater efficiency and compatibility with the operating system. It would be necessary to implement the READMS option as described in Section IX-E if computers other than FNWC's were used.

INFIELD

This routine reads input fields, partitions them, and either writes them to extended core storage or on a new mass storage file.

The field description is specified in the calling parameters. The field may be read using READNED, RDRITE or ZRANDIO depending upon the sense switch settings and is stored in its appropriate array based upon the value of IFLD.

If ITYPE equals "1" instead of "0", the terrain gradient fields are also read, partitioned and stored in their appropriate locations for use in the sea surface temperature or sea level pressure analyses.

INDAT

The subroutine INDAT retrieves a partition's worth of data from up to nine different 187 x 187 arrays. The 187 x 187 arrays must have already been partitioned and written to either mass storage or extended core storage.

On each call to INDAT, one of a possible sixteen partitions may be specified via a calling sequence argument. This partition number applies to all arrays. Arrays are specified through the word ICODE in labeled common /MEM/ defined under Section I-G-8. The right-most nine octal digits of ICODE are used and arrays 1 through 9 are specified in the order left to right. A "1" digit means retrieve a partition from the array and a "0" digit means don't. For example, ICODE = 111111111B would cause a partition's worth of data from all nine arrays to be retrieved. ICODE = 100100100B would cause a partition's worth of data to be retrieved from arrays 1, 4 and 7.

The subroutine has two entry points, INDAT and INDATV. When INDAT is called, the subroutine assumes that the nine arrays are A, B, C, D, E, F, P, G and S in that order; and when INDATV is called, the subroutine assumes that there are seven arrays in the order A, DQ, U, V, ZU, ZV and S. INDAT is called by scalar programs such as SSTHEM, and INDATV is called by the vector program WINDHEM. For an explanation of the arrays named here, see the corresponding program descriptions and listings.

When retrieving data from the arrays P, U and V, the peripheral points of the partition are updated from points in the surrounding partitions where appropriate. (Peripheral points being those points which overlap surrounding partitions and which would not necessarily contain the latest updated values.)

INDAT has been hard coded to unpack all arrays (see PACK and UNPACK) when processing in extended core storage mode. INDATV has been hard coded to unpack only the arrays U, V, ZU and ZV. The arrays A, DQ and S are not packed.

Internal to INDAT, the table IN(8,16) guides the peripheral points processing. IN contains a list of eight partition numbers associated with each partition. For example, Partition 10 has the list 9, 11, 6, 14, 7, 5, 15 and 13 which are the numbers of the partitions peripheral to partition 10, as illustrated here:

13	14	15
9	10	11
5	6	7

Note that the list is in the order left side, right side, bottom side, top side, lower-right corner, lower-left corner, upper-right corner and upper-left corner. Another example is Partition 8, which has the list 7, 0,4 12, 0, 3, 0 and 11. The zero entries indicate that no peripheral partitions exist on the right-side, the lower-right corner and the upper-right corner of Partition 8, as illustrated here:

11	12	0
7	8	0
3	4	0

In extended core storage mode and in the scalar case, INDAT computes the storage address using the formula:

IADR = 2700*(N-1) + 17*(IP-1)

where N is an array number (1-9) and IP is a partition number (1-16).

In the vector case INDAT makes use of the internal tables JADR and JFACT (see listing of INDAT), so that the formula becomes:

```
IADR = JADR(N) + JFACT(N)*(IP-1)
```

where N and IP are as previously defined.

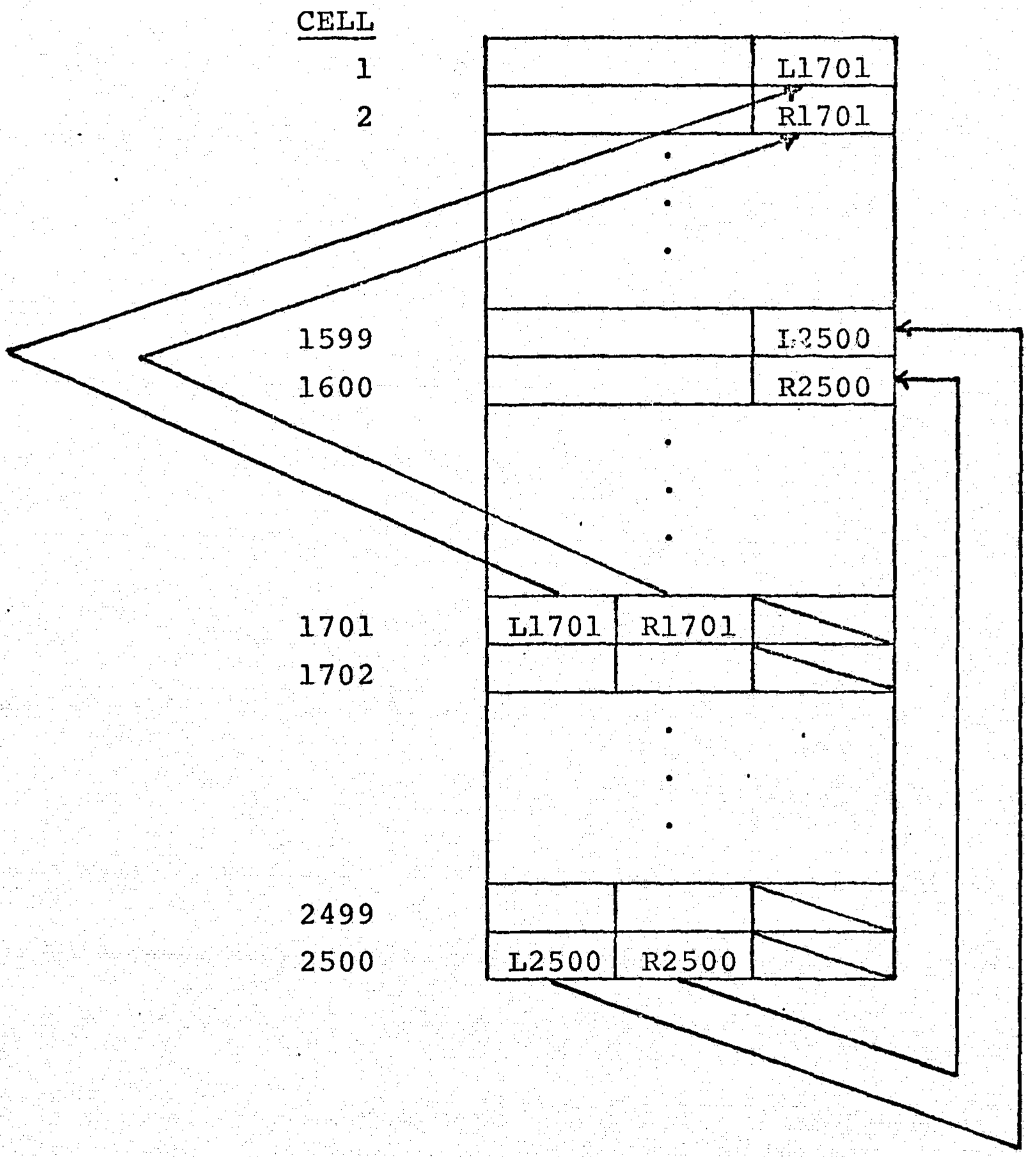
In mass storage mode the record number is computed using the formula:

```
IREC = 16*(N-1) + IP
```

where, again, N and IP are as defined above. Prior to calling INDAT for the first time, the random file JF (a variable in labeled common /MEM/) must have been opened. This file uses the numerical index option. (See the description of READMS in the FORTRAN manual.)

PACK and UNPACK

The subroutine PACK packs an array of 2500 real formatted numbers into the first 1700 cells of the same array, as illustrated here:



L and R indicate the left 20 bits and the right 20 bits, respectively, of the 40 most significant bits of a real formatted word. The 20 least significant bits of words 1 - 1600 and words 1701 - 2500 are lost.

The subroutine UNPACK unpacks the 1700 cells into the 2500 cells using a reverse procedure.

C18750

The subroutine C18750 divides a 187 x 187 array into the 16 partitions defined under Section I.G.1.

When the flag IOPT in labeled common /OPTION/ is set to "0", C18750 transposes the 187 x 187 array before partitioning.

C18750 writes partitions to mass storage or to ECS, depending on the setting of the variable MEMTYP in the labeled common /MEM/. (Labeled commons are described under Section I.G.8.)

On each call to C18750, just one 187 x 187 array is processed. The array is specified via an argument in the calling sequence. When the argument is set to a negative value, the partitions are not packed, when processing under ECS mode. Formulas for computing ECS addresses and record numbers are similar to those used by INDAT.

C50187

The subroutine C50187 builds a 187 x 187 array from the 16 partitions of the array.

C50187 reads partitions from mass storage or from ECS depending on the setting of the variable MEMTYP.

On each call to C50187 just one 187 x 187 array is built. The array is specified via an argument in the calling sequence.

C50187 always transposes the 187 x 187 array once the array has been built.

Formulas for computing ECS addresses and record numbers are similar to those used by INDAT and C18750.

TRANSPOS

The subroutine TRANSPOS transposes an array. It is called by the subroutines C18750 and C50187 to have 187 x 187 arrays transposed.

OUTDAT

The subroutine OUTDAT stores a partition's worth of data from up to nine different 187 x 187 arrays. Partitions are stored either on mass storage or in extended core storage depending on the setting of the variable MEMTYP.

OUTDAT uses the word JCODE in /MEM/ in a fashion similar to the way that INDAT uses ICODE.

On each call to OUTDAT, one of a possible sixteen partitions may be specified via a calling sequence argument. This partition number applies to all arrays.

The subroutine has two entry points, OUTDAT and OUTDATV. When OUTDAT is called, the subroutine assumes that the nine arrays are A, B, C, D, E, F, P, G and S in that order; and when INDATV is called, the subroutine assumes that there are seven arrays in the order A, DQ, U, V, ZU, ZV and S. Formulas for ECS addresses and record numbers are similar to those used by INDAT.

LINEAR and CUBIC

The subroutine LINEAR performs a two-dimensional linear interpolation of points contained in a 63 x 63 grid.

The subroutine CUBIC calls routines to generate a two-dimensional interpolation of points contained in a 63 x 63 grid using a cubic spline technique.

The routines LINEAR and CUBIC are used to generate 187 x 187 arrays from 63 x 63 arrays.

ASSIGN

The subroutine ASSIGN assigns a partition number to each data report. Partition numbers assigned are in the range 1 - 49, where the partitions numbered 17 - 49 are composites of those numbered 1 - 16.

A 187 x 187 array is first partitioned as follows:

187

187

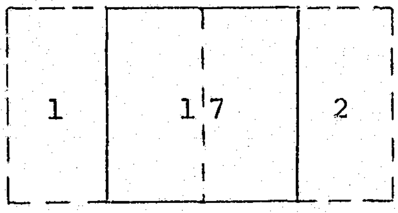
13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

Then additional partitions are roughly positioned as follows:

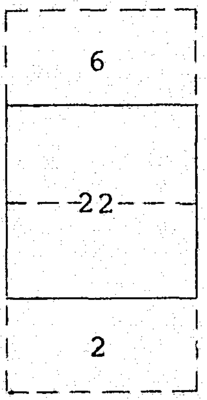
187

	13	47	14	48	15	49	16
	40	41	42	43	44	45	46
	9	37	10	38	11	39	12
187	30	31	32	33	34	35	36
	5	27	6	28	7	29	8
	20	21	22	23	24	25	26
	1	17	2	18	3	19	4

For example, Partition 17 overlaps Partitions 1 and 2 as follows:



and Partition 22 overlaps Partitions 2 and 6 as follows:



(For actual partition limits, see routine listings).

ASSIGN stores a partition number into the array KPART which has an entry for each report. It also stores the I,J coordinates of the report point relative to the partition. The KPART entry format is

20 bits	20 bits	20 bits
I	J	IP

where I and J are the report point coordinates and IP is the partition number.

As ASSIGN builds the KPART array it maintains a sum of reports to which each partition is assigned and stores these sums in the array IWHAT, which has an entry for each of 49 possible partitions. The arrays KPART and IWHAT are contained in the labeled common /REVAL/ and are used by the subroutine REVALWT to retrieve partitions for data weight re-evaluation.

The subroutine ASSIGN prints the contents of the IWHAT array just prior to returning to the calling program.

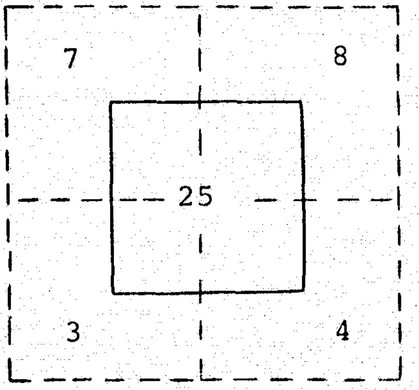
EXTRCT

The subroutine EXTRCT retrieves a partition's worth of data from the arrays A, B, C, D, E, F, P, G, S and OLDP to be used by the REVALWT routine to re-evaluate data weights.

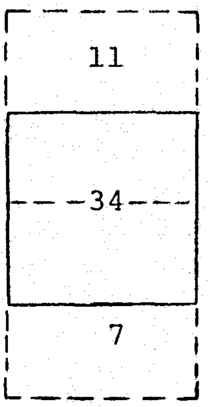
The required partition is specified via a calling sequence argument. For partitions numbered 1-16, EXTRCT retrieves a single partition (per array) corresponding to the number specified. For partitions numbered 17-49, EXTRCT retrieves all partitions necessary for the construction of the required composites.

For all arrays except OLDP, EXTRCT retrieves partitions from either mass storage or extended core storage depending on the setting of the variable MEMTYP. EXTRCT retrieves OLDP partitions from the mass storage scratch file, TAPE9.

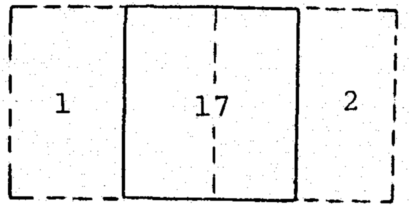
The internal table IN(4,33) is used by EXTRCT to guide the construction of composites. Each partition having a partition number in the range 17-49 has an entry in the IN table. An entry consists of four partition numbers. For example, the entry for Partition 25 has the numbers 3, 4, 7 and 8, which means that Partition 25 is a composite of partitions 3, 4, 7 and 8 as follows:



Another example is Partition 34 having the numbers 0, 0, 7 and 11, which means that Partition 34 is a composite of Partitions 7 and 11, as follows:



A final example is Partition 17 having the numbers 1, 2, 0, 0, meaning that Partition 17 is a composite of Partitions 1 and 2, as follows:



These three examples illustrate three composite cases. A Case I composite is generated from quarters of four other partitions; a Case II composite is generated from the horizontal halves of two other partitions; and a Case III composite is generated from the vertical halves of two other partitions.

Formulas for ECS addresses and record numbers are similar to those used by INDAT.

REFERENCES

HOLL, M.M., et al, 1963: Linear Transforms for State-Parameter Structure, U.S. Navy Fleet Numerical Weather Central report on Contract N228(62271)58264, Monterey, Ca, 29 pp.

HOLL, M.M. and B.R. Mendenhall, 1972: Fields by Information Blending, Sea-Level Pressure Version, U.S. Navy Fleet Numerical Weather Technical Note No. 72-2. Monterey, Ca, 66 pp.

LEA, D.A., 1961: Regression Equations for Vertical Extrapolation of Constant Pressure Surface Heights and Temperatures Between 200 mb and 30 mb, 191st meeting of the AMS, Chicago, Ill., March 1961.

POSEY, J.W. and P.F. Clapp: Global Distribution of Normal Surface Albedo, Geofisica Internacional, (date unknown).

SASAKI, Y., 1970: Some Basic Formalisms in Numerical Variational Analysis, Monthly Weather Review, 98, 12, pp. 875-883.

SASAKI, Y., 1970: Numerical Variational Analysis Formulated Under the Constraints as Determined by Longwave Equations and a Low-Pass Filter, Monthly Weather Review, 98, 12, pp. 884-898.

WEIGLE, W.F., et al, 1973: Development of FIB/UV-MED, U.S. Navy Environmental Prediction Research Facility report on Contract N66314-73-C-1444, Monterey, Ca, 38 pp.