# OBJECTIVE NEPHOLOGY



Alan M. Gerlach, ed.

ST Systems Corporation
109 Massachusetts Avenue
Lexington, MA 02173

15 April 1988

FINAL REPORT
13 February 1987 - 12 March 1988

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

This technical report has been reviewed and is approved for publication.

ALLAN J. BUSSEY
Contract Manager

FOR THE COMMANDER

ROBERT A. McCLATCHEY, Director
Atmospheric Sciences Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify AFGL/DAA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution unlimited |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>AFGL-TR-88-0109 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>ST Systems Corporation (STX)<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>Air Force Geophysics Laboratory |
| 6c. ADDRESS (City, State and ZIP Code)<br>109 Massachusetts Avenue<br>Lexington, MA 02173 | 7b. ADDRESS (City, State and ZIP Code)<br>Hanscom AFB<br>MA 01731 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>Air Force Geophysics Laboratory<br>8b. OFFICE SYMBOL (If applicable)<br>LY | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>Contract No. F19628-87-C-0046 |
| 8c. ADDRESS (City, State and ZIP Code)<br>Hanscom AFB<br>MA 01731 | 10. SOURCE OF FUNDING NOS.<br>PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2310; TASK NO G7; WORK UNIT NO CA |
| 11. TITLE (Include Security Classification)<br>Objective Nephology | |
| 12. PERSONAL AUTHOR(S)<br>Gerlach, Alan M. (ed.) | |
| 13a. TYPE OF REPORT<br>FINAL<br>13b. TIME COVERED<br>FROM 87/2/13 TO 88/3/12 | 14. DATE OF REPORT (Yr., Mo., Day)<br>88/4/15<br>15. PAGE COUNT<br>152 |
| 16. SUPPLEMENTARY NOTATION | |

| 17 COSATI CODES | | |
|---|---|---|
| FIELD | GROUP | SUB. GR |
| 0401 | | |
| 0402 | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Limited area objective analysis; Global cloud climatology; Satellite cloud analysis; Boundary layer diffusion; Radar cloud/precipitation mapping; Cloud truth database

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Reports on meteorological research on a relocatable limited-area objective forecast model, global cloud characterization using digital satellite data, morphology and motions of cloud/precipitation systems using radar data, automated global cloud climatology, and atmospheric diffusion models.

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐ | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Allan J. Bussey | 22b. TELEPHONE NUMBER (Include Area Code)<br>(617)377-2977<br>22c. OFFICE SYMBOL<br>LY |

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

# FOREWORD

This is the Final Report under Contract F19628-87-C-0046 with the Atmospheric Sciences Division, Air Force Geophysics Laboratory. The contract work performance period extended from February 13, 1987 to March 12, 1988.

This report details the results and status of research performed by ST Systems Corporation (STX) during the contract period.

The contract called for effort in five discrete technical areas, reported upon in the monthly R&D Status Reports under these headings:

Improved Regional Cloud Forecast Model
Global Cloud Characterization from Digital Satellite Data
Cloud/Precipitation Systems: Morphology and Motions
Automated Global Cloud Climatology
Atmospheric Transport and Diffusion

Personnel associated with these areas at any time during the contract, in alphabetical order, were:

Improved Regional Cloud Forecast Model -
Isidore M Halberstam, Ph.D.
Chris Johnson, B.S.
Shu-Lin Tung, M.S.

Global Cloud Characterization from Digital Satellite Data -
Gary B. Gustafson, B.S.
Charles F. Ivaldi, Jr., B.S.
Barry Mareiro

D. Keith Roberts
Ronald F. Wachtmann, M.S.

Cloud/Precipitation Systems: Morphology and Motions -
Alberto Bianco, B.A.
Paul R. Desrochers, M.S.
Ralph J. Donaldson, Jr., S.M.
F. Ian Harris, Ph.D.

Automated Global Cloud Climatology -
Albert R. Boehm, M.S.
James H. Willand

Atmospheric Transport and Diffusion -
Joan M. Ward, A.B.

Principal Investigator was Alan M. Gerlach, Ph.D.

Reports were prepared by the scientists, engineers, and mathematicians identified in the Table of Contents.


| Accession For | |
|---|---|
| NTIS GRA&I | ☑ |
| DTIC TAB | ☐ |
| Unannounced | ☐ |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

TABLE OF CONTENTS

# I. IMPROVED REGIONAL CLOUD FORECAST MODEL

## Application and Expansion of the Relocatable Limited-Area Model (RLAM)

### 1. Introduction

The relocatable limited-area model (RLAM) developed by STX for the Air Force Geophysics Laboratory (AFGL) has been discussed in Gerlach (1986, 1985, and 1984) and in the open literature by Tung et al. (1987) and Halberstam et al. (1988).

Since those reports were published, RLAM has undergone some modification and has been employed in studies to determine its ability to forecast a storm system off the East Coast of the US. Physical parameterization has remained the same in RLAM with a Kuo convection scheme, dry convective adjustment, and bulk boundary layer fluxes borrowed from Mathur's (1983) model. Indeed, Mathur's model, known to the Air Force Global Weather Central (AFGWC) as the regional window model (RWM), has also been employed as a backdrop for RLAM. Input and output from RLAM and RWM have been made compatible so that similar synoptic situations can be tested with both models. Because both share the same physics, differences in the forecasts stem only from numerical considerations, including smoothing and damping terms.

Much effort has also been expended in incorporating a semi-implicit scheme. This temporal scheme is designed to be an efficient formulation allowing large time steps and hence quicker computer turn around. Unfortunately, it also requires careful reformulation of the equations, separating linear from non-linear terms. The linear portion allows for much larger time steps without causing instability while the non-linear terms serve as an adjustment to the linear

segment. The semi-implicit scheme will be detailed below, as will the experiments involving RLAM and RWM.

## 2. Review of RLAM

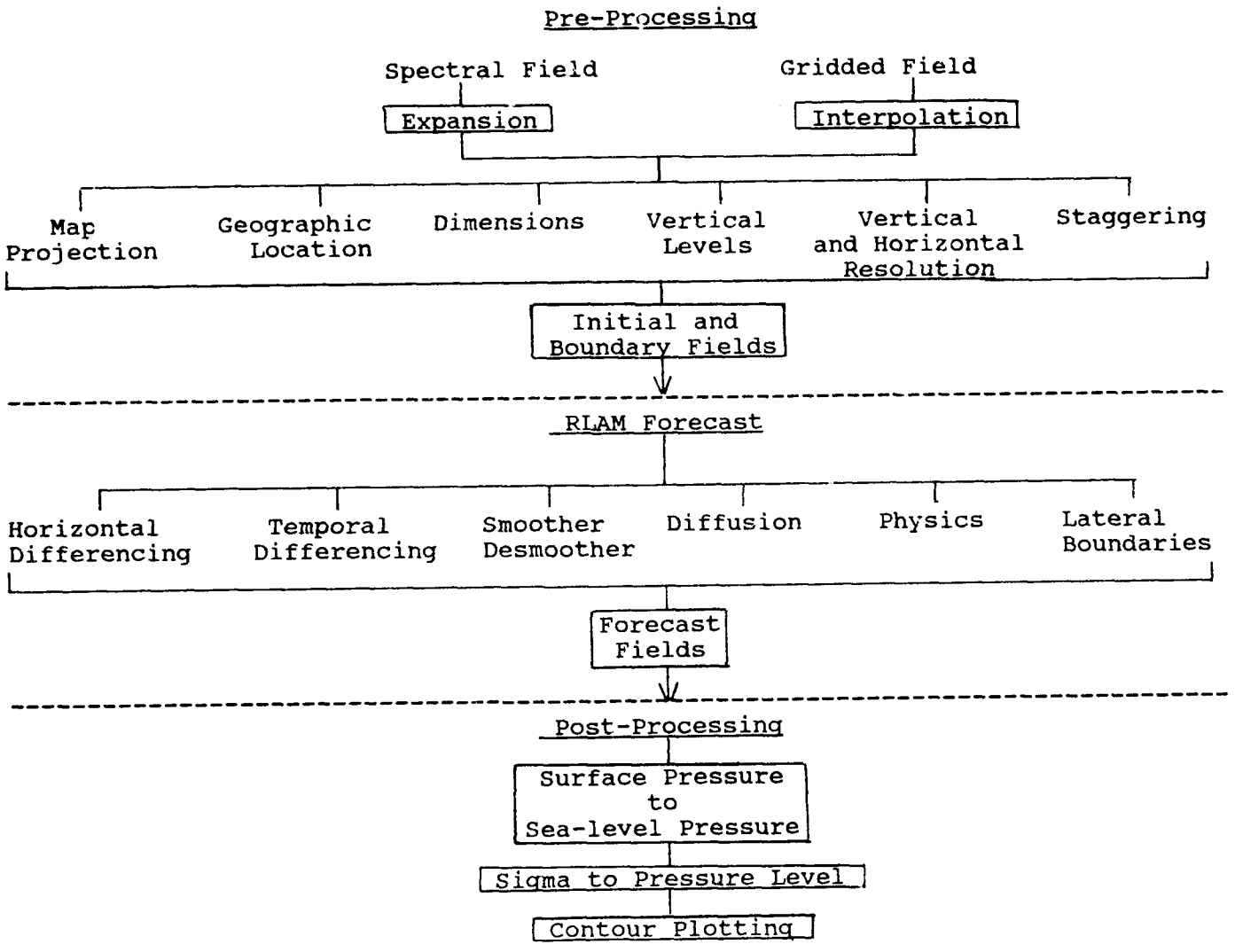
The features and selections of RLAM are summarized in the flow chart in Table 1. The diagram does not represent exactly the current state of the model but an intended goal. The current status is discussed in this section.

### a. Pre-processing

The pre-processor is geared to accepting either gridded or spectral data and converting them to gridded data for RLAM. Spectral fields are expanded at grid points by use of polynomials in the sine of latitude as discussed in Gerlach (1985). So far, only wavenumbers up to rhomboidal 30 have been expanded for RLAM purposes. However, tests of the method have shown that for higher wavenumbers, extra precision is necessary even for the 60-bit words on the AFGL CDC system.

Merilees (1973) also proposed a substitute for the Legendre functions in expanding spectral coefficients at specific latitudes and longitudes. Instead of polynomials in sin $\phi$, where $\phi$ is latitude, he suggested expanding about cos j $(\phi + \frac{\pi}{2})$, where j is the wavenumber. One of the advantages of using the cosine function is that analytical expressions are available that will generate the Legendre functions in terms of the cosine functions. It is then a straightforward procedure to transform an expansion in terms of Legendre functions into an expansion of cosine functions. Accuracy of the expansion is also more easily controlled in this fashion because the analytical generator can be tuned to as high a precision as possible on any specific computer, while the precision of cosines of variables is more readily

TABLE 1. FLOW CHART FOR RLAM



preserved than powers of sines. Tests were conducted with the cosine functions with considerable success.

Other selections in the pre-processor have not changed much from their initial formulation. Geographic location is chosen by entering the desired latitude and longitude of the center of the field. The grid system is then built around the central point. The map projection selection still allows for one of four choices: polar stereographic, Lambert, Mercator, or latitude-longitude. Only polar stereographic is allowed poleward of $85^{o}$, but no other restrictions on mappings have yet been implemented. The dimensions and size of the domain are also optional and are selected by choosing the number of grid points in either direction and the fraction of the standard mesh size (380 km) desired. Most of the test cases examined have had 39 x 33 grid points. Increasing the number of points leads to space problems in the current AFGL system. The staggering of grids is also chosen in the pre-processing in order to determine where the primary variables should lie. It is also necessary to determine the map factors, coriolis coefficient, and other fixed spatially dependent variables on the different, staggered grids. It would have been logical for the vertical structure to be determined in the pre-processor as well, as depicted in Table 1. To date, RLAM has no provision for vertical interpolation of variables and this option does not exist. All tests and experiments were conducted with the 12-level configuration of the AFGL global, spectral model (GSM).

b. Boundary Values

The boundary conditions are determined from cubic interpolation from various times selected by the user. Boundary values are generated from the GSM forecasts (or any other chosen field) for the full period of the intended RLAM

forecasts, and coefficients for a cubic polynomial in time t are generated from the prescribed values. The time spacing between the boundary fields is arbitrary, but the practical choice for all experiments and tests has been six hours. The prescribed valid time for any group of coefficients will depend on where (in time) it appears in the forecast run. Coefficients at the beginning and end of the forecast period are valid for longer time increments because no data exist before the initial time nor after the final forecast time. For in-between times, the coefficients will change after each selected time increment (e.g., six hours). Thus, for a 24-hour forecast, where boundary values are provided every six hours from 0 to 24 hours, the initial set of coefficients will be valid for 12 h and will be derived from values at 0, 6, 12, and 18 h. Between 12 and 24 h the coefficients will be derived from values at 6, 12, 18, and 24 h. If this were a 48 h forecast, the second set of coefficients would be valid only until 18 h, at which point a new set of coefficients from 12, 18, 24, and 30 h would operate between the central 18 to 24 h segment. Because of this change in boundary conditions, a 48 h forecast should not be regarded as a simple extension of a 24 h forecast, regardless of whether one began a 48 h forecast and stopped after 24 h and then began a second 24 h forecast, or one conducted a simple 24 h forecast and then continued with a second 24 h forecast. The extent of departure of these forecasts from a straightforward 48 h forecast depends on the method of incorporating the boundary data into the model, but in any case significant differences should be anticipated because of the sensitivity of limited-area models to boundary conditions, especially in small regions.

The number of boundary rows is also arbitrary, but choosing seven rows guarantees that there will be sufficient data for most boundary schemes. Thus, most boundary data files for RLAM contain coefficients for seven boundary rows

for every variable at every level. The actual values at the boundary for any given time are calculated by expanding a cubic equation in time, given the cubic coefficients. To derive the tendency at any given boundary point at any given time, the derivative quadratic equation is evaluated. The boundary data file is kept as a randomly-accessed file so that the coefficients for any variable at any level can be retrieved easily.

With these considerations, RLAM can readily offer a choice of four lateral boundary conditions: directly-inserted boundary at one or two boundary rows, Perkey-Kreitzberg (1976) boundary, Davies (1976) boundary, or cyclical boundary. The middle two have been used more extensively than the other two because they tend to blend the exterior and interior values efficiently. Although provision has been made for radiative boundary conditions, results from their use are, at best, unpredictable and require careful scrutiny. The cyclical boundaries are maintained mainly for test purposes when other boundary conditions could create complications.

c. Forecast Model

Upon receiving an initial data file the forecast model will produce a forecast of any desired length. The file can contain either an initial data set or a previous forecast with the expectation that the model will extend the forecast. The boundary files in all cases are set in accordance with the starting time of the initial file. If the starting time is other than zero, the boundary file will search for the boundary values appropriate for the restart time.

The model solves a set of general equations where certain terms may even be omitted according to a selection

of parameters. The equations are blind to region or map projection, their dependency being hidden in the map factors, terrain fields, and other field parameters selected and stored in the initial conditions. The equations involved are two momentum equations for each of the two wind components, a thermodynamic equation, a moisture equation, the continuity equation to determine surface pressure and vertical velocity, the hydrostatic equation, and the equation of state for an ideal gas. The momentum, thermodynamic, and moisture equations offer options of diffusion, physical forcing terms, and source and sink terms. In addition, smoothing is available using a Shapiro (1970) smoother-desmoother for any variable at prescribed time steps.

In solving the equations the model offers the possibility of choosing among a second-order, fourth-order, or fourth-order compact scheme. Time differencing options allow for leap-frog, Brown-Campana, or semi-implicit schemes. All these options work in concert with staggered grids, as well as with the selected boundary conditions. The second-order differencing is a subset of fourth-order differencing and requires fewer boundary points, as described in Gerlach (1985). The fourth-order compact scheme requires fewer boundary points than the regular fourth-order scheme, but an assumption must be made regarding the nature of the derivative at the boundary. The fourth-order compact is solved by relating a numerical representation of the derivative at some point with derivatives at neighboring points. Such a relationship involves a tri-diagonal matrix which must be inverted in order to solve for the derivatives at each point. This delivers a high-accuracy solution but at the expense of sensitivity to the boundary conditions and the numerical approximations. This renders the scheme unstable in many cases unless increased smoothing is imposed. The increased

smoothing, however, detracts from the accuracy of the model and could render application of the fourth-order compact pointless.

The staggering is handled by generalized averaging of variables to the desired grid point. If the grid is unstaggered, the averaging defaults to the value of the variable itself. Details of the staggering parameters are mentioned in Gerlach (1986) and by Tung et al. (1987).

Diffusion can be added to the equations by setting a parameter that controls the call to the diffusion subroutine. Another parameter establishes whether the diffusion is of second or fourth order. The diffusion coefficients are set initially and allow up to four different values dependent on where one is performing the diffusion. The bulk of the interior of the domain is governed by one value, whereas the boundary is divided into three regions starting with five rows in from the end of the domain and ending with the last two rows. The variability in the diffusion coefficient is designed to account for enhanced diffusion at the boundaries. But, as the practice has been with other limited-area models, the coefficients are treated as constants and are not included in the horizontal derivatives.

The smoother-desmoother is invoked at optional times throughout the forecast. Smoothing at the boundaries (i.e., the last five rows) can be performed at times different from the rest of the field, if so desired. Normally it is preferable to smooth the boundaries more frequently than the interior; this was indeed the case during most of the experiments. When the boundaries are cyclical, however, the smoothing at the boundaries automatically joins the rest of the field. By repeating the smoother, the degree of smoothing is automatically raised to filter longer waves.

Horizontal smoothing or diffusion of any variable is normally performed on the variable itself and not on some departure of the variable from a mean state. In the case of surface pressure, however, smoothing or diffusion of surface pressure itself without regard to the topography that is a major component in its horizontal variation could result in very bad imbalances in the pressure field. It is then offered as an option to smooth pressure only after it has been reduced to sea-level values. After smoothing, the pressures are hydrostatically raised or reduced to equivalent surface values.

The model physics, as mentioned earlier, consists of a bulk boundary layer parameterization, a large scale convective adjustment, and a Kuo convective scheme. These were all taken from the RWM employed at GWC. The user has the choice of invoking all, any, or none of the physical parameterizations. To date, only one set of physical parameterizations is available with RLAM. However, since the physical processes are completely independent of the rest of the model, it should not be difficult to replace the current physical package with any new physical package.

The forecast fields are written to file in binary form (unformatted) at specified times and in a set order. These forecasts may then be analyzed, plotted, or serve as initial conditions for a continued forecast. The output file contains temperature, geopotential, horizontal wind velocity components, and moisture at every layer, followed by surface pressure and 12 h accumulated precipitation amounts.

d. Post-processing

Post-processing of the fields can be invoked in order to facilitate analysis and evaluation of the forecast fields. First, the surface pressure is reduced to sea-level pressure and the $\sigma$-layer variables interpolated to mandatory atmospheric pressures using the NMC subroutine PTOSIG found in the description of the AFGL GSM (Brenner et al., 1984). Also available is the NCAR plotting package, which allows contour plotting over selected portions of the globe for any given mapping. Once variables are assigned to mandatory atmospheric levels, comparisons can be made with GSM forecasts (also interpolated to the mandatory levels) and NMC or FGGE analyses, which are already available on mandatory levels.

3. Semi-implicit Scheme

Much work has been devoted to installing a semi-implicit scheme in RLAM in coordination with the modular differencing schemes. The semi-implicit temporal differencing was modeled after the Australian Numerical Meteorology Research Centre (ANMRC, now Bureau of Meteorology Research Centre, BMRC) as given by McGregor et al. (1978). The concept behind semi-implicit schemes is simply to divide the basic equations into linear and non-linear segments. The linear portion is then averaged in time while the non-linear portion is kept at the present, known time step. By substitution, the six equations governing the dependent variables are reduced to one. Once that equation is solved, the other variables may be derived from the already determined single variable.

In RLAM, the original analytical equations are given as follows:

$$\text{a.}\quad \frac{\partial u}{\partial t} = - m_x m_y \left[ u \frac{\partial}{\partial x} \left( \frac{u}{m_y} \right) + v \frac{\partial}{\partial y} \left( \frac{u}{m_x} \right) \right] - \dot{\sigma} \frac{\partial u}{\partial \sigma}$$
$$- m_x \left[ \frac{\partial \phi}{\partial x} + RT \frac{\partial \ln p_*}{\partial x} \right] + fv + F_x$$

$$\text{b.}\quad \frac{\partial v}{\partial t} = - m_x m_y \left[ u \frac{\partial}{\partial x} \left( \frac{v}{m_y} \right) + v \frac{\partial}{\partial y} \left( \frac{v}{m_x} \right) \right] - \dot{\sigma} \frac{\partial v}{\partial \sigma}$$
$$- m_y \left[ \frac{\partial \phi}{\partial y} + RT \frac{\partial \ln p_*}{\partial y} \right] - fu + F_y \qquad (1)$$

$$\text{c.}\quad \frac{\partial T}{\partial t} = \frac{T}{\pi} \frac{\partial \pi}{\partial t} - m_x u \frac{\partial T}{\partial x} - m_y v \frac{\partial T}{\partial y} + \frac{T}{\pi} \left[ m_x u \frac{\partial \pi}{\partial x} + m_y v \frac{\partial \pi}{\partial y} \right]$$
$$- \dot{\sigma} \frac{\partial T}{\partial \sigma} + \frac{\dot{\sigma}}{\pi} T \frac{\partial \pi}{\partial \sigma}$$

$$\text{d.}\quad \frac{\partial \ln p_*}{\partial t} = - m_x m_y \left[ \frac{u}{m_y} \frac{\partial \ln p_*}{\partial x} + \frac{v}{m_x} \frac{\partial \ln p_*}{\partial y} \right]$$
$$- m_x m_y \left[ \frac{\partial}{\partial x} \left( \frac{u}{m_y} \right) + \frac{\partial}{\partial y} \left( \frac{v}{m_x} \right) \right] - \frac{\partial \dot{\sigma}}{\partial \sigma}$$

$$\text{e.}\quad \frac{\partial \phi}{\partial \sigma} = - \frac{RT}{\sigma} ,$$

where

u, v are the horizontal wind components

T is temperature

$\sigma$ is the vertical coordinate = $p/p_*$

$\dot{\sigma}$ is the vertical wind component

p is pressure

$p_*$ is surface pressure

ln $p_*$ is the natural logarithm of $p_*$

$F_x$, $F_y$ are diffusion, friction, and forcing terms

$\pi$ is $\left(\frac{p_* \sigma}{p_o}\right)^{\kappa}$, $p_o$ = 1000 mb, $\kappa$ = .28562

R is the ideal gas constant for dry air

$\phi$ is the geopotential
$m_x$, $m_y$ are the map factors
x, y, and t are the independent variables in horizontal space and time.

These equations are modified by setting $T = T_o + T'$, where $T_o$ is a mean temperature profile dependent only on $\sigma$, while $T'$ is the departure of the actual temperature from the mean state. We also define a new variable $w = \dot{\sigma} + \frac{\partial \ln p_*}{\partial t}$ and rewrite Eq. (1)c and Eq. (1)d in terms of w. We separate the linear and non-linear terms of the equations and apply a centered differencing temporal scheme to forecast the variables with the provision that all linear terms are to be averaged in time. For any variable, A, the operator $\bar{A}^t$ indicates $\bar{A}^t \equiv \frac{1}{2}(A^{n+1} + A^{n-1})$, where n is the time step. Time derivatives are then approximated as $\frac{\partial A}{\partial t} \simeq (\bar{A}^t - A^{n-1})/\Delta t$ where $\Delta t$ is the time increment and $t = n\Delta t$. With these modifications and conventions Eq. set (1) becomes:

$$\text{a.}\quad \dot{u}^t + \Delta t\, m_x \left( \frac{\partial \bar{\phi}^t}{\partial x} + RT_o \frac{\partial \overline{\ln p_*}^t}{\partial x} \right) = a + \Delta t\, m_x\, RT_o \frac{\partial \ln p_*^{n-1}}{\partial x}$$

$$\text{b.}\quad \bar{v}^t + \Delta t\, m_y \left( \frac{\partial \bar{\phi}^t}{\partial y} + RT_o \frac{\partial \overline{\ln p_*}^t}{\partial y} \right) = b + \Delta t\, m_y\, RT_o \frac{\partial \ln p_*^{n-1}}{\partial y} \qquad (2)$$

$$\text{c.}\quad \bar{T}^t - \Delta t \left( \frac{T_o \kappa}{\sigma} - \frac{\partial T_o}{\partial \sigma} \right) \bar{w}^t - \Delta t\, \sigma \frac{\partial T_o}{\partial \sigma} \bar{w}_*^t = c$$

$$\text{d.}\quad \frac{\partial \bar{w}^t}{\partial \sigma} + m_x m_y \left[ \frac{\partial}{\partial x} \left( \frac{\bar{u}^t}{m_y} \right) + \frac{\partial}{\partial y} \left( \frac{\bar{v}^t}{m_x} \right) \right] = M,$$

where a, b, c, and M are non-linear terms evaluated at time steps n and n - 1. a and b also contain the term $-\Delta t R T_o \mathrm{der}(\ln p_*^{n-1})$, where der is the x or y derivative. The addition and subtraction of this term is designed to allow for combining $R T_o \mathrm{der}(\overline{\ln p_*}^t)$ with $R T_o \mathrm{der}(\ln p_*^{n-1})$ to give $\Delta t R T_o \mathrm{der}(\bar{w}_*^t)$ where, because of the condition that $\dot{\sigma} = 0$ at $\hat{\sigma} = 1$ and 0 (the circumflex signifies level $\sigma$), $w_* = \frac{\partial \ln p_*}{\partial t}$. By dividing Eqs. (2)a and b by their respective map factors, taking the respective derivative of each equation, and substituting into Eq. (2)d, one derives a relationship between w and $\phi$, i.e.:

$$(\Delta t)^{-1} \frac{\partial \bar{w}^t}{\partial \sigma} - \nabla^2 \bar{\phi}^t - \nabla^2 R T_o \Delta t \bar{w}_*^t = d \qquad (3)$$

where

$$d = m_x m_y (\Delta t)^{-1} \left[ \frac{\partial}{\partial x} \left( \frac{a}{m_y} \right) + \frac{\partial}{\partial y} \left( \frac{b}{m_x} \right) + \frac{u}{m_y} \frac{\partial \ln p_*}{\partial x} + \frac{v}{m_x} \frac{\partial \ln p_*}{\partial y} \right].$$

Numerical representation of the above expression can be stated in matrix form as follows:

$$\mathbf{C}(\bar{W}^t) - \nabla^2 \bar{\Phi}^t - \mathbf{E} \nabla^2(\bar{W}^t) = - m_x m_y D, \qquad (4)$$

where the block letters indicate K x K two-dimensional matrices and the upper-case letters indicate vectors of their lower-case counterparts, which are of dimension K x 1 and are spatially dependent. The matrix **C** serves to take the derivative of w, defined at the levels, with respect to the intermediate layer, remembering that w at $\hat{\sigma}_{K+1}$ is 0 because $\hat{\sigma}_{K+1} = 0$ and $\dot{\sigma}_{K+1} = 0$. It is, thus, of the very

simple form:

$$\mathbf{C} = (\Delta t)^{-1} \begin{pmatrix} (\Delta\sigma_1)^{-1} & -(\Delta\sigma_1)^{-1} & 0 & \cdots & 0 \\ 0 & (\Delta\sigma_2)^{-1} & -(\Delta\sigma_2)^{-1} & 0 \cdots & 0 \\ 0 & 0 & (\Delta\sigma_3)^{-1} & -(\Delta\sigma_3)^{-1} \cdots & 0 \\ \cdot & \cdot & 0 & 0 \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot \cdots & \cdot \\ 0 & 0 & 0 & \cdots & (\Delta\sigma_K)^{-1} \end{pmatrix}$$

where $\Delta\sigma_k = \hat{\sigma}_{k+1} - \hat{\sigma}_k$. The matrix $\mathbf{E}$ is an expression of the last term on the right-hand side of Eq. (3), namely:

$$\mathbf{E} = R\Delta t \begin{pmatrix} (T_o)_1 & 0 & 0 & \cdots & 0 \\ (T_o)_2 & 0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ (T_o)_K & 0 & \cdot & \cdots & 0 \end{pmatrix} .$$

By hydrostatic considerations, the geopotential $\phi$ can be related to the temperature T. The nature of this relationship has been a matter of investigation for several years. Recently, Brenner (1988) compared various options in expressing the hydrostatic equation and lists the possible pitfalls with each of them. In most of our considerations we have chosen the form attributed to Arakawa and Lamb (1977) which has certain desirable conservative properties, but which is notoriously inaccurate when trying to reverse the process and derive T from $\phi$. The matrix form of the hydrostatic equation is:

$$\mathbf{B}T = \Phi - \Phi_*$$

where T is now regarded as a K-dimensional vector and $\Phi_*$ is a K x 1 vector consisting of K equal components of $\Phi_*$, the surface geopotential (which does not depend on the vertical). For the Arakawa representation, **B** is of the form:

$$\mathbf{B} = \begin{pmatrix} \delta_1 & \delta_2 & \delta_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \delta_K \\ B_1+\delta_1 & A_2+\delta_2 & \delta_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \delta_K \\ B_1+\delta_1 & A_2+B_2+\delta_2 & A_3+\delta_3 & \delta_4 & \cdot & \cdot & \cdot & \cdot & \cdot & \delta_K \\ B_1+\delta_1 & A_2+B_2+\delta_2 & A_3+B_3+\delta_3 & A_4+\delta_4 & \delta_5 & \cdot & \cdot & \cdot & \cdot & \delta_K \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ B_1+\delta_1 & A_2+B_2+\delta_2 & \cdot & \cdot & A_{K-1}+B_{K-1}+\delta_{K-1} & & & & A_K+\delta_K \end{pmatrix}$$

where

$$\delta_k = R\Delta\sigma_k - B_k \sum_{j=k+1}^{K} \Delta\sigma_j - A_k \sum_{j=k}^{K} \Delta\sigma_j ,$$

$$B_k = \frac{1}{2} c_p \left[ 1 - \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{\kappa} \right] , \quad A_k = \frac{1}{2} c_p \left[ \left( \frac{\sigma_{k-1}}{\sigma_k} \right)^{\kappa} - 1 \right] ,$$

$$A_1 = 0, \quad B_K = 0$$

and $c_p$ is the heat capacity of an ideal gas at constant pressure.

The thermodynamic equation, Eq. (2)c, provides a relationship between temperature and the variable w. In matrix form this becomes:

$$\mathbf{A}\bar{w}^t = \bar{T}^t - C, \tag{5}$$

where here again T refers to the vector. The matrix **A** is based on the σ-structure and the values of the mean temperature $T_o$:

$$\mathbf{A} = \begin{pmatrix} G_1+s_1 & s_1 & 0 & 0 & \cdots & 0 \\ G_2 & s_2 & s_2 & 0 & \cdots & 0 \\ G_3 & 0 & s_3 & s_3 & \cdots & 0 \\ G_4 & 0 & 0 & s_4 & s_4 \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ G_K & 0 & \cdots & \cdots & s_{K-1} & s_K \end{pmatrix}$$

*where*

$$s_k = \frac{1}{2}\Delta t\left[\frac{T^K_{ok}}{\sigma_k} - \left(\frac{\partial T_o}{\partial \sigma}\right)_k\right] \text{ and } G_k = \Delta t \sigma_k \left(\frac{\partial T_o}{\partial c}\right)_k .$$

Here too the bi-diagonal structure of the matrix stems from the necessity to average values of w from the levels to the layers. Now, substituting for $\bar{w}^t$ in Eq. (4) by inverting Eq. (5) and substituting for $\bar{\Phi}^t$ using the hydrostatic relationship leads to a single equation in the vector $\bar{T}^t$:

$$\mathbf{C}\mathbf{A}^{-1}\bar{T}^t - (\mathbf{B} + \mathbf{E}\mathbf{A}^{-1})\nabla^2\bar{T}^t = -m_x m_y D$$
$$+ \mathbf{C}\mathbf{A}^{-1} C - \mathbf{E}\mathbf{A}^{-1}\nabla^2 C + \mathbf{B}\nabla^2\Phi_* \equiv R. \tag{6}$$

The vector on the right-hand side, symbolized by the letter R, contains known quantities in terms of variables at time step n or n - 1. The matrices are all time independent and are functions only of the σ-structure of the model. We can re-arrange Eq. (6) to furnish the following Helmholtz equation in $\bar{T}^t$:

$$\nabla^2 \bar{T}^t - \mathbf{U}\,\bar{T}^t = R' \tag{7}$$

where

$$\mathbf{U} = (\mathbf{B} + \mathbf{E}\mathbf{A}^{-1})^{-1}\,(\mathbf{C}\mathbf{A}^{-1}) \text{ and } R' = -(\mathbf{B} + \mathbf{E}\mathbf{A}^{-1})^{-1} R.$$

In order to solve Eq. (7), the vertical must be decoupled from the horizontal. To do this, we find the K eigenvectors and eigenvalues of the matrix **U**. Multiplying through by the matrix **V**, whose rows are the eigenvectors of **U** and remembering that $\mathbf{U} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$, where

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 & . & . & . & . & 0 \\ 0 & \lambda_2 & 0 & . & . & . & . & 0 \\ 0 & 0 & \lambda_3 & 0 & . & . & 0 \\ . & . & . & & & & & . \\ . & . & . & & & & & . \\ 0 & 0 & 0 & . & . & . & . & \lambda_k \end{pmatrix}$$

$\lambda_k$ being the $k^{th}$ eigenvalue of **U**, we get

$$\nabla^2 \bar{\tau}^t - \boldsymbol{\Lambda}\,\bar{\tau}^t = \Gamma, \tag{8}$$

where

$$\bar{\tau}^t = \mathbf{V}^{-1}\,\bar{T}^t, \quad \Gamma \equiv \mathbf{V}^{-1}\,R'.$$

Eq. (8) is a set of K Helmholtz equations in two dimensions and can be solved by conventional methods. However, because the operator $\nabla^2$ contains generalized map factors that depend on x and y, direct solution methods are not applicable. Instead, relaxation techniques are employed to solve for each $\bar{\tau}^t$; then, multiplying $\bar{\tau}^t$ by **V** retrieves the value of $\bar{T}^t$.

As an aside, we must mention that the matrix **U** does not a priori exclude complex eigenvalues or eigenvectors. Yet, in practice, we have discovered that at least for our formulation of $T_o$ and $\sigma$, **U** has only real eigenvalues and vectors. This leads to a saving of storage and facilitates computations. However, one should be aware that with any change of the $\sigma$-structure or $T_o$ values, there is a chance that complex values will be introduced and, if so, a set of complex Helmholtz equations must be solved. A real solution for $\bar{T}^t$ will result nonetheless when conjugate solutions are added and the imaginary portions cancel.

Once $\bar{T}^t$ is found we may invert Eq. (5) to obtain values of $\bar{w}^t$ which furnish updated $\dot{\sigma}$ and ln $p_*$ values. The hydrostatic equation can also provide $\bar{\Phi}^t$. These, when inserted in Eq. (2)a and b, yield $\bar{u}^t$ and $\bar{v}^t$. Moisture values are derived strictly in an explicit formulation similar to a leap-frog scheme. Once all variables are derived, smoothing and boundary adjustments may be made.

## 4. Results

### a. Semi-implicit Scheme Experiments

Despite theoretical consistency, the semi-implicit scheme has not functioned well in context with RLAM. The cause of the problem remains unclear. McGregor et al.

(1978) state that the scheme should be stable for up to $\Delta t$ = 2400 s. However, that depends a great deal upon the spatial resolution (which in their case is 250 km) and the wind speeds encountered. In mid- to high-latitudes, with wind speeds close to 100 m $s^{-1}$ at higher elevations and spatial resolutions of 200 km or less, time steps of 1200 to 1800 s are more appropriate. But even with these time increments, apparent instabilities arise in the forecast which eventually swamp any reasonable solution. To aid in the investigation, analytical initial conditions suggested by Fritsch et al. (1980) were employed with cyclical boundaries in the east-west direction and fixed boundaries in the north-south direction. These tailored conditions allowed for pinpointing possible problems without the additional problems generated by boundaries. The major fault seems to lie with the surface pressure tendency. Despite reasonable changes of temperature with time (about 0.1°C per 1200 s), Eq. (5) still yields large pressure changes. Indeed, a norm of the inverse of the matrix **A** has a value of about $10^{-5}$. When combined with values of $\bar{T}^t - c$ of even 0.1°C, the change in ln $p_*$ is of the order of $10^{-6}$, which is much too large even for a sizeable time step. It was surmised that perhaps the relationship expressed in Eq. (5) is similar to the hydrostatic relationship of Arakawa in that it works well in only one direction, i.e., T to $\phi$ or w to T - c, but not in an inverse mode.

Thus in an attempt to avoid using Eq. (5) directly, Eq. (4) was used instead with substitutions for $\Phi$ and T, yielding a single equation in terms of only $\bar{W}^t$, which could then be found by the same procedure as mentioned previously for $\bar{T}^t$. Unfortunately, this, too, led to unreasonably large values of $\bar{W}^t$ and hence large surface pressure changes.

The next attempt to limit the large pressure change involved using Eq. (5) only as a "first guess" in deriving $\bar{W}_*^t$.

Then, once $\bar{u}^t$ and $\bar{v}^t$ were known from Eqs. (2)a and b with the help of $\overline{\ln p_*}^t$ supplied from the first guess of $\bar{w}_*^t$, Eq. (2)d could be integrated vertically to supply a new value for $\bar{w}^t$. This approach had some limited success. Although it seems to ask for an iterative process whereby the second guess values of $w_*$ would again be substituted into Eqs. (2)a and b, such a procedure would of necessity consume a great deal of time and eliminate the benefits of a semi-implicit scheme. Thus, only one iteration was permitted, yielding, at times, a significantly different value for $w_*$ after the integration of Eq. (2)d. This kept the forecast somewhat stable but only for regions with fairly smooth terrain. When the method was introduced over mountainous areas, the large swings in surface pressure returned.

Fig. 1 displays a 30 h GSM forecast of sea-level pressure, verifying at 1800 GMT 18 February 1979, over an area of the Pacific Ocean centered at 15°S, 135°W. The domain covers an area of approximately 65° longitude by 55° latitude. Fig. 2 is the equivalent RLAM 24 h forecast verifying at the same time but beginning with the GSM 6 h forecast (to avoid initial imbalances). Here the semi-implicit scheme was employed with a time step of 1200 s on an unstaggered grid, Mercator projection, with a 1/3 mesh (approximately 1.67° longitude or latitude).

Boundary conditions consisted of a sponge boundary with smoothing every time step at the boundaries and every other time step in the interior. Because of the region of the world involved, the sea-level pressure is rather flat for most of the interior. There is a relatively large high-pressure area in the southeast corner of the domain with a low right off the southern boundary. RLAM tends to build the high by approximately 6 mb more than the GSM, but this may be an effect of the boundary condition. On the other hand, pressures seem to be higher in many other locations.

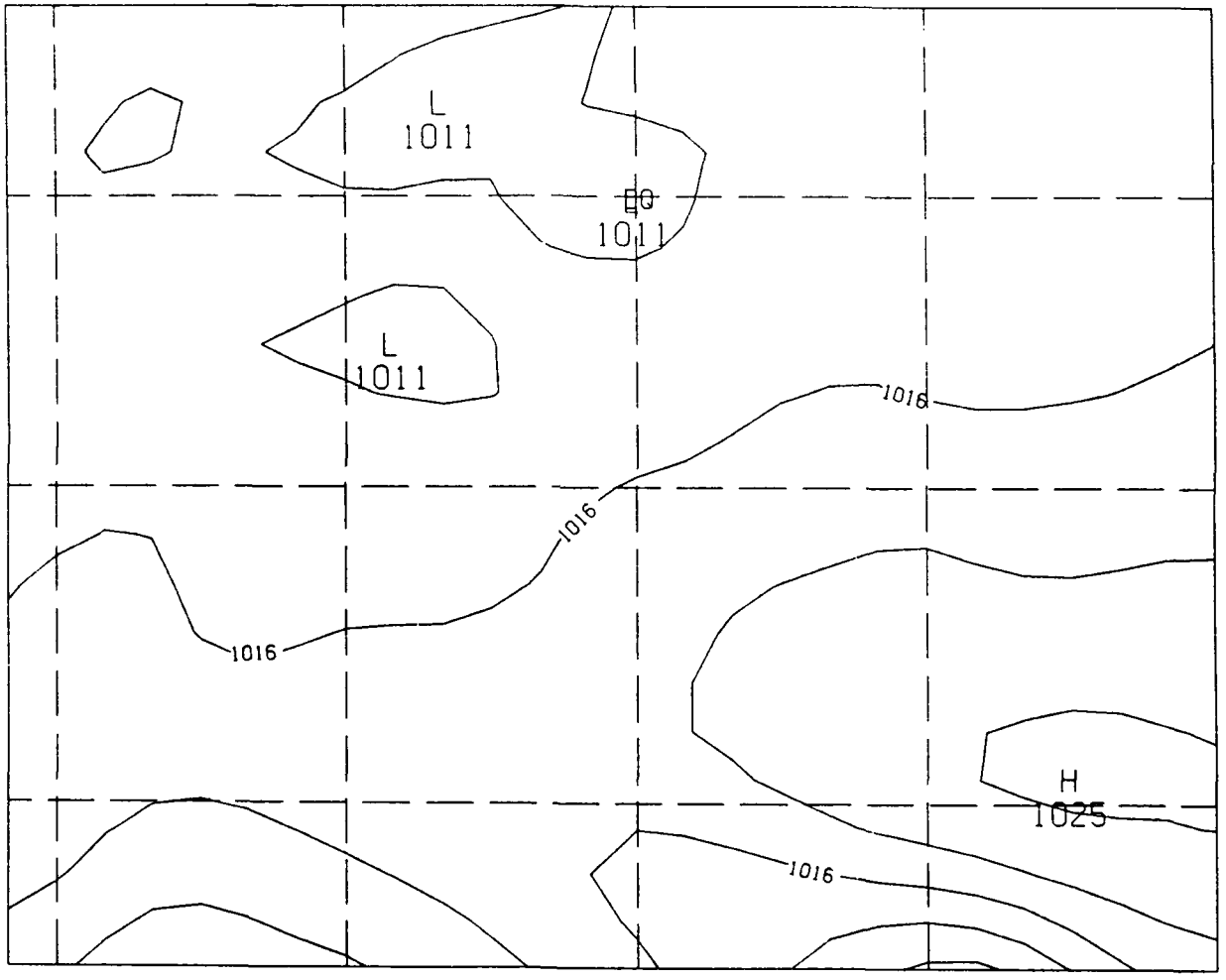



Fig. 1. GSM 30 h Forecast of Sea-level Pressure Verifying at 1800 GMT 18 February 1979. Forecast Area Centered at 15°S, 135°W.

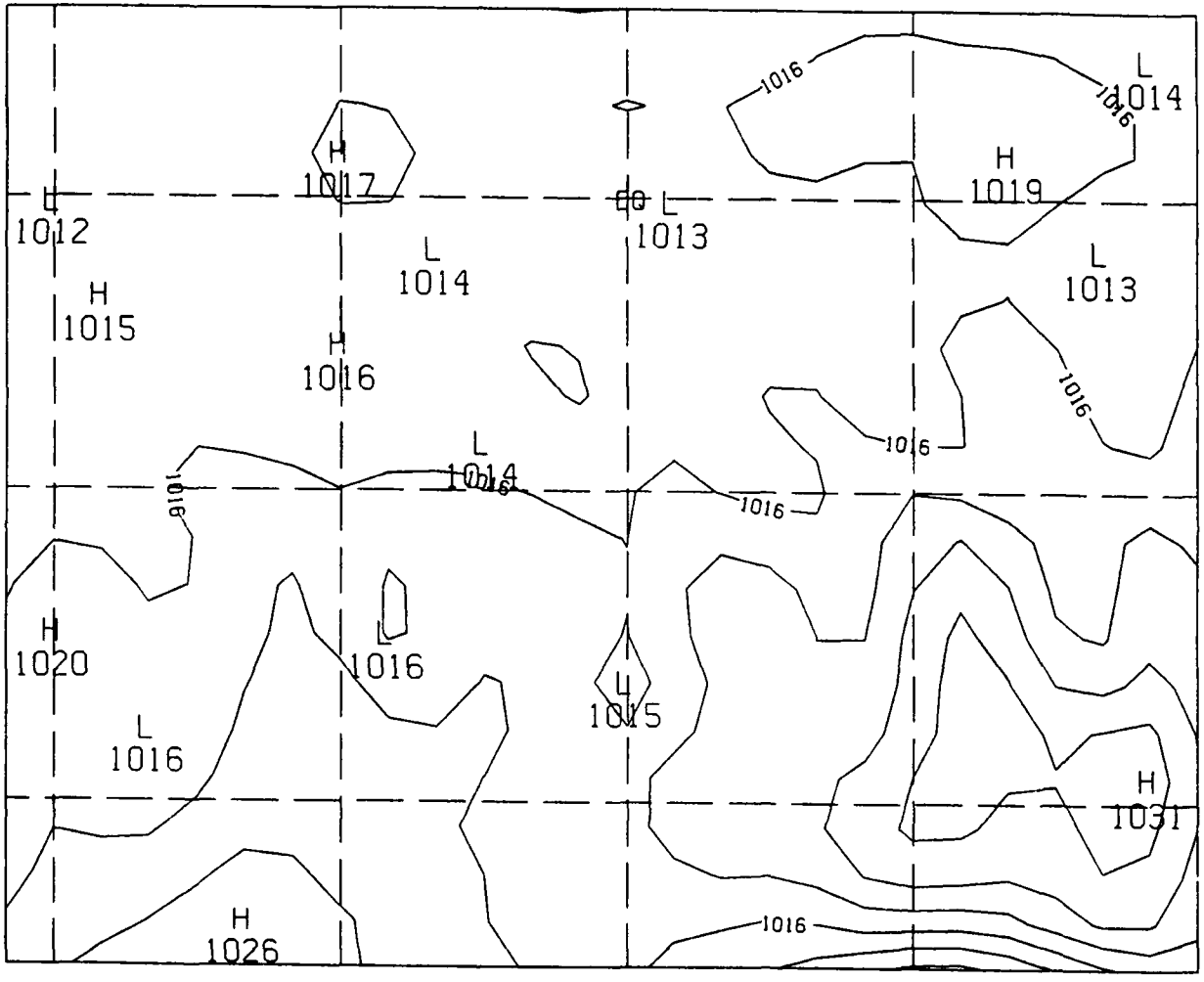



Fig. 2. RLAM 24 h Forecast of Sea-level Pressure Verifying at 1800 GMT 18 February 1979 for Same Region as in Fig. 1. Forecast Initialized with 6 h Forecast of GSM and Produced with Semi-implicit Scheme and Time Increment of 2400 s.

Part of the choppiness in the RLAM forecast may be due to the numerical scheme, while part is also due to not invoking the post-processor smoothing routine for this field. On the whole this is not an impressive 24 h forecast and does not bode well for longer term forecasts or for forecasts in more dynamic areas of the world where noise is increased and leads to instabilities.

In another attempt to avoid Eq. (5), we attempted to apply Eq. (3) to find $\bar{w}^t$. After deriving $\bar{T}^t$ and applying the hydrostatic relationship to obtain $\bar{\phi}^t$, we integrated Eq. (3) vertically to produce a Helmholtz equation in $\bar{w}_*^t$, since $\bar{w}^t$ at $\sigma = 0$ is zero. Once $\bar{w}_*^t$ is known it is fairly simple to integrate upwards and calculate $\bar{w}^t$ at all levels. This method still did not furnish a stable forecast of surface pressure, although it lengthened the period of integration before the oscillations in the pressure field became untenable.

After consultation with L. Leslie, one of the original authors of the ANMRC model, it was decided that the $\sigma$-structure may be a culprit in causing the discrepancies. The Australian model employs neither Arakawa's vertical structure nor his version of the hydrostatic equation. Instead, it utilizes Brenner's (1988) scheme B, attributed to Bourke (1974), which derives $\sigma$ (at layers) by strictly averaging $\hat{\sigma}$ (at levels). Also unlike Arakawa, Bourke's hydrostatic assumption does not require an integral constraint to determine the lowest layer height; instead, it depends only on temperature at the lowest layers. A few experiments with RLAM, substituting the Bourke hydrostatic assumption for the Arakawa assumption, were inconclusive. It is thought that a detailed comparison of RLAM with the Australian model's code might lead to a solution.

b. Presidents' Day Storm Forecast Simulations

On 18-19 February 1979 an intense cyclone developed off the eastern shore of the US and brought heavy snow and high winds to the mid-Atlantic coastal states and the District of Columbia. This storm, known as the Presidents' Day storm because of its concurrence with that holiday, has been intensely studied by meteorologists. The storm was particularly difficult to forecast, mainly because it was a small scale disturbance that developed rapidly and deepened precipitously, catching man and model offguard (Bosart, 1981). In an attempt to display the flexibility of RLAM and to pinpoint some of the elements that may have been crucial in forecasting the storm, a series of RLAM forecasts was generated with different options and compared with the GSM and RWM forecasts.

Fig. 3a shows the FGGE IIIB 500 mb analysis over North America at 1200 GMT 17 February 1979; Fig. 3b is the sea-level pressure analysis for the same time. The weather pattern is dominated by a very large high-pressure area over the eastern two-thirds of the US and a deepening upper air trough over the Rockies. Figs. 4a and b are the same respective analyses 48 h later at 1200 GMT 19 February 1979. The upper air trough has deepened over the East Coast, while the surface high has split and allowed an intense cyclone to form offshore. The central pressure of 1011 mb is in sharp contrast to the much higher pressure at the centers of the neighboring highs, the large pressure gradients being instrumental in increasing the ferocity of the storm. The GSM 48 h forecast, based on an initialized version of the FGGE IIIB analysis of 17 February, develops a storm in about the correct position but underestimates its intensity and the depth of the 500 mb trough, as depicted in Figs. 5a and b.

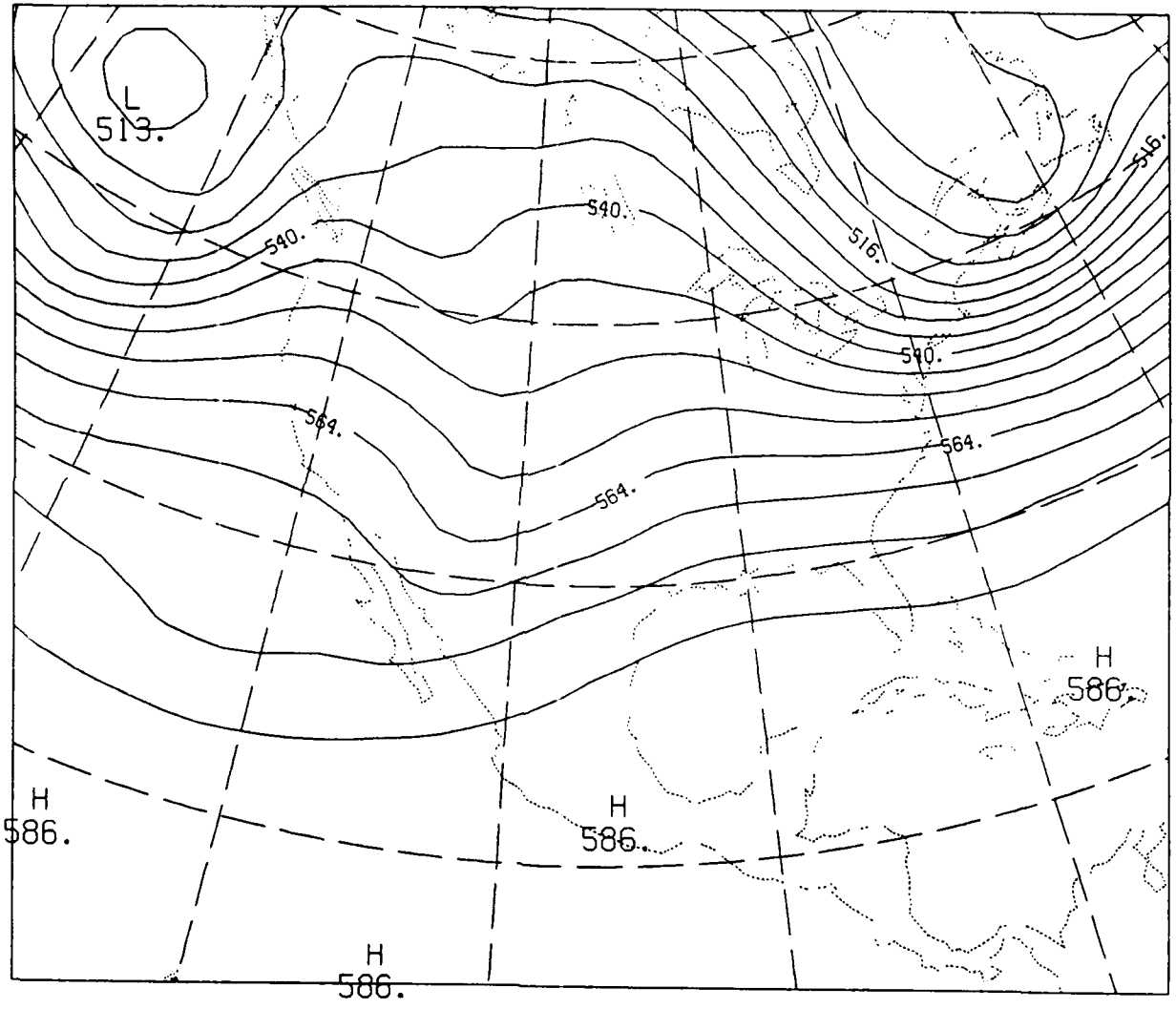



Fig. 3a. FGGE III-B Analysis of 500 mb Heights at 1200 GMT 17 February 1979. Domain Centered at 35°N, 100°W.

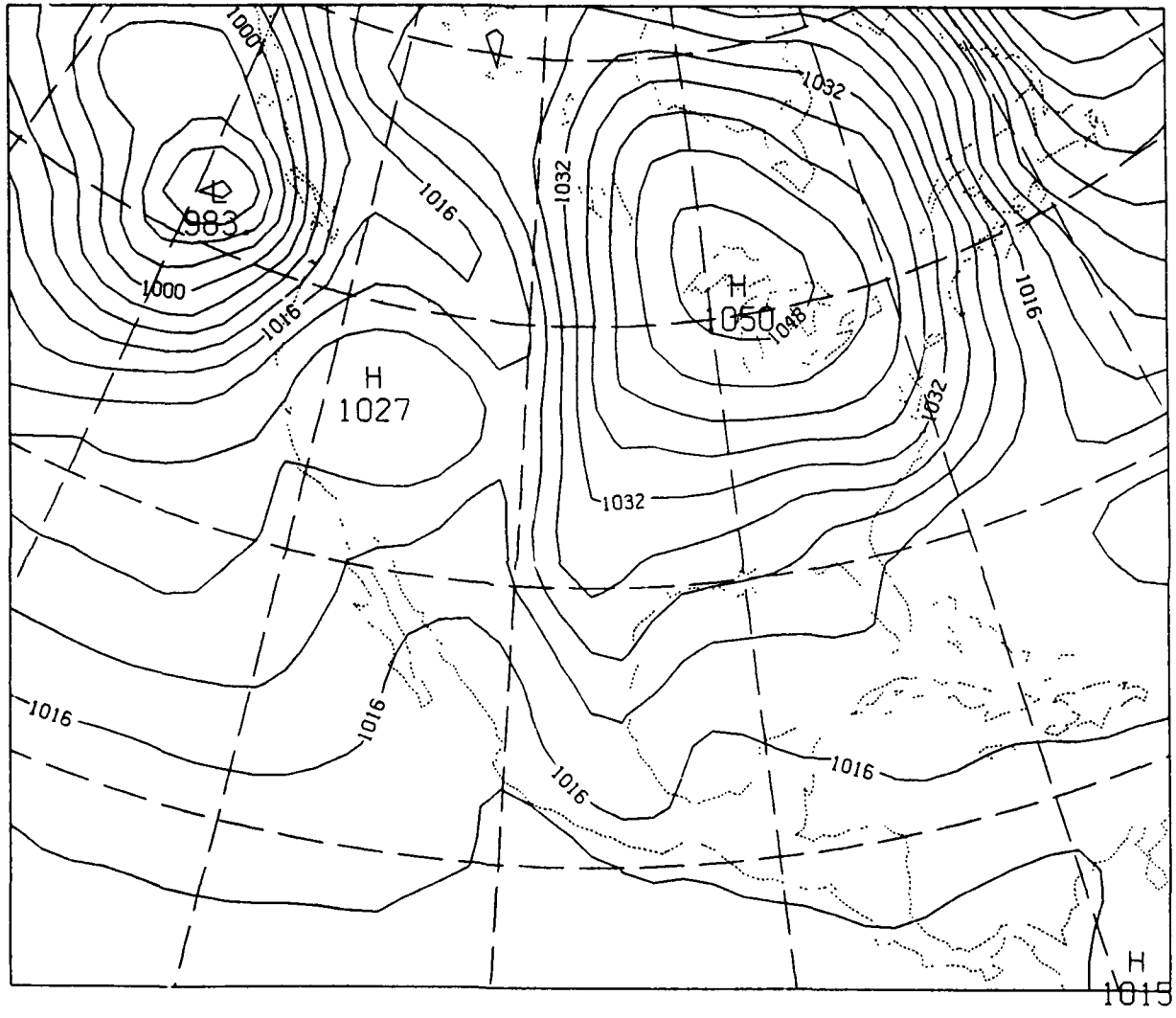



Fig. 3b. FGGE III-B Analysis of Sea-level Pressure at 1200 GMT 17 February 1979. Domain Centered at 35°N, 100°W.

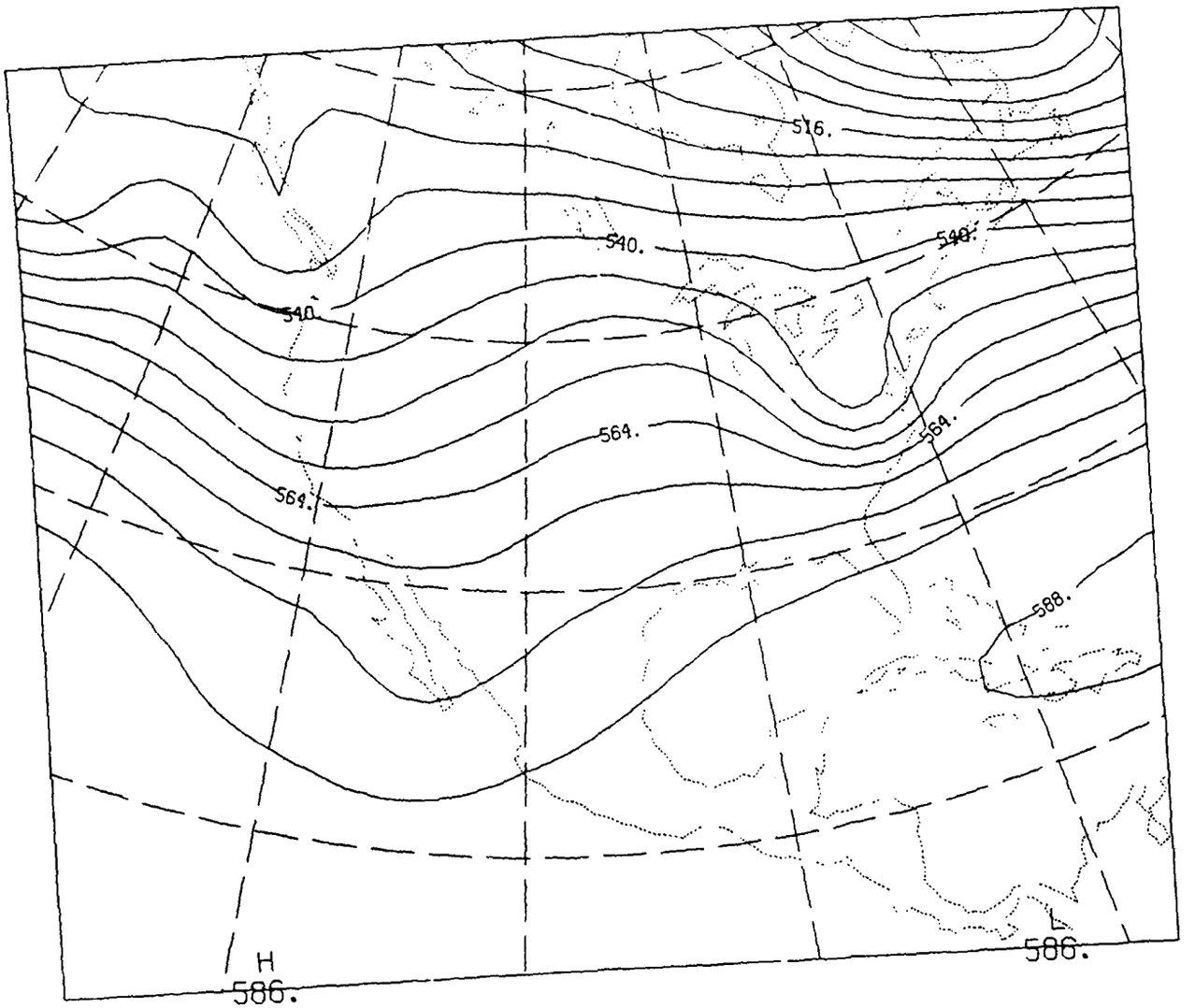



Fig. 4a. Same as Fig. 3a but for 1200 GMT 19 February 1979.

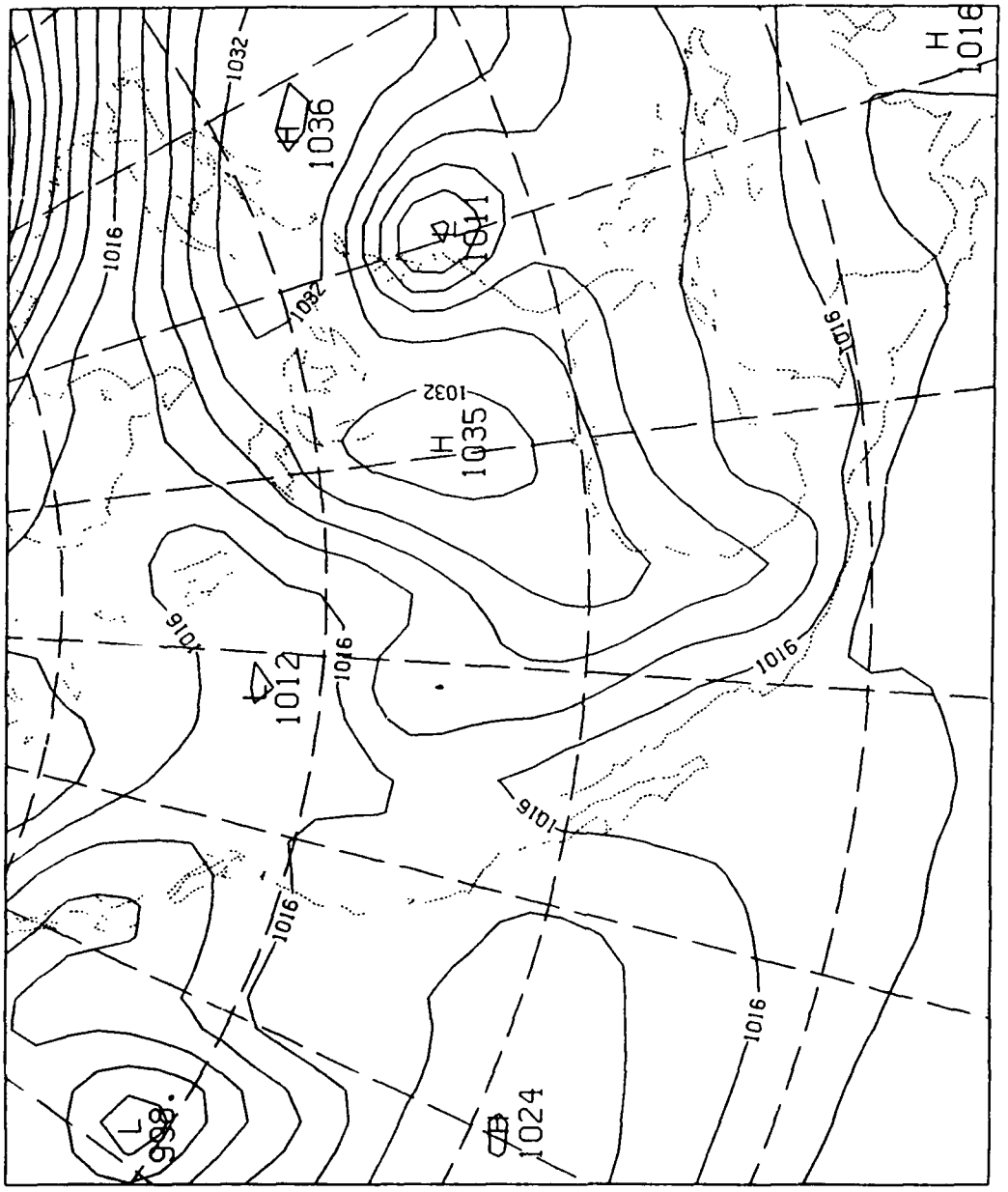



Fig. 4b. Same as Fig. 3b but for 1200 GMT 19 February 1979.

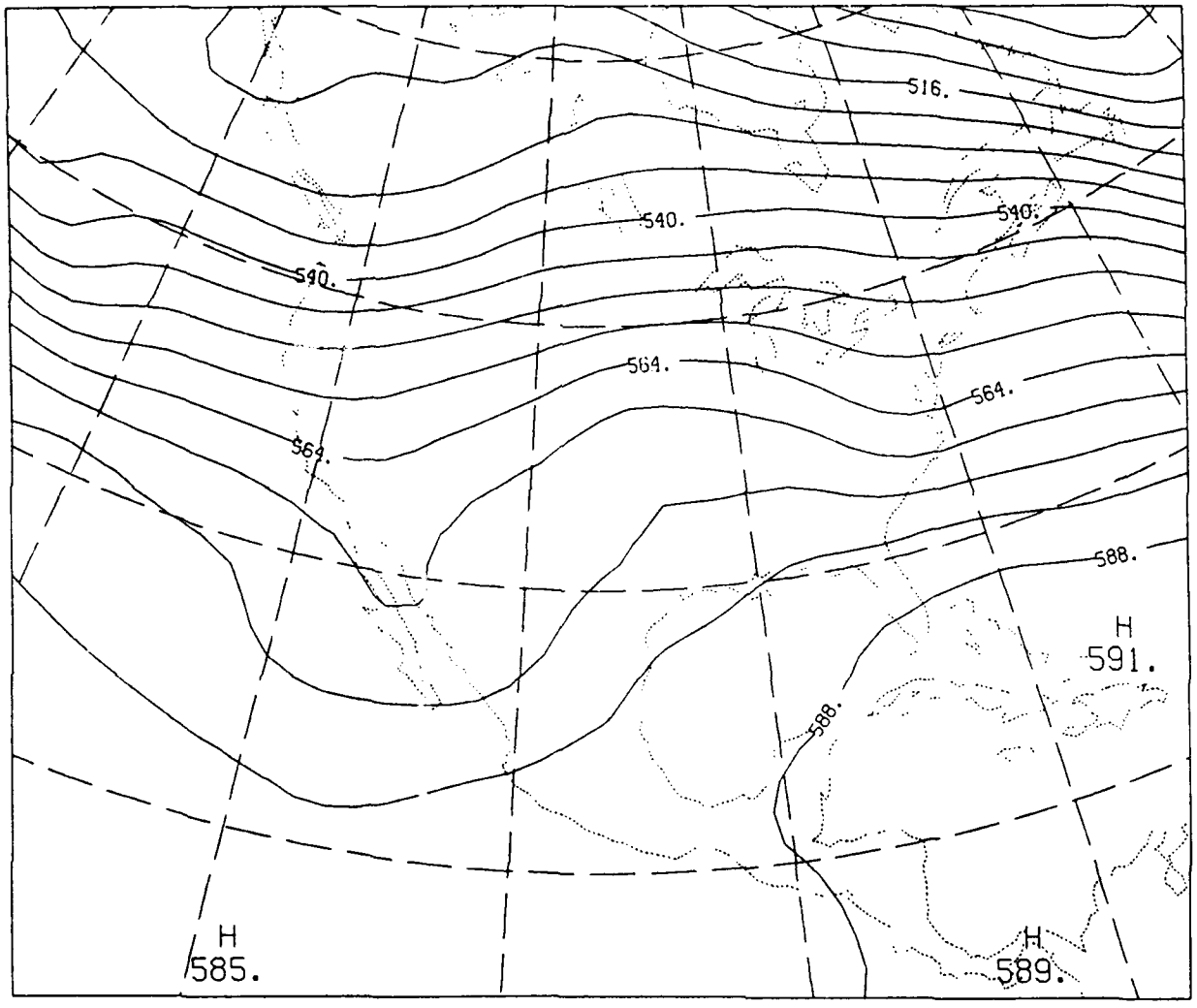



Fig. 5a. GSM 48 h Forecast of 500 mb Heights Verifying at Same Time and Region as Fig. 4a.

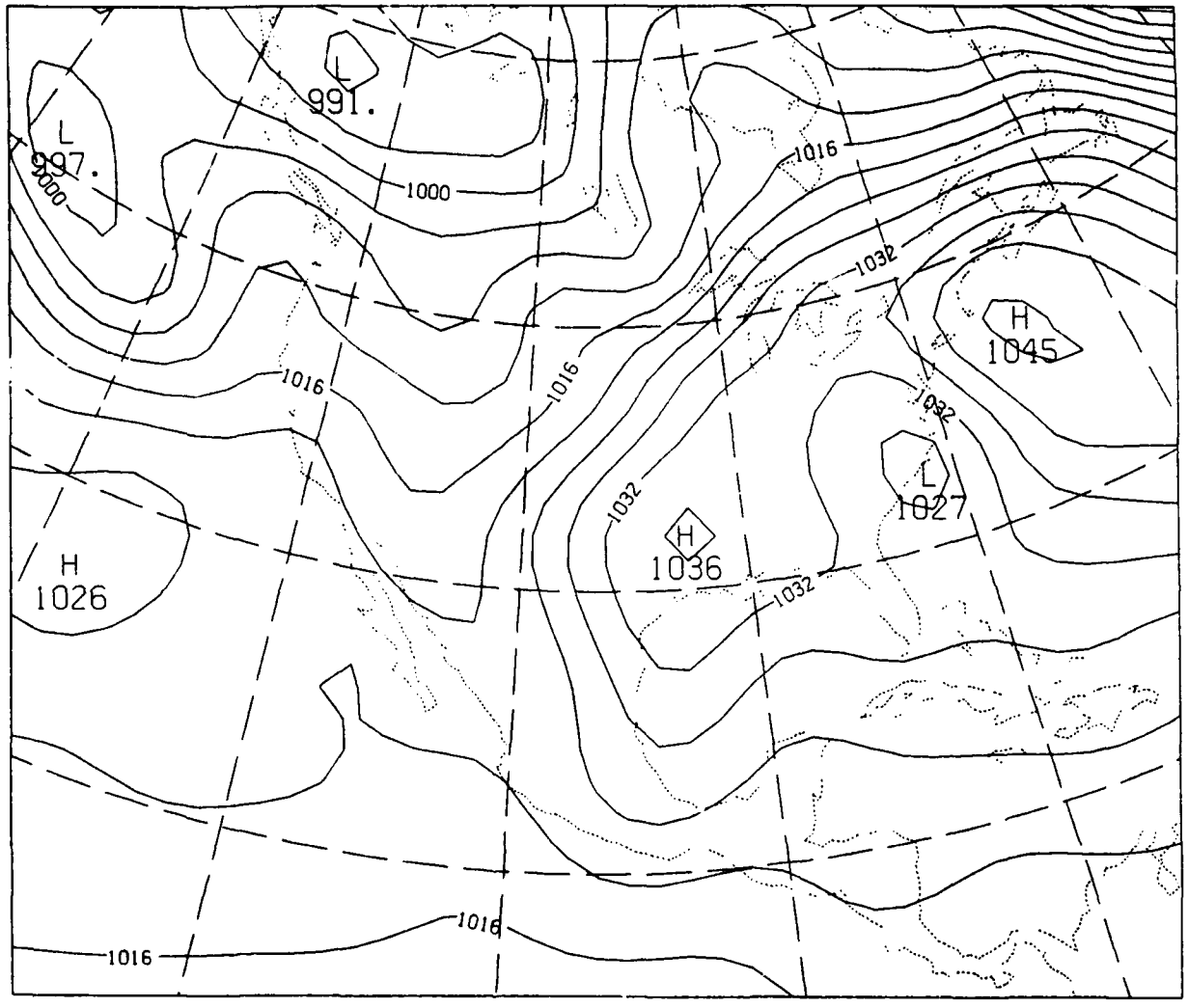



Fig. 5b. GSM 48 h Forecast of Sea-level Pressure Verifying at Same Time and Region as Fig. 4b.

RLAM produced several 48 h forecasts beginning with the same initial state as the GSM and with boundary conditions derived from the GSM fields every 6 h. The map projection for this region was Lambert conformal and the lateral boundary conditions imposed were always the Davies type which, in general, yielded results very similar to the Perkey-Kreitzberg sponge boundary. Some experiments were performed using a 1/2 mesh grid centered over the mid-US while some invoked a 1/4 mesh grid centered over the eastern US. These will be discriminated in the discussion below as well as in the figures. The grids were normally unstaggered.

Lacking a working semi-implicit scheme, the model ran with a Brown-Campana temporal scheme with a time step of 200 s for the coarse grid and 120 s for the higher resolution grid. The GSM forecasts depicted in Figs. 5a and b were generated with an updated physical package which on the whole resembles the physics of the RWM and RLAM, the major difference being an arbitrary halving of evaporative fluxes over the oceans to prevent the model atmosphere from becoming too moist.

Fig. 6 is an RLAM 48 h forecast of sea-level pressure verifying at the same time as Fig. 5, produced by taking second-order differencing and smoothing every third time step at the boundaries and every ninth time step over the entire field (3, 9 smoothing). This forecast does not produce a pronounced cyclone at all, just a hint of some low pressure far off the East Coast. Fig. 7 shows a similar RLAM forecast with fourth-order differencing and smoothing at the boundaries at every time step and every other time step over the entire field (1, 2 smoothing). The smoother appearance of the contours is due to post-processing of the plot fields to remove high frequency noise. Here again the cyclone does

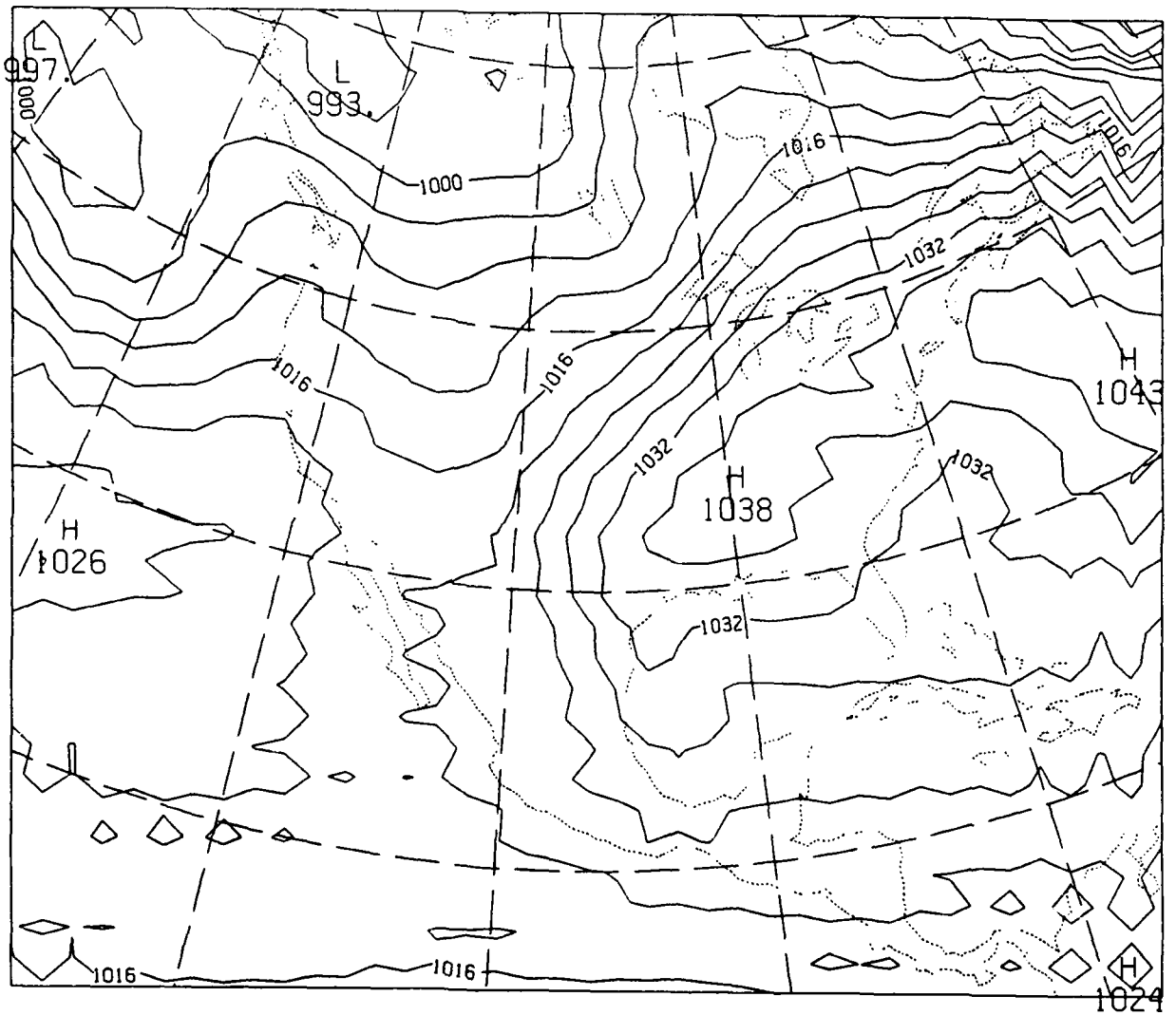



Fig. 6. RLAM 48 h Forecast of Sea-level Pressure Verifying at Same Time and Region as Fig. 4b, Produced with Second-order Differencing and 3, 9 Smoothing on Grid with 1/2 Standard Mesh Resolution.

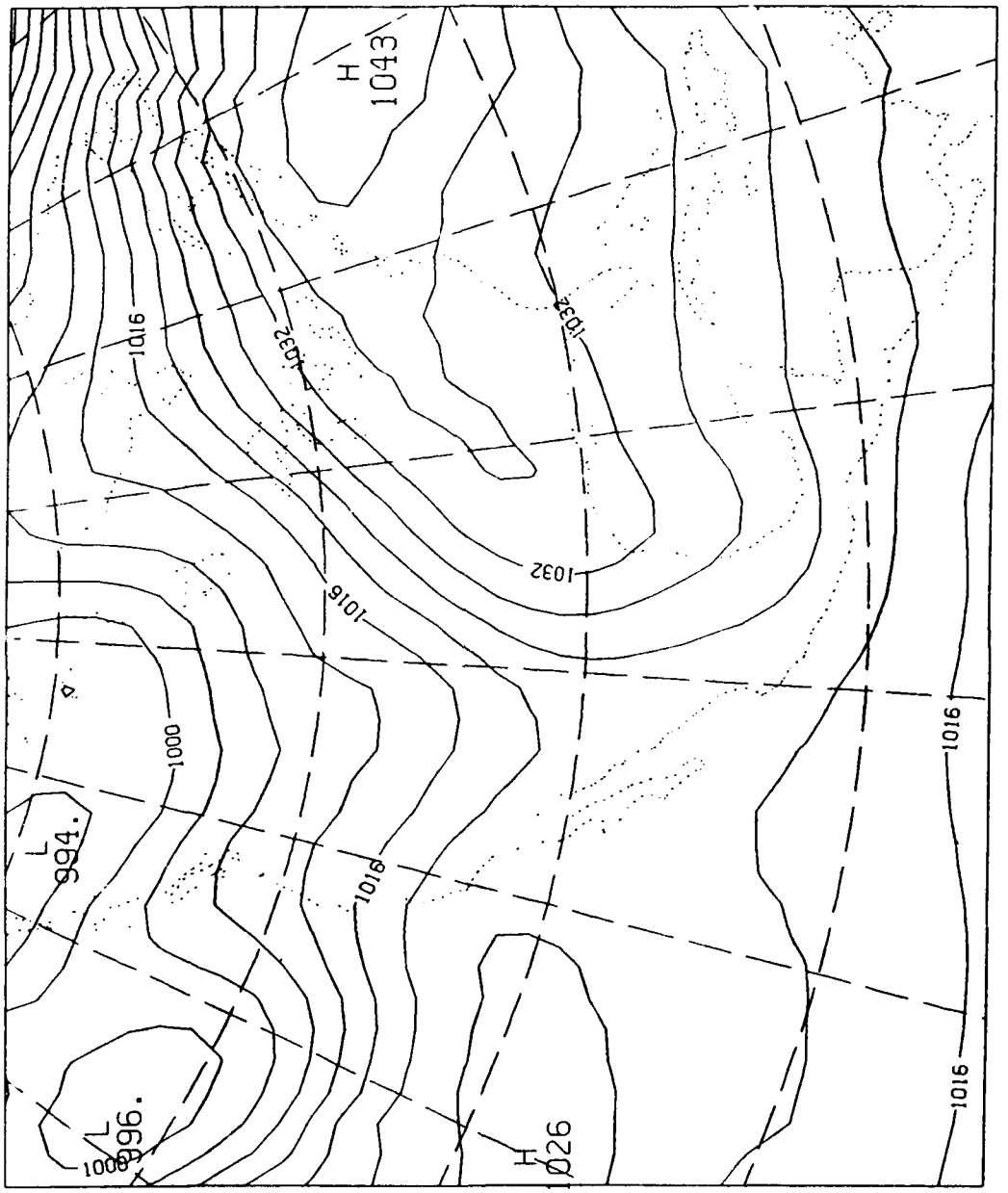



Fig. 7. Same as Fig. 6 but with Fourth-order Differencing and 1, 2 Smoothing.

not develop and in fact is less visible than in the previous case.

In an attempt to determine whether oversmoothing was the culprit in extinguishing the storm, an RLAM forecast similar to the one depicted in Fig. 6 was generated but with 5, 15 smoothing rather than 3, 9 smoothing. The results appear in Fig. 8 and appear very similar to those in Fig. 6, demonstrating that the Shapiro filter had very little to do with misforecasting the storm.

Resolution is another possible area of investigation, especially since the Presidents' Day storm was a product of small scale interactions (Bosart, 1981). To test whether increasing the resolution would have any effect, the spatial and temporal increments were halved. Because of limited computer storage, however, the domain size also had to be halved. Fig. 9 illustrates the new domain with a plot of the GSM 48 h sea-level pressure field. Fig. 10 shows a 48 h RLAM forecast over the smaller domain generated by second-order differencing and 3, 9 smoothing. The storm is still underpredicted and still too far east, although the higher resolution does allow for another closed contour and a lower central pressure.

Staggering the grid according to the Arakawa C-grid produced no real improvement, as was expected. An experiment with fourth-order differencing and a staggered grid produced a forecast very similar to the one depicted in Fig. 7.

A suspicion that perhaps some essential ingredient was missing from RLAM's initial conditions that was present in the global model led to an experiment in which the GSM 24 h forecast served as initial conditions for RLAM. The RLAM 24 h forecast could then be compared with the other 48 h

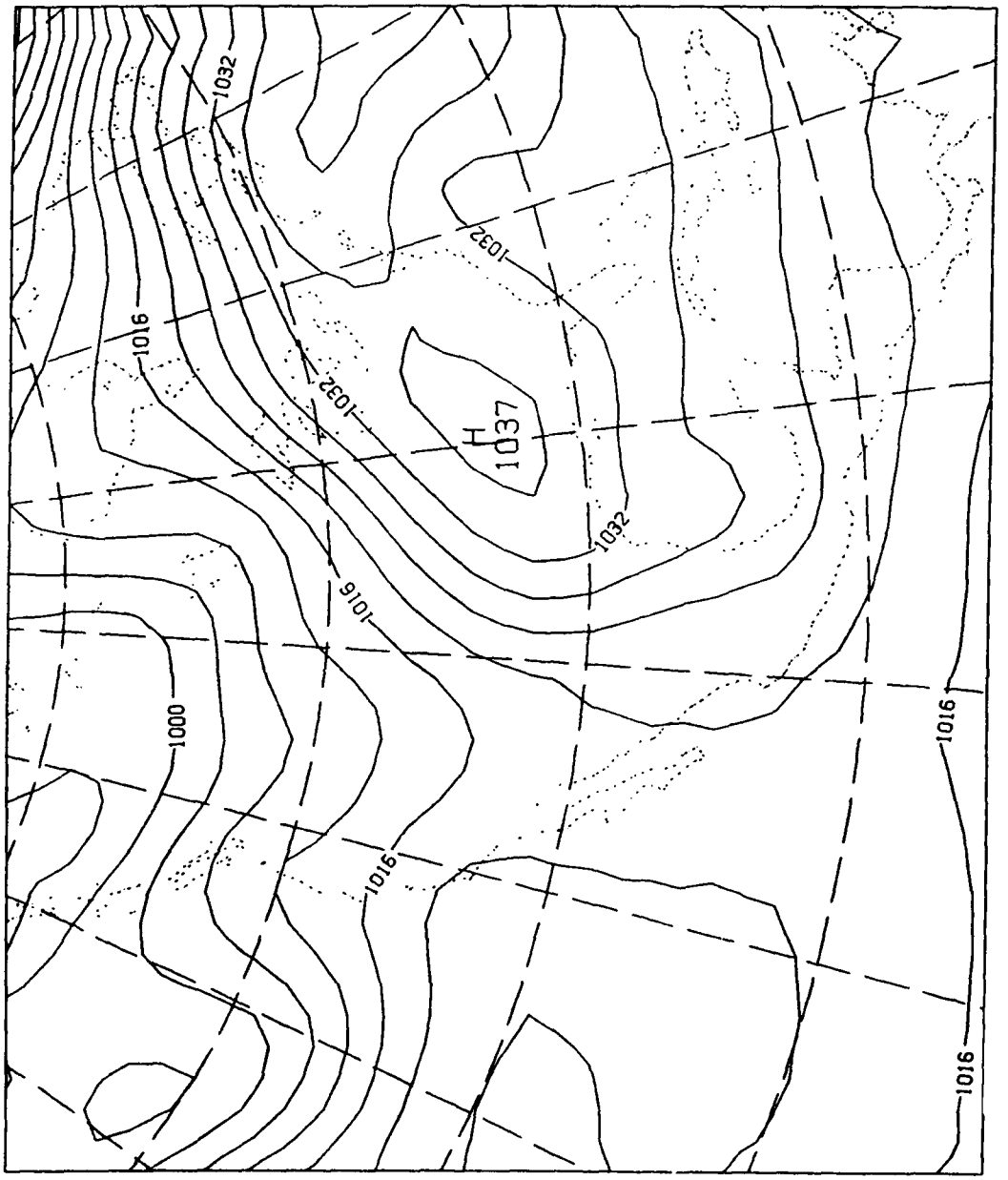



Fig. 8. Same as Fig. 6 but with 5, 15 Smoothing.

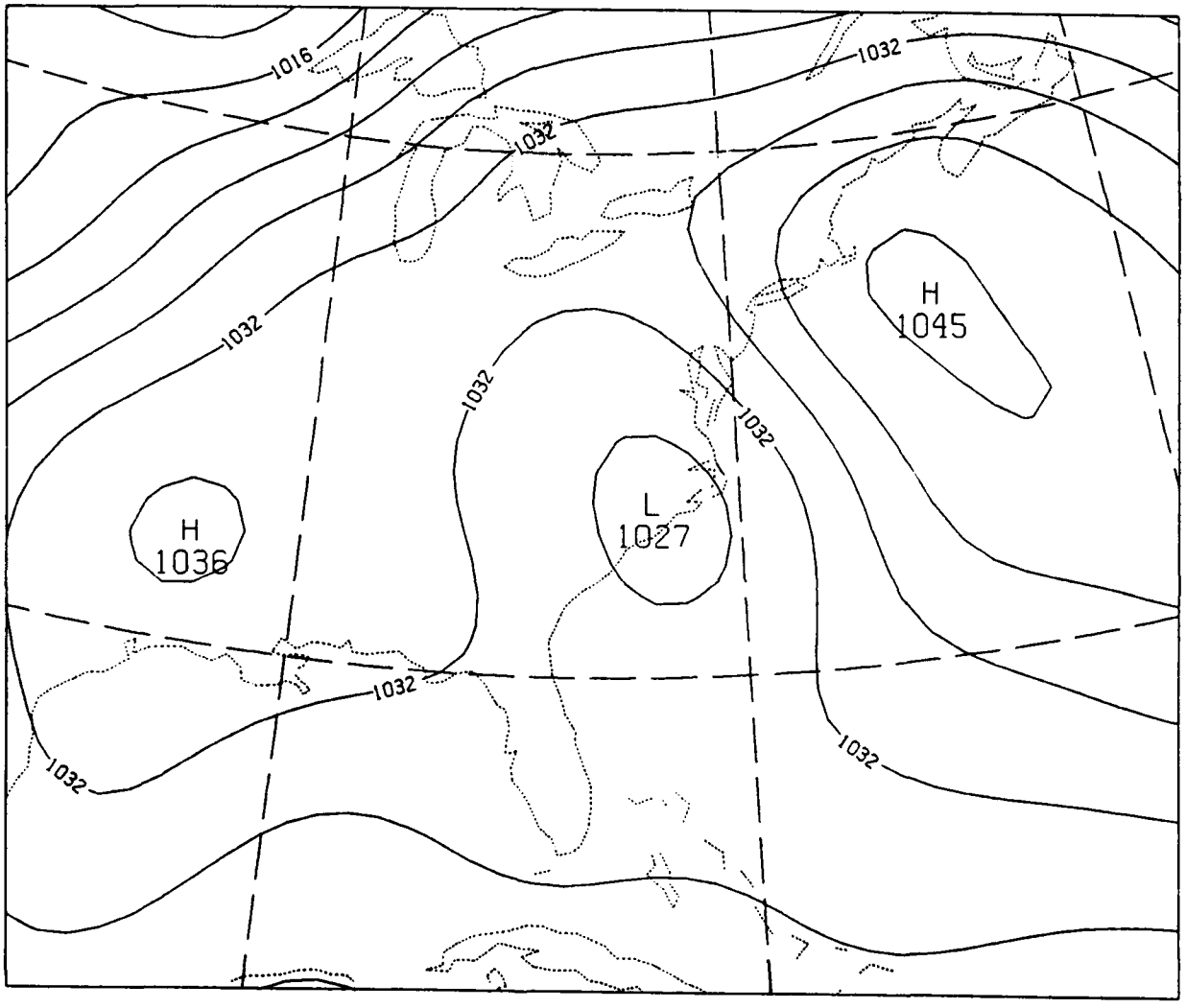



Fig. 9. GSM 48 h Forecast Verifying at Same Time as Fig. 5 but Magnified over Smaller Domain Centered at 35°N, 80°W.

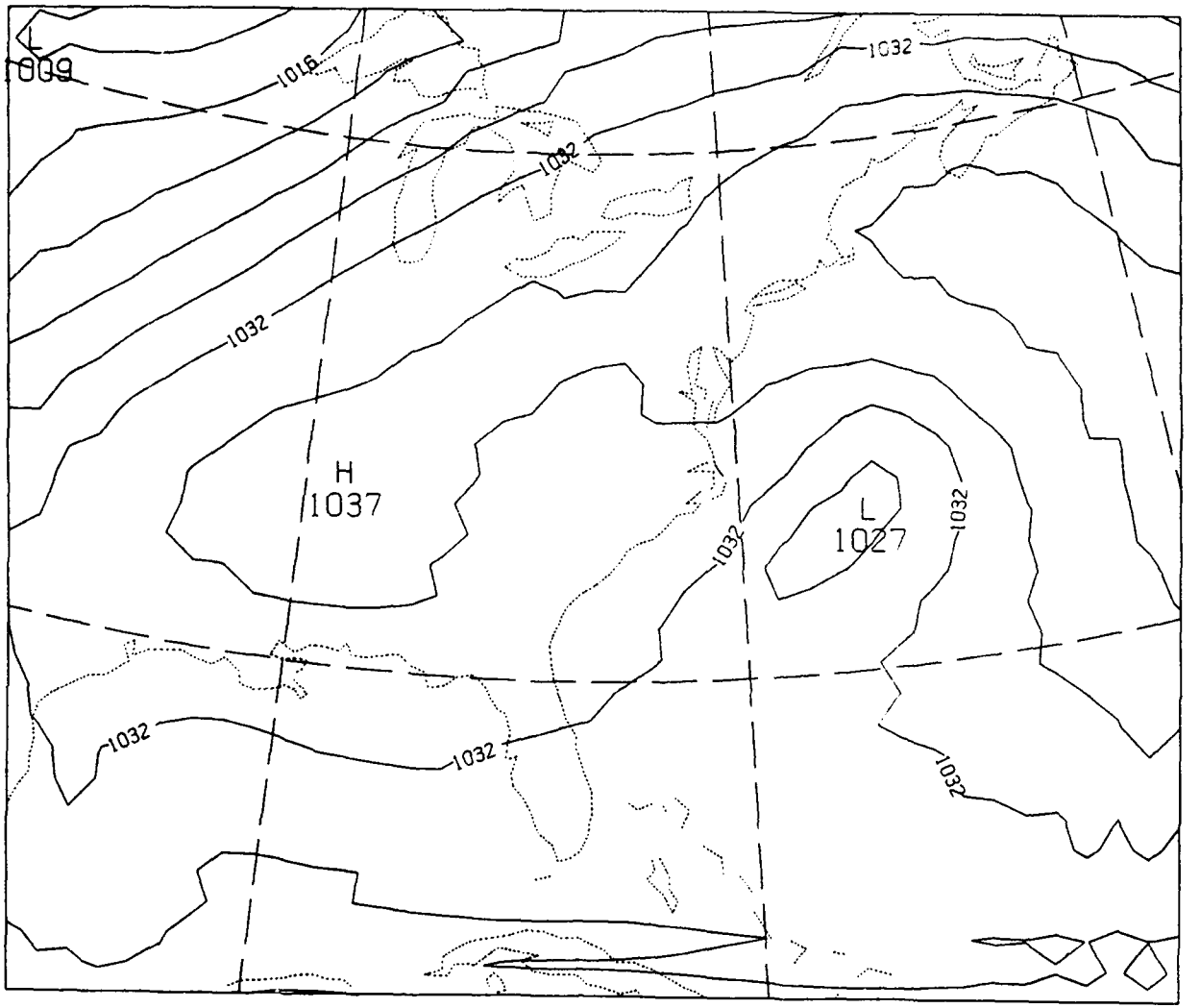



Fig. 10. Same as Fig. 6 but over Smaller Domain and over Grid with 1/4 Standard Mesh Resolution Centered at 35°N, 80°W.

forecasts. Figs. 11a and b result from a 24 h forecast starting with the GSM 24 h forecast verifying at 1200 GMT 18 February, with second-order differencing and 3, 9 and 5, 15 smoothing respectively. The two are rather similar but do show minor differences owing to the difference in smoothing frequency. Despite the "head start," however, no real improvement is noted in the development of the storm.

With none of the numerics able to improve the forecast of the storm to any degree, the importance of the physical parameterization was put to the test. First, RLAM's physics were all turned off, including boundary layer fluxes, Kuo convection, and large scale convective adjustments. Second, the GSM was rerun with the older version of physics which scales evaporation more like the packages on RWM and RLAM. The results of the first experiment appear in Fig. 12. The differences are striking. Without the physics there is not even a suggestion of a storm, while the high over the Great Lakes has not moved offshore or split and has gained in intensity rather than diminishing as observed. On the other hand, the GSM forecast with the older physics shown in Fig. 13 seems to be an improvement. Although still 12 mb too shallow, the storm is more intense at the expense of the offshore high, as observed. This again indicates the sensitivity of the situation to the physics involved and the need for accurate physical parameterization. It is still unresolved, however, as to why the GSM fares better than RLAM even when their physics are similar.

The RWM has a physics package that is identical to RLAM's but its numerics are completely different. Besides employing Lagrangian advection, it has strong diffusion on its boundaries and divergence damping at each time step. Boundary conditions are derived by assuming a constant tendency at its outermost rows for 6 h. Like RLAM the resolution, location, map projection, vertical structure,

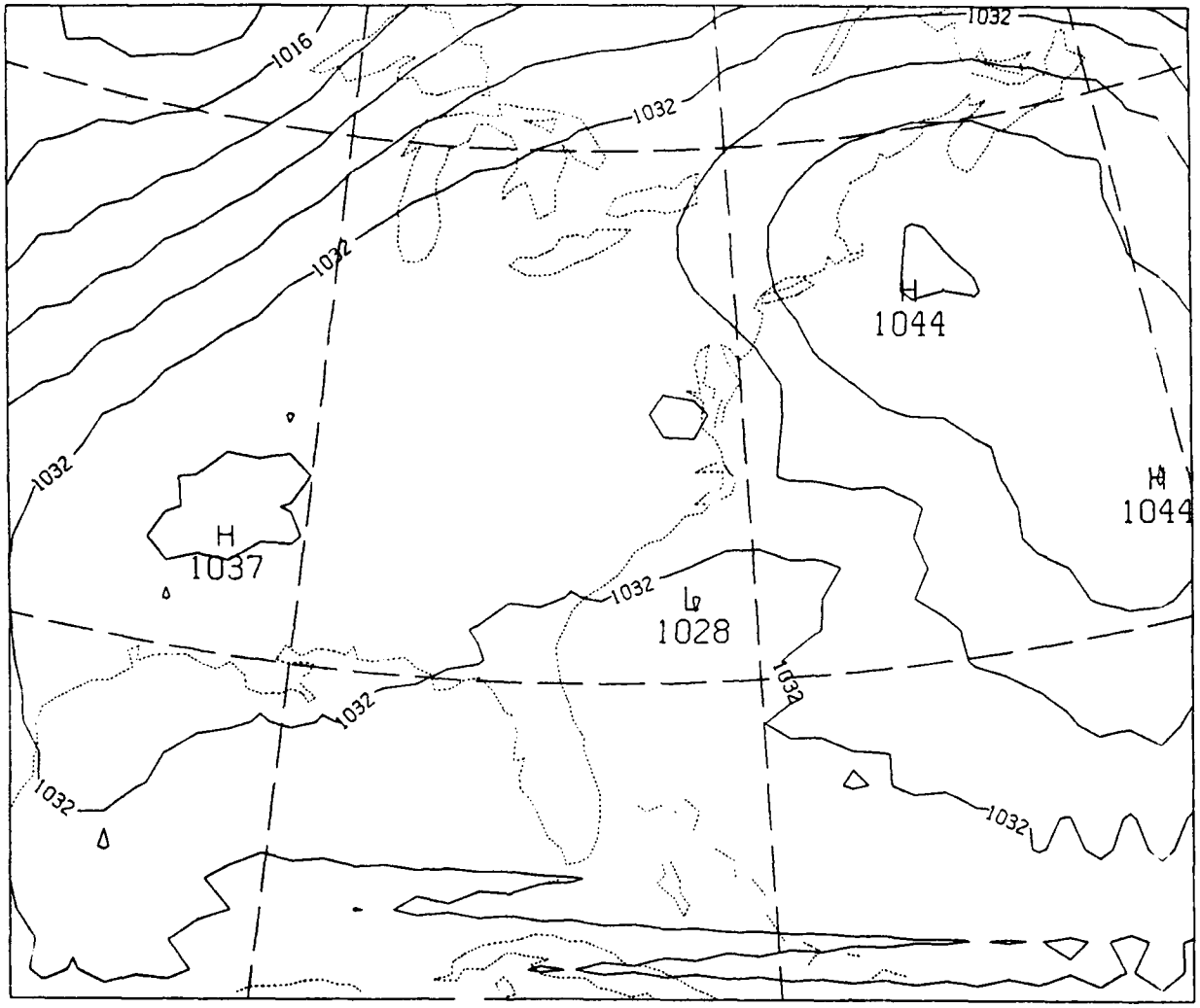



Fig. 11a. RLAM 24 h Forecast of Sea-level Pressure at Same Time and Region as Fig. 10 but Initialized with GSM 24 h Forecast Verifying at 1200 GMT 18 February 1979. Forecast Produced with Same Parameters as in Fig. 10 but with 3, 9 Smoothing.

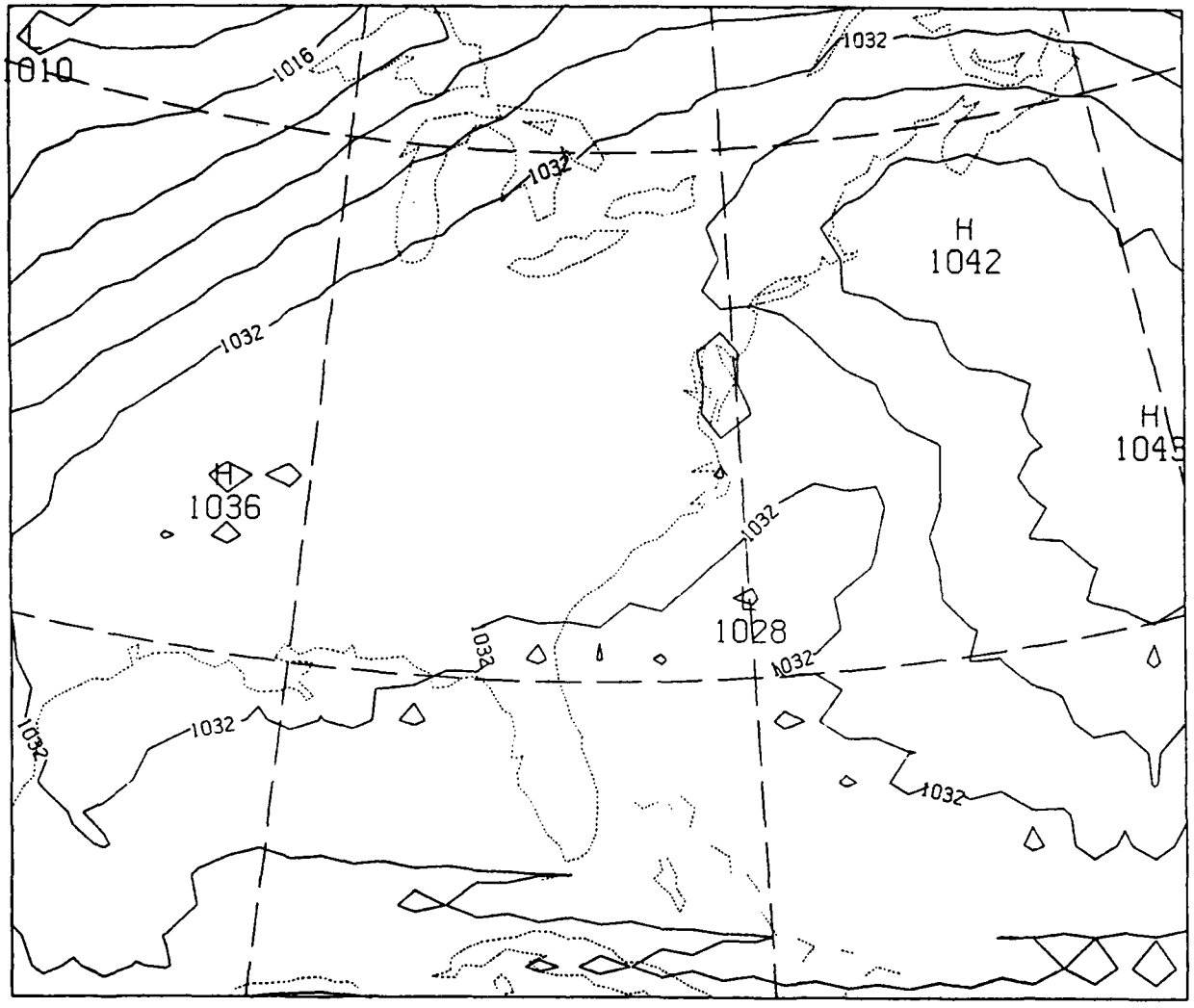



Fig. 11b. RLAM 24 h Forecast of Sea-level Pressure at Same Time and Region as Fig. 10 but Initialized with GSM 24 h Forecast Verifying at 1200 GMT 18 February 1979. Forecast Produced with Same Parameters as in Fig. 10 but with 5, 15 Smoothing.

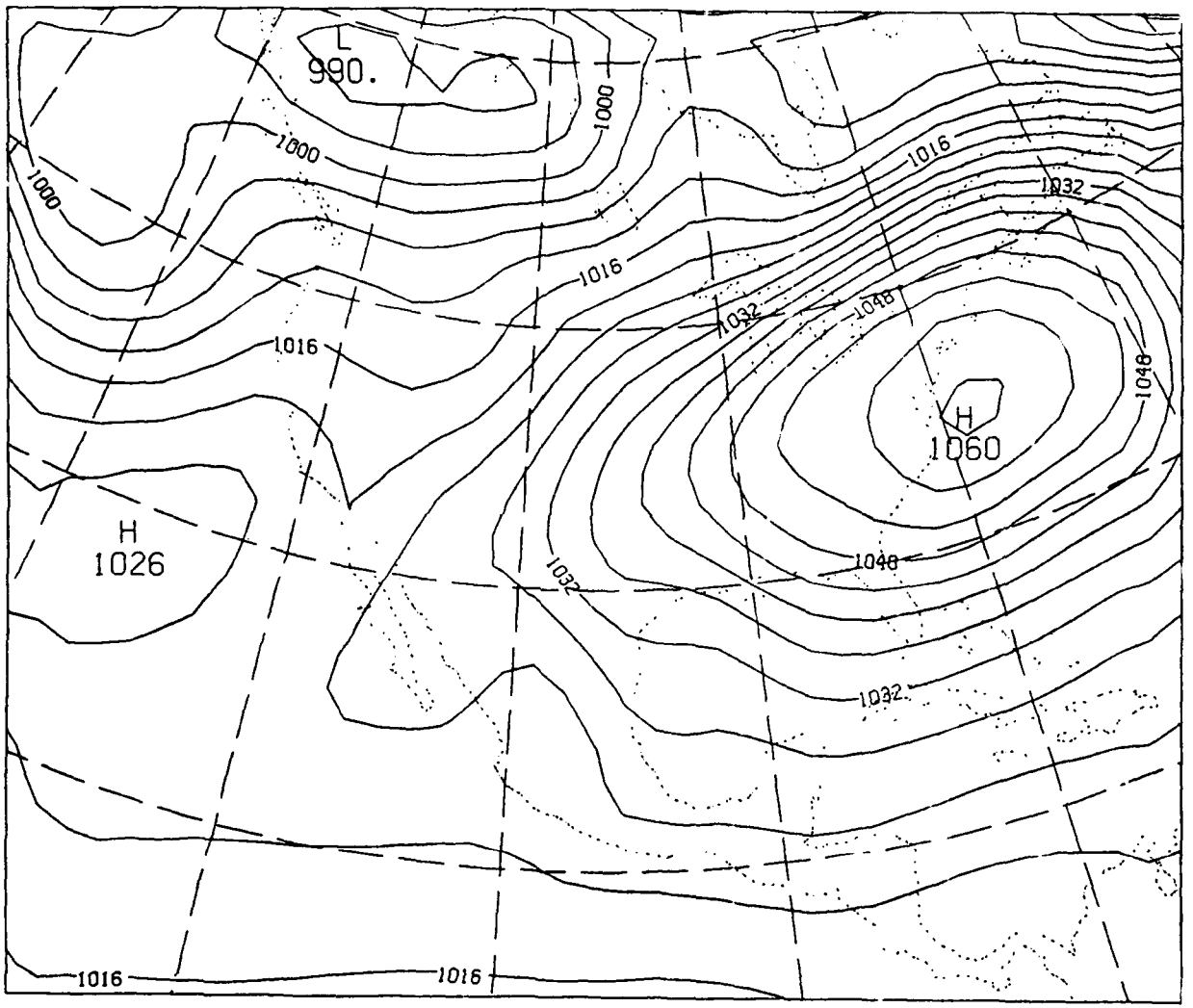



Fig. 12. Same as Fig. 6 but with All Physical Parameterizations Set to Zero.

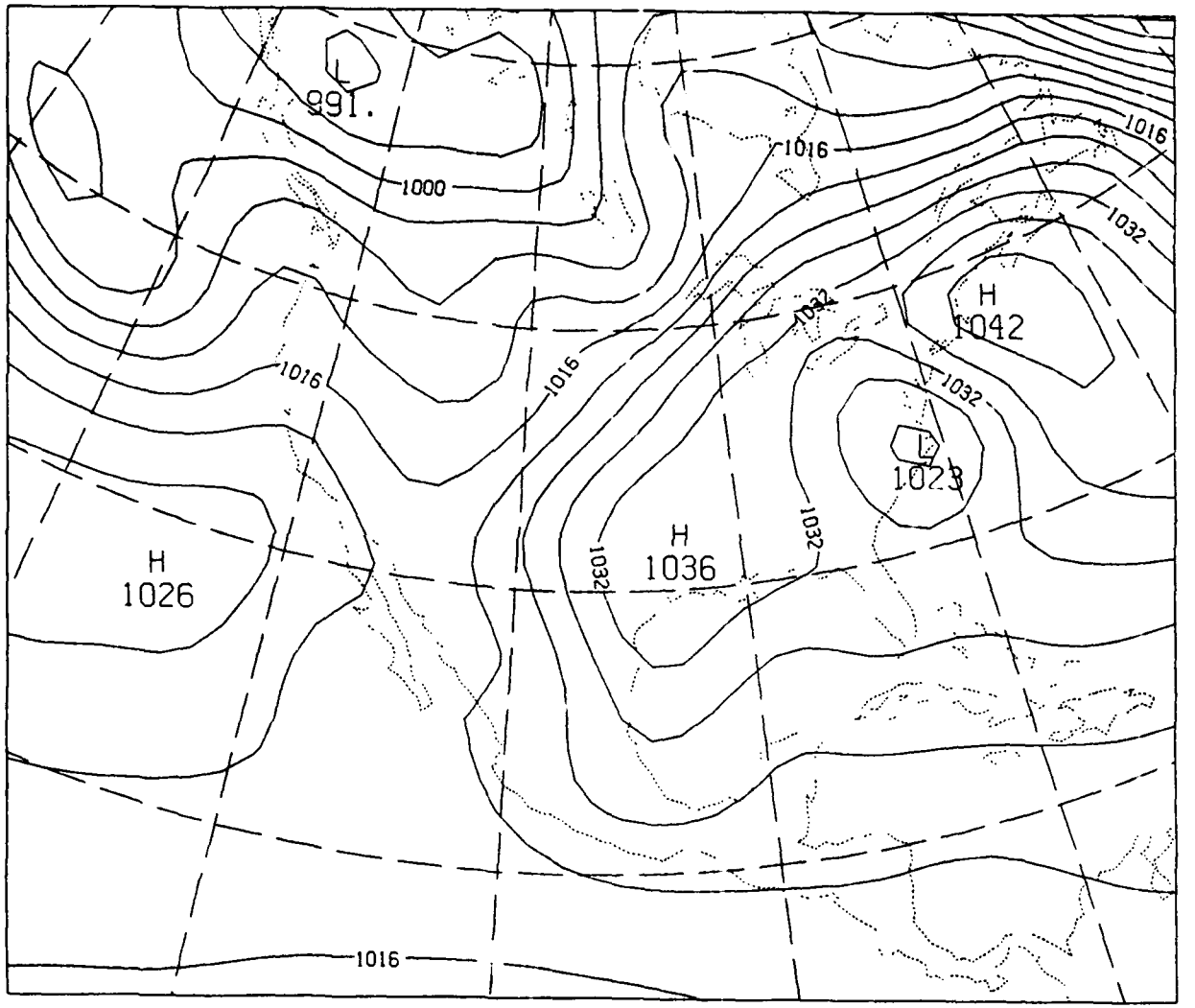



Fig. 13. Same as Fig. 5b but with Older Version of Physical Parameterization.

time step, forecast length, and output times are all optional and specifiable. The RWM and RLAM file structures are not the same but a simple program to convert one to the other was fashioned to allow RLAM data to be used in the RWM and vice-versa. Thus it was possible to run a 48 h forecast with the same data from 1200 GMT 17 February as were used by RLAM. Fig. 14a is the resultant forecast from the RWM for the 1/2 mesh resolution and Fig. 14b is for the 1/4 mesh resolution. With the fine resolution there is only a hint of something forming, but, as with RLAM, it is too far east and south. The coarse resolution is almost as poor as RLAM without physics, as shown in Fig. 12. As a possible explanation, one can single out the heavy smoothing and divergence damping at each time step. Attempts to decrease the frequency of smoothing, however, led to unstable solutions.

## 5. Conclusions

The experiments concerning the Presidents' Day storm are intriguing if not conclusive. They indicate, to no surprise, that proper numerical schemes are a necessary but not sufficient condition for accurate forecast of mesoscale cyclones. Proper physical parameterization of the boundary layer-moisture-atmosphere interactions is also a necessary ingredient. It is not clear, however, why a global model with a 30-wavenumber resolution should predict better than a highly resolved finite difference model with basically the same physical parameterization. One could suggest that wave motion in the upper atmosphere is more realistically captured in a spectral model than in a finite-difference model and the interaction between upper atmosphere and surface may have been a crucial factor. But such a conjecture will not be proved without many more tests of the kind presented here.

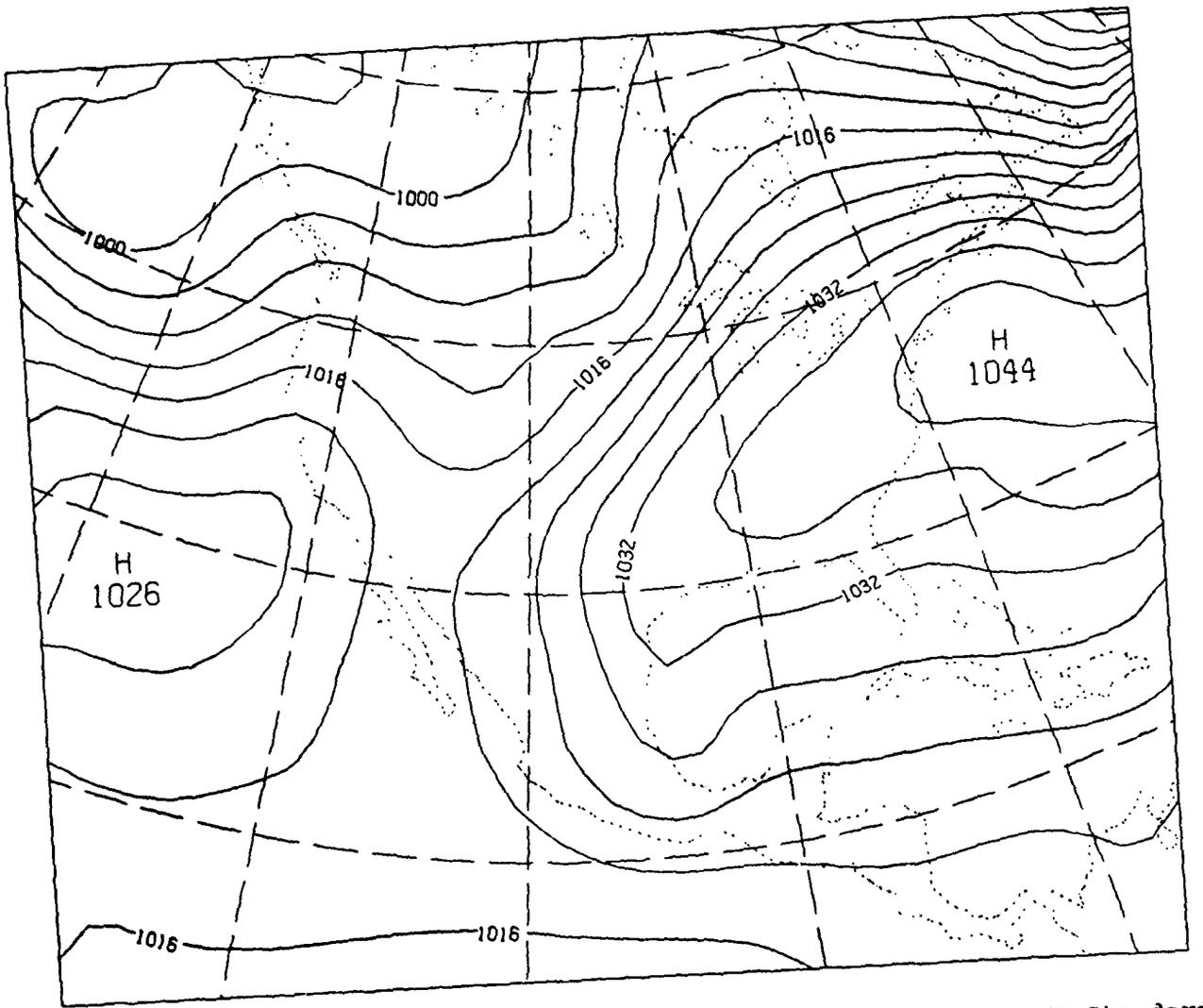



Fig. 14a. RWM 48 h Forecast Verifying at Same Time as Fig. 4 with 1/2 Standard Mesh Resolution.

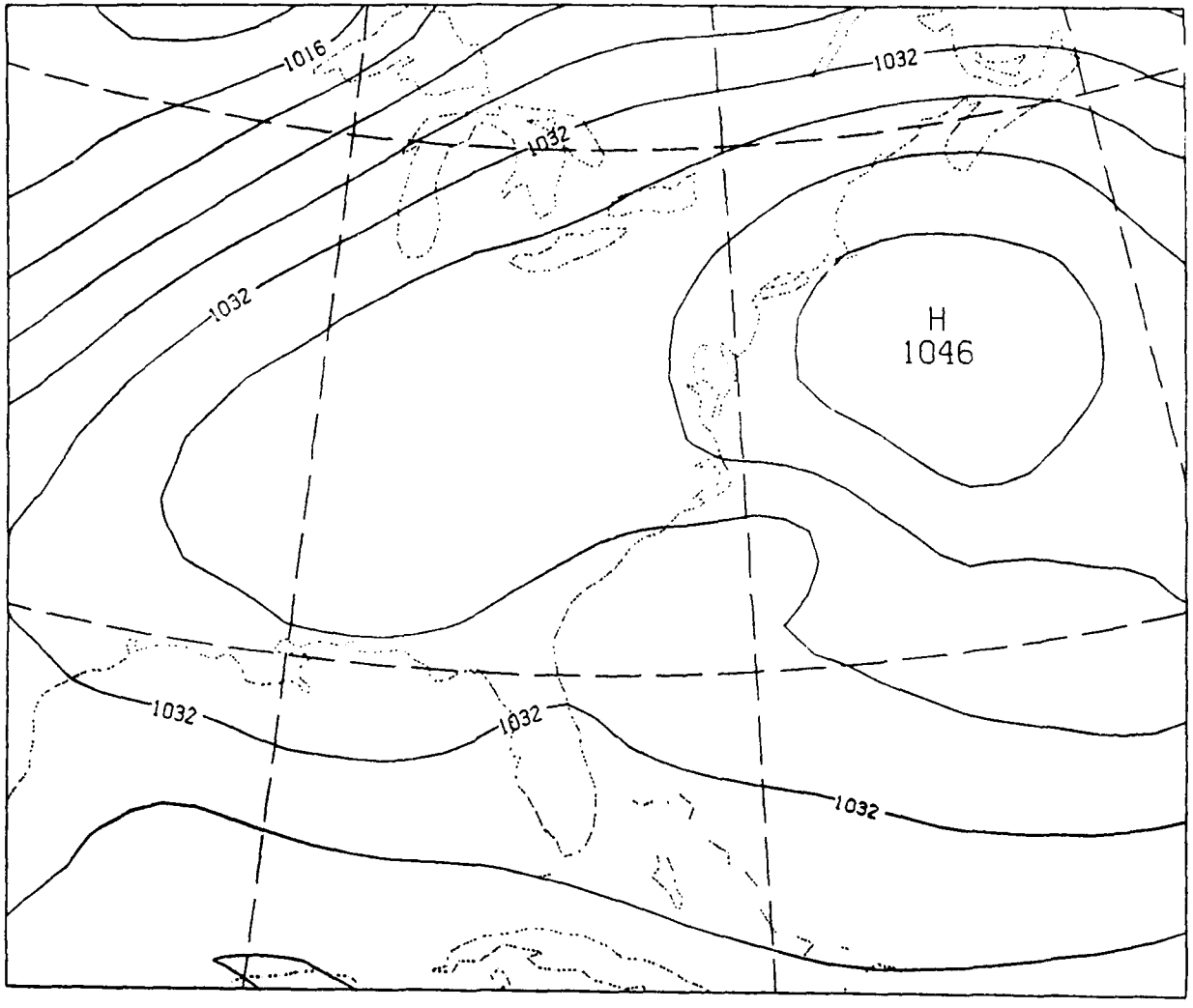



Fig. 14b. RWM 48 h Forecast Verifying at Same Time as Fig. 4 with 1/4 Standard Mesh Resolution.

With regard to the performance of RLAM, its value as a research tool has been demonstrated. It was possible to isolate various factors, such as smoothing frequency, resolution, or horizontal differencing, simply by setting some parameters and re-running the model. In this experiment, RLAM helped us to understand the role such factors play in the prediction of a complex synoptic situation. Further testing of this and similar situations could also provide insights into the importance of numerical and physical parameterizations for cloud and precipitation forecasts.

## 6. References


Arakawa, A., and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model. Methods in Computational Physics, Vol. 17, J. Chang, ed., Academic Press, NY, 173-265.

Bosart, L. F., 1981: The Presidents' Day snowstorm of 18-19 February 1979: A subsynoptic-scale event. Mon. Wea. Rev., 109, 1542-1566.

Bourke, W., 1974: A multi-level spectral model. I. Formulations and hemispheric integrations. Mon. Wea. Rev, 102, 687-701.

Brenner, S., 1988: A comparison of various numerical solutions of the hydrostatic equation. Preprints, Eighth Conference on Numerical Weather Prediction, Baltimore, MD; Amer. Meteor. Soc., Boston, 438-441.

_____, C.-H. Yang, and K. Mitchell, 1984: The AFGL Global Spectral Model: Expanded Resolution Baseline Version. AFGL-TR-84-0308, Air Force Geophysics Laboratory, Hanscom AFB, ADA160370.

Davies, H. C., 1976: A lateral boundary formulation for multi-level prediction models. Quart. J. Roy. Meteorol. Soc., 102, 405-418.

Fritsch, J. M., E. L. Magaziner, and C. F. Chappell, 1980: Analytical initialization for three-dimensional models. J. Appl. Meteor., 19, 809-818.

Gerlach, A. M., ed., 1984: Objective Analysis and Prediction Techniques - 1984. AFGL-TR-84-0328, Contract F19628-82-C-0023, Systems and Applied Sciences Corporation, ADA160377.

_____, 1985: Objective Analysis and Prediction Techniques - 1985. AFGL-TR-86-0002, Contract F19628-82-C-0023, SASC Technologies, Inc., ADA169746.

_____, 1986: Objective Analysis and Prediction Techniques - Final Report. AFGL-TR-87-0013, Contract F19628-82-C-0023, ST Systems Corporation, ADA183450.

Halberstam, I. M., S.-L. Tung, and C. Johnson, 1988: Development and application of a modular limited area model. Preprints, Eighth Conference on Numerical Weather Prediction, Baltimore, MD. Amer. Meteor. Soc., Boston, 328-330.

Mathur, M. B., 1983: A quasi-Lagrangian regional model designed for operational weather prediction. Mon. Wea. Rev., 111, 2087-2098.

McGregor, L., L. M. Leslie, and D. J. Gauntlett, 1978: The ANMRC limited-area model: Consolidated formulation and operational results. Mon. Wea. Rev., 106, 427-438.

Merilees, I. E., 1973: An alternative scheme for the summation of a series of spherical harmonics. J. Appl. Meteor., 12, 224-227.

Perkey, D. J., and C. W. Kreitzberg, 1976: A time-dependent lateral boundary scheme for limited-area primitive equation models. Mon. Wea. Rev., 104, 744-755.

Shapiro, R., 1970: Smoothing, filtering, and boundary effects. Rev. Geophys. Space Phys., 8, 359-387.

Tung, S.-L., I. Halberstam, and C. Johnson, 1987: Development of an improved modular relocatable limited area model (RLAM), Short- and Medium-Range Numerical Weather Prediction. Collection of papers presented at the WMO/IUGG NWP Symposium, Tokyo, 4-8 August 1986; special edition of J. Meteor. Soc. Japan, 485-493.

# II. GLOBAL CLOUD CHARACTERIZATION FROM DIGITAL SATELLITE DATA

## A. Ground Truth for Objective Evaluation of Automated Nephanalysis

### 1. Introduction

During the contract period an interactive technique was developed to produce an optimal cloud analysis from human interpretation of computer enhanced satellite imagery. An analysis produced by this technique can be used as ground truth to evaluate the accuracy of automated cloud analysis programs such as the Air Force RTNEPH model. The technique was implemented as a series of routines on the AFGL Interactive Meteorological System (AIMS) and can be run on either of the two available image processing workstations. Collocated imagery from up to three different meteorological satellite systems can be used to produce the analysis, although data from a single sensor are sufficient. The technique requires a trained analyst to examine all available imagery for a scene under consideration. Standard image processing techniques are used to enhance interactively various features in an image to aid in cloud boundary and extent determinations. When the analyst has identified a cloud boundary location, this information is transferred to a "cloud truth" database (CTDB) by a local threshold blanking technique.

Human observers are often considered to be superior to computer algorithms for interpretation of satellite imagery because of their skill in recognizing spatial patterns. Since, however, temporal and spectral patterns are much more difficult for people to detect and interpret, some form of computer processing is often required to present these

patterns in a different, more comprehensible format. Four processes were identified in which manual interpretation of imagery could benefit from computer preprocessing or enhancement:

1) Area estimates
2) Detection of small scale (pixel or sub-pixel) features
3) Interpretation of multispectral data
4) Detection of a large number of grayshade levels.

The primary focus of the work described in this section was to improve an analyst's ability to interpret and categorize satellite imagery. Image processing algorithms were used to modify or enhance images as an aid to this effort. Algorithms were selected and implemented to improve image interpretation generally in the area of pattern recognition and specifically in the four areas listed above.

## 2. Image Processing on the AIMS

AIMS was developed through a joint AFGL-STX effort to support research in satellite meteorology at AFGL. A description of the system can be found in Gustafson et al. (1987). The two image processing workstations that are part of this system are built around ADAGE 3000 image display systems. At the beginning of the work described in this section there was no comprehensive, general purpose library of primitive routines available to control the ADAGE systems. The first task was to collect whatever routines did exist, generalize them, and place them into a universally accessible structure. From that base a complete series of primitives was written to perform the control functions necessary to use the ADAGE in this project and also to meet many of the requirements of the user community at large. Appendix A contains a brief functional

description and the call syntax of these utility routines. These routines are located in a library called ADAGE.OLB in the directory pointed to by the AIMS system logical ADG_RTL. Using routines in ADAGE.OLB, the workstation can be programmed to display images up to 24 bits deep, with an additional 8 bits of overlay, using any combination of the 32 available bit planes. Digital enhancement of imagery is controlled by four 8-bit color look up tables (LUT), one for each 8 bits of input. Using these LUTs, up to four 8-bit images from, say, four different sensors can be enhanced separately and displayed simultaneously in different quadrants on the monitor. Other primitives perform generic functions such as pan, scroll, and zoom; coordinate conversion; erase and fill; cursor position and pick; generate and plot histograms; initialize the LUTs; remap the bit planes and establish overlay priority; and toggle individual bit planes or images on and off. Most of these control functions have been combined into a single ADAGE utility program known as ADG. A description of the program functions can be obtained by typing in ADG/HELP at the DCL prompt on the ARCIP node of AIMS.

## 3. Image Display Techniques

Interactive cloud detection is performed by examining satellite-derived images of quantized infrared brightness temperatures or visible brightness counts. To minimize the impact of different sensor viewing geometry and scan characteristics, all images can be registered to a common grid projection. Since this work is being performed in conjunction with the AFGL RDNEPH project to improve the Air Force operational RTNEPH model, the AFGWC SGDB polar stereographic projection (i.e., 5.95 km resolution true at 60 degrees latitude with a base longitude of 10 degrees east) is used as the standard projection. However, remapping of the imagery data to the SGDB projection is

normally not done since this inevitably causes some distortion of the spatial and spectral characteristics of the data. Imagery can be displayed in any of three ways (described below). Depending on what type of data is to be displayed and how the analyst wants to use the data, either a single display or some combination of display techniques can be used.

a. Monochrome Display

Each of the three electron guns on an ADAGE display system has an 8-bit dynamic range (i.e., 256 different intensity levels). To facilitate display of single channel satellite imagery, the brightness temperature or visible count data are scaled to an 8-bit value and often displayed as grayshade counts ranging linearly from 0 (black) to 255 (white). Intermediate values appear as shades of gray. This produces the familiar satellite image where cold bright clouds appear white and warm dark backgrounds appear dark. From this basic presentation, images are processed to enhance any cloud features that may be present. Enhancement processing can be either radiometric or geometric.

Radiometric processing is typically used to increase the contrast over an image or to highlight specific features. Examples of this type of processing include linear contrast stretching, piece-wise linear contrast stretching, and histogram equalization. Linear and piece-wise linear contrast stretching involve expanding some portion of the grayshade range of the image to cover the full dynamic range of the ADAGE display device. This is done by generating a histogram of the grayshade values from the image and using the histogram endpoints as the limits to the enhancement curve. An enhancement is then produced that is made up of either linear or piece-wise linear segments over the histogram range. Histogram equalization is a

method of providing approximately equal contrast over all brightness levels in an image. This is accomplished by modifying the histogram of the image so that it is essentially flat. The procedure involves normalizing the histogram at each grayshade level by the cumulative frequency distribution at that level.

Geometric enhancements are applied at each pixel by using information from surrounding pixels to modify the grayshade value of the target pixel. These operations are commonly called "convolution operations" because of their similarity to convolution in signal processing theory. The process is performed numerically by replacing a target pixel value with a weighted sum of itself and the neighboring pixels; this is repeated for every pixel in the image. The weighting factors are called the convolution kernel. By using different kernels, an image can be enhanced to remove noise, smooth or blur out edges, sharpen the contrast at edges, or locate specific features in the image. While the convolution operation is computationally expensive, it is easily implemented on the fast bit-slice processor that is part of the ADAGE display subsystem. A typical 3 x 3 convolution kernel can be run on a 512 x 512 8-bit image in about two seconds.

b. False Color Multispectral Display

Multispectral analysis is very difficult for an analyst to do visually. Most presentations of multispectral data consist of several separate images, one for each spectral band. The analyst is faced with the problem of somehow recognizing a feature signature by integrating the several images in his head. Simple image processing techniques such as image subtraction and addition can be useful but unless the situation is very well behaved interpretation can still be problematic. However, using a false color technique that

was adapted to AVHRR data by d'Entremont and Thomason (1987) at AFGL, it is possible to do a visual qualitative analysis of two- or three-channel multispectral data on the AIMS. The procedure involves controlling the intensity of the three electron guns of the display system separately using scaled grayshade values from the different sensor channels. Each of the electron guns produces light at a different color: red, green, or blue. By combining the output from all three guns on the screen, a composite false color multispectral image is produced. Interpretation of this type of image is relatively straightforward. Regions where the response in all three channels is approximately equal will appear as a shade of gray. However, where the response of one channel varies from the other two (e.g., due to a different emissivity of the observed surface at the different wavelengths) a predominant color appears. For example, the emissivity of a large water droplet cloud is significantly less at AVHRR channel 3 (3.7 microns) than at channels 4 (11 microns) and 5 (12 microns). Because of this, scaled brightness temperatures from channel 3 will be colder (i.e., brighter) and the color gun being driven by those data will dominate the other two colors. This results in a distinctive signature on a false color image. False color images can be produced by combining any two or three channels of data from the DMSP OLS, the SSM/I, the NOAA AVHRR, or the GOES VAS.

c. Four Quadrant Multi-image Display

Collocated imagery from up to four different sensors can be displayed simultaneously in different quadrants on the display screen. A separate LUT can be assigned to each quadrant, allowing different enhancement or convolution operations to be performed on each. This type of display is useful for maintaining a reference image while performing different operations on a duplicate target image.

## 4. Interactive Cloud Analysis Technique

Actual cloud analysis is performed at an AIMS image processing workstation by a meteorologist who is familiar with the characteristics of the different satellite sensors being used. Fast, interactive routines have been written that implement the various enhancement functions described above. Radiometric enhancements are performed by manipulating the four color look up tables in the display systems and, since LUT operations are performed at frame refresh rates (1/30th sec), the results are virtually instantaneous. Convolution operations are destructive to the image they operate on. To preserve the original data, convolution results are usually placed in another area of frame buffer memory. The 4 MB of memory available on each workstation is normally adequate to store convolution results locally, thus eliminating the need for time consuming disk storage procedures. This helps to keep the process interactive since convolutions can be performed and displayed at the speed of the bit-slice processor. Multispectral displays are simply a matter of loading the imagery into the frame buffer and programming the control registers for a false color display.

Once the analyst has the data assembled, a region of interest is selected from one or more of the satellite images. These regions are selected to contain only a small number of features and may be further subdivided at any time as required by the analyst. Collocated imagery from up to four different sensors is displayed in different quadrants on the display screen, each image controlled by a separate LUT. Different enhancement or convolution operations may be performed on each quadrant. While the images from different sensors are selected so that they are collocated in space, there will normally be some temporal displacement that will

have to be filtered by the person in the loop. Once the analyst has detected a cloud feature and decided on the boundary locations, the next problem is transferring that information into the database. Techniques available to do this typically involve visually estimating the extent of cloud cover or possibly outlining the region on the screen with some type of graphic device. However, these techniques may produce poor approximations of the actual extent and location of the cloud feature that the analyst has identified. A more accurate method is threshold blanking. This requires that an 8-bit monochrome image be selected as a target image and be loaded and displayed as a 24-bit grayshade image (i.e., the 8-bit image is mapped equally to each color electron gun). Usually reference images are also loaded into other quadrants of the display screen. A threshold level is interactively raised or lowered over the target image. The level of the threshold is displayed by altering the LUT for grayshades below the threshold level such that one of the color guns (usually red) has an intensity somewhat less than the other two. This results in a translucent color background being established around the cloud feature. While the boundary of the threshold is evident to the analyst, the information below the threshold level is still easily discernible. The threshold is raised and lowered until the analyst is satisfied that the cloud feature is accurately delineated. At that point the location and extent information is digitally transferred to the "cloud truth" database and tagged with valid time, satellite sensor, and estimated cloud type.

## 5. Summary

An interactive technique has been developed to perform manual cloud analysis from satellite imagery. Trained analysts use standard image processing techniques to improve their ability to interpret and detect cloud features in the

imagery. Collocated images from different satellite sensors are used to generate the analysis. Multiple images can be displayed and enhanced simultaneously on an image processing workstation. The analyst can selectively display and enhance the different images to aid in the subjective determination of where the cloud boundaries lie. Identified cloud features are transferred to a "cloud truth" database through a threshold blanking technique. The available image processing functions aid the analyst in overcoming four problem areas identified with manual interpretation of satellite imagery. These are (1) area estimates of cloud cover are made directly by the computer through threshold blanking of clear areas, (2) the inability of most observers to discern a large number of grayshades is reduced through various contrast enhancement utilities, (3) interpretation of multispectral data is aided by 24-bit false color displays, and (4) small scale features are enhanced by edge detection convolution algorithms that enhance high frequency features in the data.

## 6. References

d'Entremont, R. P., and L. W. Thomason, 1987: Interpreting meteorological satellite images using a color-composite technique. Bull. Amer. Meteor. Soc., 68, 762-768.

Gustafson, G., D. Roberts, C. Ivaldi, R. Schechter, T. Kleespies, K. Hardy, R. d'Entremont, G. Felde, and R. Lynch, 1987: The AFGL Interactive Meteorological System. Preprints, Third International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, New Orleans; Amer. Meteor. Soc., Boston, 151-154.

## Appendix A - Adage Run-time Library Utilities

NOTE: When linking in any of these routines the adage option file must be included in the LINK command stream:

```
LINK file_name,ADG_SYSTEM:ADAGE/OPT
```

ADG_CHECK_CHANNEL(ICH,ONOFF)

Checks the main crossbar for 'channel correspondence.' Returns array of 4 numbers which indicate to which channels the crossbar is set. 0=R,1=G, 2=B, 3=A, channel is -1=nonsense. Also indicates whether all planes in the channel are on.
ICH(4) (I*4) returned by reference. Location in the array refers to the channel (1)=R, (2)=B, etc.
*ONOFF*(4) (I*4) 1=all planes on, 0=at least one plane off.

ADG_CONTRAST_STRETCH (LO, HI)

Do a linear contrast enhancement between the specified pixel levels. Display pixels outside limits as either 0 or 255 for above or below the limits respectively.
LO (I*4, by value)
Lower pixel level limit.
HI (I*4, by value)
Upper pixel level limit

ADG_CONVERT_CV (X, Y)

Convert from cursor (screen coordinates plus cursor offset) to viewport coordinates.
X (I*4, by reference)
Passed as cursor x coord, returned as viewport x coord.
Y (I*4, by reference)
Passed as cursor y coord, returned as viewport y coord.

ADG_CONVERT_CW (X, Y)

Convert from cursor window (display memory) coordinates.
X (I*4, by reference)
Passed as cursor x coord, returned as window x coord.
Y (I*4, by reference)
Passed as cursor y coord, returned as window y coord.

ADG_CONVERT_VC (X, Y)

Convert from viewport to cursor coordinates.
X (I*4, by reference)
Passed as viewport x coord, returned as cursor

x coord.
Y (I*4, by reference)
Passed as viewport y coord, returned as cursor y coord.

ADG_CONVERT_WC (X, Y)
Convert from window to cursor coordinates.
X (I*4, by reference)
Passed as window x coord, returned as cursor x coord.
Y (I*4, by reference)
Passed as window y coord, returned as cursor y coord.

ADG_CONVERT_VW (X, Y)
Convert from viewport to window coordinates, includes zoom factor.
X (I*4, by reference)
Passed as viewport x coord, returned as window x coord.
Y (I*4, by reference)
Passed as viewport y coord, returned as window y coord.

ADG_CONVERT_WV (X, Y)
Convert from window to viewport coordinate, includes zoom factor.
X (I*4, by reference)
Passed as window x coord, returned as viewport x coord.
Y (I*4, by reference)
Passed as window y coord, returned as viewport y coord.

ADG_CURLOC(X, Y, VALUE)
Return cursor hotspot display memory (i.e., window) coordinates, and also the 32-bit value in memory underneath the cursor hotspot.
X (I*4, by reference)
Returns the window x coord ("column").
Y (I*4, by reference)
Returns the window y coord ("row").
VALUE (I*4, by reference)
Returns the 32-bit value in window memory at location X,Y.

ADG_DONE_BPS()
Set the done flag in scratch pad memory.

ADG_ERASE (X, Y, MASK)
Do a fast software erase over the specified region for the specified bit planes.
X (I*4 array of 2 elements, by reference)
UL and LR x window coordinates of image area to

erase, will be truncated to even multiples of 16 for lores or 32 for hires.

Y (I*4 array of 2 elements, by reference)
UL and LR y window coordinates of image area to erase.

MASK (I*4, by value)
Write mask set to bit planes that are to be erased.

ADG_ERASE_FRAME (FRAME, MASK)
Do a hardware erase on specified frame for specified bit planes, if frame not specified (i.e., FRAME = 0) use current FBC settings. Due to hardware bug with GM256 memory the viewport extents are temporarily reduced so that only the specified frame is erased.

FRAME (I*4, by value)
Frame number to erase (1-4).

MASK (I*4, by value)
Write mask set to bit planes that are to be erased.

ADG_ERASE_VIEW (MASK)
Do a hardware erase over the current viewport for the specified bit planes. Due to a hardware bug with GM256 memory there is some runover into the memory region beyond the right edge of the viewport.

MASK (I*4, by value)
Write mask set to bit planes that are to be erased.

ADG_FILL (X, Y, MASK, FILL)
Do a fast software fill over the specified region for the specified bit planes using the specified fill bit pattern.

X (I*4 array of 2 elements, by reference)
UL and LR x window coordinates of image area to fill, will be truncated to even multiples of 16 for lores or 32 for hires.

Y (I*4 array of 2 elements, by reference)
UL and LR y window coordinates of image area to fill.

MASK (I*4, by value)
Write mask set to bit planes that are to be filled.

FILL (I*4, by value)
Bit pattern to fill in over specified range.

ADG_GET_CHANNEL (CHAN)
Get the current channel number from the LUVO crossbar setting.

CHAN (I*4, by reference)
Returned as the channel number.

ADG_GET_ORIGIN (X, Y)
Returns the window coordinates that correspond to viewport coordinates of (0,0).
X (I*4, by reference)
Returned as X window coordinate.
Y (I*4, by reference)
Returned as Y window coordinate.

ADG_GEN_CURSOR(ICURSOR, XOFF, YOFF)
Generate a user-defined cursor shape on the ADAGE.
ICURSOR(32,32) (I*4, by reference)
Cursor shape array. Non-zero elements are turned on.
XOFF (I*4, by reference)
Cursor column offset from upper left corner to hotspot.
YOFF (I*4, by reference)
Cursor row offset from upper left corner to hotspot.

ADG_GET_CURSOR(ICURSOR)
Retrieve cursor shape from ADAGE Global Common and place it in the I*4 array ICURSOR(32,32).
ICURSOR(32,32) (I*4, by reference)
Array containing ADAGE cursor shape.

ADG_HISTOGRAM (X, Y, CHAN, MASK)
Generate a grayshade histogram over the specified image area and channel. Place in global variable "hist[256]".
X (I*4 array of 2 elements, by reference)
UL and LR x window coordinates of image area.
Y (I*4 array of 2 elements, by reference)
UL and LR y window coordinates of image area.
CHAN (I*4, by value)
Channel number on which to generate histogram.
MASK (I*4, by value)
8-bit read mask to apply to image data.

ADG_INIT (RESET, RESOLUTION)
Open a channel to the Adage processor associated with the workstation that the call was initiated from and optionally initialize the registers to a known state.
RESET (I*4, by value) Reset flag.
1 - reset the Adage registers to default values
0 - use previous settings for Adage registers
RESOLUTION (I*4, by value) Set hi/lo res, used only when RESET =1.
1 - high resolution
0 - low resolution

ADG_INIT_BPS ()
Clear the BPS "done" flag in scratch pad memory.

```
ADG_LOAD_BPS (PROGRAM_NAME)
     Download the specified ICROSS binary file in load file
     format to microcode memory.
     PROGRAM_NAME (C*(*), by reference) Character variable
          containing name of the load file.  If no type
          extension is specified, the program assumes
          ".lod".

ADG_LUT_HISTOGRAM (X, Y, CHAN, PLOT)
     Generate a LUT [grayshade] histogram over the specified
     image area and channel.  Optionally generate a plot in
     the magenta overlay plane.
     X (I*4 array of 2 elements, by reference)
          UL and LR x window coordinates of image area.
     Y (I*4 array of 2 elements, by reference)
          UL and LR y window coordinates of image area.
     CHAN (I*4, by value)
          Channel number on which to generate histogram.
     PLOT (I*4, by value)
          Flag to indicate whether a histogram plot is to
          to be generated, if >=0 then generate plot
          starting at column = PLOT.

ADG_MAP_SEC ()
     Map the ADAGE common block to the global section
     associated with the workstation that the call was
     initiated from.

ADG_MONO_ENHANCE (LO, HI)
     Do a linear contrast enhancement between the specified
     pixel levels.  Display pixels outside limits as either
     0 or 255 for above or below the limits respectively.
     LO (I*4, by value)
          Lower pixel level limit.
     HI (I*4, by value)
          Upper pixel level limit.

ADG_MOVE_CURSORV (X, Y)
     Positions the adage cursor at the specified adage
     viewport location.
     X (I*4, by value)
          X viewport coordinate.
     Y (I*4, by value)
          Y viewport coordinate.

ADG_MOVE_CURSORW (X, Y)
     Positions the adage cursor at the specified adage
     window location.
     X (I*4, by value)
          X window coordinate.
     Y (I*4, by value)
          Y window coordinate.
```

ADG_MULTI_LUT (PLANE, QUAD, LUT)
Initialize page bit in the specified refresh memory bit plane such that data in the specified quadrant will be routed through the specified LUT.
PLANE (I*4, by value)
Bit plane to use for page bit (normally 24).
QUAD (I*4, by value)
Quadrant number (1-4) or entire frame (0).
LUT (I*4, by value)
LUT number.

ADG_OVLY_DEFAULT()
Set default colors in overlay LUT such that there is a different color for each plane and data in higher bit planes will overwrite data in lower planes. 1-black, 2-red, 3-green, 4-yellow, 5-blue, 6-magenta, 7-cyan, 8-white.

ADG_PAN_SCROLL(OPTION,X,Y)
Performs pan/scroll such that the memory location X,Y is at a viewport location specified by OPTION.
OPTION (I*4, by reference)
1 = (X,Y) at upper left corner of viewport
2 = (X,Y) at center of viewport
3 = (X,Y) at lower right corner of viewport
X (I*4, by reference) X memory location
Y (I*4, by reference) Y memory location

ADG_PIC_BOX (X, Y)
Use the bitpad puck to drive the ADAGE cursor and define a rectangular pic region on the screen. The region is outlined in the yellow overlay plane.
X (I*4 array of 2 elements, by reference)
Returned as the UL and LR corner x window coords of the pic box.
Y (I*4 array of 2 elements, by reference)
Returned as the UL and LR corner y window coords of the pic box.

ADG_PIC_QUAD (X, Y)
Use the bitpad puck to select a quadrant from a lo-res image on the screen. The quadrant is outlined in the yellow overlay plane.
X (I*4, by value)
Returned as the UL corner x window coords of the quadrant.
Y (I*4, by value)
Returned as the UL corner y window coords of the quadrant.

ADG_PLOT_HIST (X, Y, HEIGHT)
Plot a histogram previously generated by one of the adg_... histogram routines. Can be plotted either

horizontally or vertically.
X (I*4, by value)
    X coordinate of histogram plot origin, specify as negative for a vertical plot.
Y (I*4, by value)
    Y coordinate of histogram plot origin.
HEIGHT (I*4, by value)
    Height (in pixels) of histogram plot.

AGL_READ_BYTE(X, Y, COLS, ROWS, CHAN, BYTE_ARRAY)
Read in a "COLS-by-ROWS" byte array from ADAGE display memory. Upper left corner of the array starts at absolute memory location (X, Y), and "CHAN" specifies the channel number from which display memory is read.
X (I*4, by reference)
    Window x (column) coordinate to start read from.
Y (I*4, by reference)
    Window y (row) coordinate to start read from.
COLS (I*4, by reference)
    Number of columns to be read into output array.
ROWS (I*4, by reference)
    Number of rows to be read into output array.
CHAN (I*4, by reference)
    Channel number to read in (0-3).
BYTE_ARRAY(COLS,ROWS) (Byte, by reference)
    Byte array containing ADAGE memory contents.

ADG_READ_FULLWORD(X, Y, COLS, ROWS, FULLWORD_ARRAY)
Read in a "COLS-by-ROWS" fullword array from ADAGE display memory. Upper left corner of the array starts at absolute memory location (X,Y).
X (I*4, by reference)
    Window x (column) coordinate to start read from.
Y (I*4, by reference)
    Window y (row) coordinate to start read from.
COLS (I*4, by reference)
    Number of columns to be read into output array.
ROWS (I*4, by reference)
    Number of rows to be read into output array.
FULLWORD_ARRAY(COLS,ROWS) (Byte, by reference)
    Byte array containing ADAGE memory contents.

ADG_READ_SPM (NARG, ARG)
Read from the return area of scratch pad memory, used to communicate from the BPS to host.
NARG (I*4, by value)
    Number of arguments to read.
ARG (I*4, by reference)
    Array to receive the arguments read from scratch pad.

ADG_REPLACE_CURSOR(NAME, ICURSOR)
Replace the user-defined cursor shape of name "NAME" with the contents of the array ICURSOR.

NAME (C*(*), by descriptor)
Name of the cursor to be replaced.
ICURSOR(32,32) (I*4, by reference)
Cursor shape array. Non-zero values turn the cursor pixel on; zero values turn it off.

ADG_RUN_BPS ()
Start the BPS and let it run until the DONE flag is set, then stop the processor.

ADG_SAVE_CURSOR(NAME, ICURSOR)
Write the user-defined cursor shape to the ADAGE cursor shape data file using the shape in the I*4 ICURSOR(32,32) array.
NAME (C*(*), by descriptor)
Cursor name.
ICURSOR(32,32) (I*4, by reference)
Cursor shape array. Non-zero values turn the cursor pixel on; zero values turn it off.

ADG_SCALE_LUT(TABLE, PERCENT)
Scale down the lookup table intensities.
TABLE (I*4, by reference)
Lookup table to scale down. 1=Image LUT, 2= Overlay LUT, 3=Cursor LUT, 4=Cursor + Overlay LUT, 5=All 4 at once.
PERCENT (I*4, by reference)
Scale factor; scale the intensities to "percent" of their current value (0-100).

ADG_SET_CSRCOLOR(COLOR)
Change the cursor color.
COLOR (C*(*), by descriptor)
Color to change the cursor to. User must supply as many characters as needed to make the color choice unambiguous. Choices:

| | | |
|---|---|---|
| Black | Blue | Cyan |
| Gray | Green | Purple |
| Red | White | Yellow |

ADG_SET CHANNEL (IN_CHANNEL, OUT_CHANNEL)
Set the crossbar switch such that the specified input channel is mapped equally to all three color guns out through either the background or alpha path.
IN_CHANNEL (I*4, by value)
Input channel number (0-3) or 4 for identity (i.e., full color).
OUT_CHANNEL (I*4, by value)
Output type, 0 for background (i.e., channels 0 - 2), 1 for alpha (i.e., channel 3).

```
ADG_SET_CURSOR(NAME, ICURSOR)
     Read in a user-defined cursor shape from the ADAGE
     Cursor Shape data file and place it into an I*4(32,32)
     array.  Also loads shape into ADAGE.
     NAME (C*(*), by descriptor)
          Cursor name to be loaded.
     ICURSOR(32,32) (I*4, by reference)
          Array cursor shape that is read in.  Non-zero
          pixels are turned on; zero pixels turned off.

ADG_SET_FRAME (FRAME)
     Set the viewport and window to the specified frame.
     FRAME (I*4, by value)
          Frame number (1-4).

ADG_SET_LUT (TABLE, INDEX, COUNT, BUFFER)
     Set the specified LUT values in the specified table to
     the values provided in the calling argument.
     TABLE (I*4, by value)
          LUT number (1-4).
     INDEX (I*4, by value)
          Index position into LUT to begin sorting new
          values (1-256).
     COUNT (I*4, by value)
          Number of LUT entries to be set.
     BUFFER (I*4, by reference)
          Buffer containing the new LUT values.

ADG_SET_LUVO (CHANNEL)
     Set the LUVO channel crossbar to map specified input
     channel equally to all three color guns or reset to
     straight through.
     CHANNEL (I*4, by value)
          Input channel number (0-2 or 4 for straight
          through).

ADG_SET_ORIGIN (X, Y)
     Set the window so that the value at the specified frame
     buffer coordinates is displayed as the first pixel at
     the upper left of the viewport.
     X (I*4, by value)
          Column frame buffer coordinate.
     Y (I*4, by value)
          Row frame buffer coordinate.

ADG_SET_PIXCLK(PERCENT)
     Set the pixel clock to desired value.
     PERCENT (I*4, by reference)
          Value to set the pixel clock to.  Percent=0 ->
          Thinnest Picture; Percent=100 -> Widest Picture.
          Percent=-1 gives a square aspect ratio.
```

```
ADG_SET_SHADE (SHADE)
     Set the shade register to the specified value.
     SHADE (I*4, by value)
          Shade register value.

ADG_SET_VIEWPORT (X, Y, NCOL, NROW)
     Set the viewport location and size.  Enter a -1 to
     leave a parameter unchanged.
     X (I*4, by value)
          Column coordinate of upper left corner.
     Y (I*4, by value)
          Row coordinate of upper left corner.
     NCOL (I*4, by value)
          Number of columns.
     NROW (I*4, by value)
          Number of rows.

ADG_SET_WINDOW (X, Y)
     Set the window offset that corresponds to screen coord
     (0,0).
     X (I*4, by value)
          X window offset.
     Y (I*4, by value)
          Y window offset.

ADG_SET_WINDOW_VIEWPORT(XW,YW,XV,YV)
     Sets the window offset such that memory location XW,YW
     is visible at viewport location XV,YV.
     XW (I*4, by value) X memory location
     YW (I*4, by value) Y memory location
     XV (I*4, by value) X viewport location
     YV (I*4, by value) Y viewport location

ADG_SET_WMASK (MASK)
     Set the write mask to the specified value.
     MASK (I*4, by value)
          Write mask value.

ADG_SET_WMODE (MODE)
     Set the write mode to the specified transfer word size
     for byte (1), half word (2), or long word (4) writes to
     frame buffer memory.
     MODE (I*4, by value)
          Write mode value.

ADG_SET_XBAR(CROSSBAR)
     Set the ADAGE Crossbar Switch, pass (-1) to leave
     unchanged and 63 to turn off.
     CROSSBAR (I*4 array of 32 elements, by reference)
          Array containing the value to be written to the
          crossbar switch.
```

```
ADG_SET_XBAR(OUT, IN)
     Set a single value of the Adage crossbar switch.
     OUT (I*4, by value)
          Output bit-plane of the crossbar that is to be
          set (i.e., the index of the adage common xbar
          array starting at 0).
     IN (I*4, by value)
          Input bit-plane that is to be mapped to the
          specified output bit-plane (i.e., the value to be
          stuffed into the adage common xbar array at index
          OUT).

ADG_SET_XBAR_CHAN (COLOR)
     Set the crossbar so that the specified channels are
     output to the red, green, blue color gun or overlay
     respectively, use -1 to leave setting unchanged.
     COLOR (I*4 array of 4 elements, by reference)
          Each element of the array contains the channel
          number to map to the respective color gun (i.e.,
          COLOR(1) = channel for red gun, COLOR(2) = channel
          for green gun, COLOR(3) = channel for blue gun and
          COLOR(4) = channel for overlay).

ADG_START_BPS ()
     Start the BPS running.

ADG_STOP_BPS
     Stop the running BPS.

ADG_TOGGLE_CHANNEL (CHAN, FLAG)
     Toggle the specified output channel, or all channels,
     on and off using the crossbar switch.
     CHAN (I*4, by value)
          Output channel (0-4) to toggle.
     FLAG (I*4, by value)
          On/off flag (1-on, 0-off).

ADG_TOGGLE_CURSOR (FLAG)
     Toggle the cursor on and off using the FBC control
     register.
     FLAG (I*4, by value)
          On/off flag (1-on, 0-off).

ADG_TOGGLE_PLANE (PLANE, FLAG)
     Toggle the specified bit plane on and off using the
     crossbar switch.
     PLANE (I*4, by value)
          Bit plane (0-33) to toggle.
     FLAG (I*4, by value)
          On/off flag (1-on, 0-off).

ADG_VCR (FLAG)
     Switch between external (VCR mode) and internal synch,
     use RS-170.
```

FLAG (I*4, by value)
On/off flag (1-on, 0-off).

ADG_WAIT_BPS
Wait for the BPS done flag to be set.

ADG_WRITE_BYTE(X, Y, COLS, ROWS, CHAN, BYTE_ARRAY)
Write out a "COLS-by-ROWS" byte array from ADAGE display memory. Upper left corner of the array is written to absolute memory location (X,Y), and "CHAN" specifies the channel number to which display memory is written.
X (I*4, by reference)
Window x (column) coordinate to start write at.
Y (I*4, by reference)
Window y (row) coordinate to start write at.
COLS (I*4, by reference)
Number of columns to be written into display memory.
ROWS (I*4, by reference)
Number of rows to be written into display memory.
CHAN (I*4, by reference)
Channel number to read in (0-3).
BYTE_ARRAY(COLS,ROWS) (Byte, by reference)
Byte array containing contents to be written to ADAGE memory.

ADG_WRITE_FULLWORD(X, Y, COLS, ROWS, FULLWORD_ARRAY)
Write out a "COLS-by-ROWS" fullword array from ADAGE display memory. Upper left corner of the array is written to absolute memory location (X,Y).
X (I*4, by reference)
Window x (column) coordinate to start writing to.
Y (I*4, by reference)
Window y (row) coordinate to start writing to.
COLS (I*4, by reference)
Number of columns to be written to display memory.
ROWS (I*4, by reference)
Number of rows to be written to display memory.
FULLWORD_ARRAY(COLS,ROWS) (Byte, by reference)
Byte array containing contents to be written to ADAGE memory.

ADG_WRITE_IMAGE (FILE, TYPE, LINES, ELES, ROW, COL, X, Y, PAGE)
Write image data from specified file into refresh memory, specify data type, number of lines and elements to load, starting row and column number in the file, starting refresh memory address, and page bit value.
FILE (C*n, by reference - null terminated string)
Fully qualified file name.
TYPE (I*4, by value)
Data format type; 1 - 1 byte/pixel, 2 - 1 word/pixel, 3 - 1 word/pixel (MPS) format.

```
      LINES (I*4, by value)
           Number of lines to load.
      ELES (I*4, by value)
           Number of elements to load.
      ROW (I*4, by value)
           Starting row (byte or word) number in file.
      COL (I*4, by value)
           Starting column (record) number in file.
      X (I*4, by value)
           Starting x address in refresh memory.
      Y (I*4, by value)
           Starting y address in refresh memory.
      PAGE (I*4, by value)
           Page bit value; 0 for LUT 1&2, 1 for LUT 3&4.

ADG_WRITE_SPM (NARG, ARG)
      Write to the input area of scratch pad memory, used to
      communicate from the host to the BPS.
      NARG (I*4, by value)
           Number of arguments to write.
      ARG (I*4, by reference)
           Array containing the arguments to write to scratch
           pad.

ADG_XFER (IN_X, IN_Y, NCOL, NROW, IN_CHAN, OUT_X, OUT_Y,
      OUT_CHAN)
      Do a fast transfer of one channel of image data to
      another location in refresh memory.
      IN_X (I*4, by value)
           UL x coord of input image.
      IN_Y (I*4, by value)
           UL y coord of input image.
      NCOL (I*4, by value)
           Number of columns to transfer.
      NROW (I*4, by value)
           Number of rows to transfer.
      IN_CHAN (I*4, by value)
           Input image channel number (0-3).
      OUT_X (I*4, by value)
           UL x coord of output image.
      OUT_Y (I*4, by value)
           UL y coord of output image.

ADG_ZOOM(ZOOM)
      Set the x and y zoom registers to ZOOM.
      ZOOM (I*4, by reference) (0-15)

ADG_ZOOM_XY(ZX,ZY)
      Sets the x and y zoom registers to ZX and ZY
      respectively.
      ZX (I*4, by reference) X zoom factor
      ZY (I*4, by reference) Y zoom factor
```

ADG_ZOOM_PAN_SCROLL(ZX,ZY,X,Y)

Performs zoom/pan/scroll such that memory location X,Y is in the center of the viewport.

ZX (I*4, by reference) X zoom factor

ZY (I*4, by reference) Y zoom factor

X (I*4, by reference) memory X coordinate for center of viewport.

Y (I*4, by reference) memory Y coordinate for center of viewport.

## B. GOES Satellite Data for AIMS

### 1. Introduction

AFGL has a continuing need for satellite sounding data for test and evaluation of temperature retrieval algorithms, and for temperature profiles for input into numerical weather prediction models. The Geostationary Operational Environmental Satellite (GOES) VISSR Atmospheric Sounder (VAS) provides sounding data on an operational basis. The AFGL groundstation has a capability to ingest mode AAA data from GOES-VAS and thus is a significant potential source of the needed data. STX was tasked to develop software to accomplish data ingest, data storage, and data transmission.

### 2. User Requirements for Sounding Data

The original requirements for dwell sounding data from GOES-VAS specified a capability to interactively request user-defined subsets of the dwell sounding frame extracted from the broadcast mode AAA data stream. All 12 channels would be ingested, re-interleaved by pixel, and stored in a non-competitive environment. Along with the actual sounding data two additional data sets would require consistent handling and storage. The first, known as directory data, would contain information identifying the data and its areal extent, resolution, and location. The second, known as navigation data, would contain orbital parameters describing the satellite's location and motion.

During the contract year informal discussions with users of GOES-VAS dwell sounding data revealed that higher than expected noise had been consistently identified in the data, despite an imposed spin budget for each channel. Quantitative use of the data was found to be marginal at best, requiring extensive spatial averaging techniques in an

attempt to minimize the effects of instrument noise. Based on this finding AFGL decided to discontinue support for dwell sounding data from GOES-VAS and redirect efforts toward acquiring sounding data from the next generation of GOES satellites (GOES-NEXT). The GOES-NEXT program will introduce a total of five spacecraft with increased capabilities over the current series; see for example Komajda (1987). The first launch of this series is tentatively scheduled for Fall 1989.

Reorientation of effort led to deferral of development of the sounding data ingest software, in part because of lack of finalized information on the GOES-NEXT retransmission format. Further, sounding data are only a portion of the data specified for the AIMS GOES groundstation. The software design for the AIMS GOES groundstation is based on specifications, developed earlier by STX (see Gerlach, 1986), for a fully-functional system that supports multispectral imagery, ephemeris, and grid information as well as sounding data. This report describes what STX accomplished during the contract year to establish for AIMS a capability for data ingest support, data storage, and data transmission for GOES sounding data, multispectral imagery, navigation data, and grid information.

## 3. Data Storage

### a. Database Design

#### 1). Satellite Digital Disk Areas

The database design includes the concept of satellite digital disk areas. An area can be visualized as a two-dimensional array composed of disk-based satellite lines arranged one below the other and numbered from top to bottom. Each line contains a sequence of satellite elements

arranged across the line and numbered from left to right. Elements represent anywhere from one to 32 bits of information and reside in either an 8-, 16-, or 32-bit computer unit of storage. The area numbering scheme assigns coordinates (known as the area coordinates) to each line/element pair.

Each area element may consist of multiple measurements from different spectral bands for the same spatial location. Under this configuration each element on a given line will contain measurements for the bands arranged in sequence from left to right. This interlacing technique increases the length of the area line by a factor of N, where N is the number of bands available for the given satellite line. The primary reason for interlacing multispectral data is to minimize the amount of I/O required to access the data. This is important in the interactive use of sounding data where a user will typically want to access all spectral bands for a particular geographic location.

2). Area Directory

When an area is added to the AIMS GOES groundstation data set, it is assigned a unique number that serves as a pointer in a supporting data file called the area directory. The area directory is a fixed-length file composed of logical records with a predefined structure. These logical records are known as area directory entries and contain information describing the salient characteristics of the data as well as other information, such as data format. The position of the directory entry for a particular area is simply indexed using the area number. If a directory entry is empty, the corresponding area does not exist. The structure of the area directory entry is adapted from Dengel and Santek (1986) and is shown in Table 1.

TABLE 1. AREA DIRECTORY ENTRY STRUCTURE

| Word | Description |
|---|---|
| 1. | Area status (<0 area does not exist<br>=0 area exists, no data present<br>=1 area exists, data present<br>>1 area in use) |
| 2. | Satellite identification, year, and Julian day of satellite data in SSYYDDD format |
| 3. | Time of satellite data in HHMMSS format |
| 4. | Upper left-hand line in satellite coordinates |
| 5. | Upper left-hand element in satellite coordinates |
| 6. | Number of satellite lines in area |
| 7. | Number of satellite elements per line in area |
| 8. | Number of bytes per data element (1, 2, or 4) |
| 9. | Line resolution (sampling/averaging rate) |
| 10. | Element resolution (sampling/averaging rate) |
| 11. | Z resolution (number of bands in interlaced data) |
| 12,13. | User name that created the area |
| 14. | Day area was created in YYDDD format |
| 15. | Time area was created in HHMMSS format |
| 16. | Intended data type (e.g., for VAS: visible<br>small detector IR<br>large detector IR) |
| 17. | Multispectral band numbers (bit mapped, where<br>LSB = band 1<br>MSB = band 32) |
| 18. | Validity code (0 = no validity code) |
| 19. | Length of line prefix (in bytes) |
| 20. | Length of prefix documentation section (in bytes) |
| 21. | Length of prefix calibration section (in bytes) |
| 22. | Length of prefix level map (in bytes) |

| Word | Description |
|---|---|
| 23. | Satellite source type (e.g., VAS, VISSR) |
| 24. | Satellite source sub-type (e.g., for VAS: MCI, DS) |
| 25. | Area calibration type (e.g., raw measurement<br>brightness count<br>brightness temperature<br>radiance) |
| 26. | Group number |
| 27-30. | Identification (user assigned) |
| 31-32. | Reserved for system use |

An area line is composed of two sections, a line prefix and the actual line data. The line prefix contains supplementary information for a given line and can be essentially ignored when the information is not of interest. The size of the prefix and the information contained in it depend on the area type, which in turn is determined by the data source. The area type is indicated by word 23 of the area directory entry. Every area line for a particular area has the same length prefix. This length is measured in bytes and is given in word 19 of the area directory entry. The line prefix may optionally begin with a validity code. If utilized, the value of the validity code is set in word 18 of the area directory entry as a non-zero integer. When present, the validity code occupies the first four bytes of the line prefix. To use the validity code, a comparison is made between the code imbedded in the given line and the value set in the area directory entry. If the two do not match, the line is considered invalid and should be ignored. This provides a quick method of detecting missing (e.g., due to ingest problems) or anomalous lines prior to any processing involving the line.

In addition to the validity code, the line prefix may contain as many as three other regions. The length of each of these regions in bytes is given by words 20, 21, and 22 of the area directory entry. If present, the first region contains information specific to a satellite line, such as line documentation. The second region contains satellite line-dependent calibration coefficients repeated once for every band present in the area line. The final region that may exist in the line prefix is the level map. For each band present in an interlaced line a one byte entry containing the band number is allocated. For areas containing a single band (e.g., VISSR IR data), the level map is optional; otherwise it must be present.

Because a level map must be associated with every area line if present, it can easily accommodate the fact that area lines containing multispectral data may not necessarily comprise the same number of bands (though space is reserved) and/or share the same order of sequence of bands. This is accomplished by ordering the level map to reflect the configuration of the given line and utilizing the map as an index. Only the left-most N bytes of the level map will contain non-zero data, where N is the actual number of bands comprising each element of a particular line. Ordering the level map in this way, the i'th byte will correspond to the i'th band slot within each element of the line. Unused level entries as well as unused band slots are set to binary zero.

3). Navigation Data

In addition to directory and satellite data, navigation data are also an integral part of the database. Earth location parameters are extracted from the broadcast data stream and stored in a separate data file. When used in conjunction with software that provides a generalized solution to the earth location problem, precise earth location of satellite data is possible. Since navigation data are transmitted along with satellite data, their availability is governed only by the broadcast schedule. In practice, these parameters are updated weekly unless a satellite maneuver is executed. Consequently, there does exist a number of redundant sets at any point in time; however, considering the relatively small size of a set (25 32-bit words), it is a small price to pay for a database that automatically retains the most up-to-date version. The structure of the navigation data set is shown in Table 2.

TABLE 2. NAVIGATION ENTRY STRUCTURE

| Word | Description |
|---|---|
| 1. | Satellite identifier, year, and Julian day of navigation in SSYYDDD format |
| 2. | Time of navigation in HHMMSS format |
| 3. | Orbit type (1 = Classical, 2 = Phillips) |
| 4. | Epoch day in YYMMDD format |
| 5. | Epoch time in HHMMSS format |
| 6. | Semi-major axis |
| 7. | Eccentricity |
| 8. | Inclination |
| 9. | Mean anomaly |
| 10. | Argument of perifocus |
| 11. | Right ascension of ascending node |
| 12. | Attitude - Spin axis declination |
| 13. | Attitude -Spin axis right ascension |
| 14. | Attitude - Picture center line |
| 15. | Spin rate |
| 16. | Frame geometry - Angle of N-S frame scan |
| 17. | Frame geometry - Total number of lines (encoded) |
| 18. | Frame geometry - Angle of E-W frame scan |
| 19. | Frame geometry - Total number of elements |
| 20. | Camera geometry - Pitch |
| 21. | Camera geometry - Yaw |
| 22. | Camera geometry - Roll |
| 23. | Sun sensor mounting angle |
| 24. | Precession rate |
| 25. | Precession direction |

4). Grouped Areas

If satellite data ingest consisted of single channel data and navigation information, interrelating these data sets with a corresponding area directory entry would be straightforward. However, with the introduction of multispectral data under GOES-VAS, navigation data become collectively associated with multiple data sources. To maintain an interrelationship among area directory, data, and navigation, the concept of grouping data files according to user-specified criteria was introduced.

As applied to the acquisition and dissemination of GOES-VAS satellite data, areas are grouped by satellite/date/time. This grouping scheme is implicit in that a group number that uniquely identifies a group of satellite areas is assigned at ingest, an event that is uniquely identified by the satellite of interest, date, and time. The group number is recorded in word 26 of the area directory entry for each area that is a member of the group. A group is not required to retain all of its original members to remain a group. For instance, if an area that is a member of a group is deleted, the remaining members retain the identity of the group.

With the additional concept of groups, an interrelationship among the three data types can easily be established. Each area directory entry has exactly one satellite data file associated with it and a group of satellite data files is associated with a single navigation data set. Taking advantage of this interrelationship, the group number, in much the same manner as the area number, serves as a pointer to the navigation data set associated with the group of satellite data files. In addition, the name of a satellite data file is chosen to include the area number as an integral part. By applying the concept of

areas and groups of areas, only the area directory need be searched to locate any information regarding satellite data since the area directory entry location defines the name of the satellite data file, and the entry contains the group number of which the area is a member and which, in turn, is the key to the corresponding navigation data set. This design also reduces disk I/O requirements due to the relatively small size of the area directory file. Fig. 1 illustrates the interrelationships among the area directory, satellite data, and navigation data.

b. Database Management

1). Management Software

With a database structure defined for each of the data types, the software modules comprising the database manager could be designed and written. The primary set of routines that collectively define a satellite digital disk area library consists of modules for area management, rotating area archive management, and I/O routines to access the area directory. Where appropriate the I/O subsystem described in Gerlach (1986) was used to provide low-overhead, unbuffered I/O services in accessing the area directory file. In all, 22 routines comprise the area library. The primary set of routines that collectively define a navigation library contains modules for single entry retrieval and transformations between earth-based coordinates and satellite coordinates. The navigation library is composed of eight routines. An additional library was generated to group modules that perform general utility functions. This library consists of 15 modules.

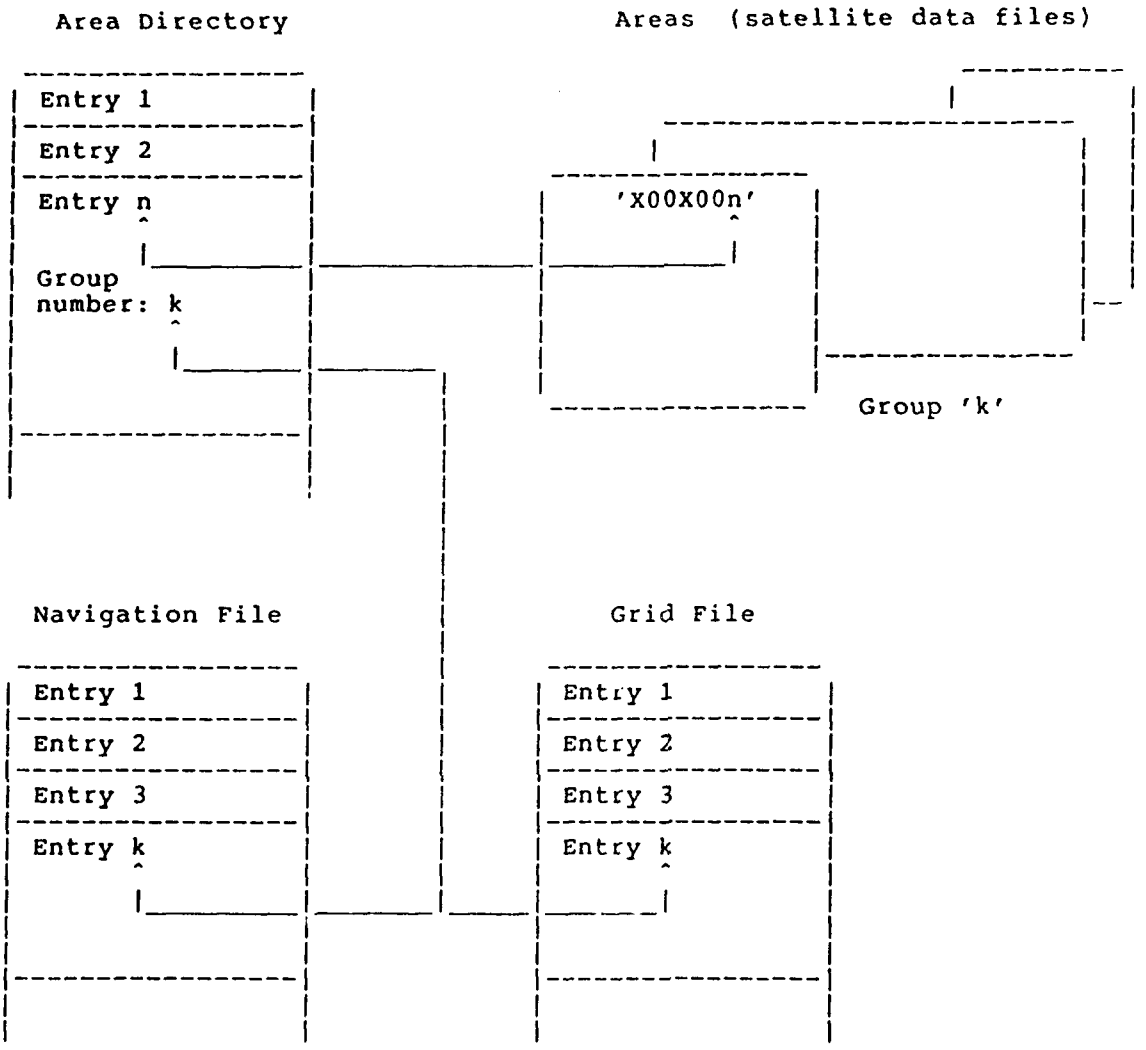



Fig. 1. Illustration of Interrelationship among Area Directory, Satellite Digital Disk Areas, Navigation File, and Grid File.
(Grid File is Possible Future Database Component, Included for Illustration Only.)

2). Satellite Disk Configuration and Management

Combination of diverse data sets from GOES-VAS (i.e., multispectral imagery, dwell sounding, and sounding products) and the availability of three disks to store the data raised the question of where to store what data. One possibility was to hardwire a particular data set to a particular disk; however, this solution would impose software modifications if the configuration were altered or a new disk were added to the system. As an alternative, STX proposed and developed the concept of an interactive satellite disk configuration utility. The utility eliminates the need for software changes and accommodates a total of 11 disks. It allows the user to associate satellite data types and valid times with a specific disk, specify the maximum number of areas to be mapped to the disk, and logically connect two or more disks to provide extended archiving capabilities.

The satellite disk configuration utility utilizes a disk-based data structure to manage configurations. This structure resides in the first sector of the area directory file and is accessible whenever the area directory is read from mass storage. Applications programs are automatically prevented from accessing this structure to avoid indiscriminate access and/or contamination. The structure consists of 11 entries, each entry composed of 17 32-bit words. Table 3 shows the structure of a configuration entry. Areas are mapped to a disk that is identified by volume and directory name. These names reside in words 1-4 of the configuration entry. There is no limit to the number of areas that can be mapped to a single disk; however, the same areas cannot be mapped to multiple disks. The areas to be mapped are delineated by words 9 and 10 of the entry, the begin and end area numbers. Area content is described by the type of data and the valid time of the data. For real

TABLE 3. SATELLITE DISK CONFIGURATION ENTRY STRUCTURE

| Word | Description |
|---|---|
| 1,2. | Name of volume (disk) |
| 3,4. | Name of directory on volume |
| 5. | Volume status |
| 6. | Data types (s) (e.g., for VAS: multispectral imagery, raw sounding data, sounding products, rapid scan data) |
| 7,8. | Data time(s) in minutes past the hour (bitmapped, where LSB = 0 minutes MSB = 63 minutes) |
| 9. | Beginning area number |
| 10. | Ending area number |
| 11. | Oldest area number |
| 12. | Next area number to be generated in sequence |
| 13. | Total amount of volume space available for areas (in bytes) |
| 14. | Total amount of volume space currently occupied by areas (in bytes) |
| 15. | Link activation flag (=false link not activated =true link activated) |
| 16. | Link forward address (=0 no link >0 valid link address 'NONE' link terminator) |
| 17. | Link backward address (=0 no link >0 valid link address 'NONE' link terminator) |

time acquisition of GOES-VAS data, data types include multispectral imagery, raw dwell sounding data, and sounding products. The valid time of GOES-VAS data corresponds to the ingest time. Data type and time are given by words 6-8 of the configuration entry.

Configuration entries are identified by an entry number that is assigned according to the position of the entry. Using the entry number assignment, entries can be linked together to form a chain of entries. The purpose of linking entries is to extend the archive capabilities of a particular data type/time across multiple disks in a manner that is transparent to the user. The first entry in a chain is a terminating entry, that is, an entry that can only be linked in one direction (forward). Similarly, the last entry in a chain is a terminating entry because it can only link in the backward direction. Entries that link to terminating entries and to each other are non-terminating because they have links in both directions. Forward and backward link addresses are stored in words 16 and 17 of the configuration entry. Non-zero values indicate that the disk associated with the given entry is to be link activated according to the link addresses after the last area in the series has been generated. Word 15 indicates whether the link is active or not. In a typical scenario the first entry in a chain is accessed and the corresponding disk is utilized until the final area is generated. The entry is then link activated so that any further archive of a particular data type/time is automatically routed to the entry pointed to by the link forward address in the link activated entry. This procedure continues until the end of the chain is reached. The chain is then traversed backward until the first entry is found. All entries are then link deactivated and the procedure repeats itself.

The configuration entry also contains information important to maintaining a disk-based rotating area archive primarily for GOES-VAS multispectral imagery. Two pointers are required for this capability, one that points to the oldest area on the disk and one that indicates the next area to generate in sequence. The oldest area is the first to be deleted if the amount of disk space is insufficient to generate a new area. The deletion process continues, if necessary, with the next oldest area until sufficient space exists. These pointers are given by words 11 and 12 of the configuration entry. The amount of disk space, in bytes, occupied by all areas is recorded in word 14 of the configuration entry and the total number of bytes available for disk space allocation is given by word 13. These two parameters are used to determine if existing areas must be deleted to make room for new areas.

4. Data Ingest

a. Automated Scheduling of the Real Time Ingest Sequence

Automated scheduling of the real time ingest sequence represents the combined effort of a number of software components continuously to ingest GOES-VAS data on a fixed schedule without operator assistance. The design philosophy has been to develop independent programs that perform specific tasks and then utilize these programs in a macro that when executed will achieve the desired result. The alternative to this approach would be to develop a single program responsible for managing every aspect of this process. The choice of components is important for several reasons. First, as programs are developed they can be tested and debugged independently of other programs. This shortens the software development cycle since problems are easily isolated. Secondly, the user has more control over how the process is executed. Every program is designed to

accept information interactively and thus provides an environment for the user to tailor the process to meet changing needs. Finally, the user has the option of interactively executing programs if automation is not desirable or the user requires more control over the process.

The programs necessary to provide collectively an automated ingest capability include (1) those that comprise the ingest sequence, (2) scheduling software to insure that programs are activated at the proper time and to provide a contingency plan should external events prevent the schedule from being met, and (3) software to manage macro execution, that is, a program to insure synchronous execution of each program within a macro based on program order. With these programs developed, macros can be written to automate the ingest sequence and provide continuous automated ingest sequencing.

1). Ingest Sequence Programs

The ingest sequence is comprised of four programs. These programs are interactive in nature, requiring the user to key-in the program command name and command parameters that may follow. For some parameters, default values supplied by the program provide the user the option of specifying those parameters. All four programs utilize the software libraries described earlier as foundation software to perform specific tasks. The first program in the sequence, GENAREA, allocates sufficient disk space for a satellite digital disk area, updates the corresponding area directory entry with the area dimensions, and zero-fills the area to insure that no extraneous data are present prior to ingest. The second program, DEFAREA, fills the remaining words of the area directory entry with parameters necessary to define the earth disk subset. Satellite, date, time,

data type, resolution, and reference point are specified by the user and entered into the area directory entry by DEFAREA. The third program is GRPAREA, which is responsible for grouping one or more areas by assigning a number that will uniquely identify the areas as a group. GRPAREA updates the area directory entries of the corresponding areas to be grouped with the assigned number. The final program in the sequence is the ingest program, INGAREA. This program utilizes the area numbers as an index to access the appropriate area directory entries. At this point the entries contain all the information necessary for INGAREA to ingest and extract the user-requested satellite data.

2). Scheduling Software

The scheduling software consists of two programs, the scheduler, program SKEDULER, and the schedule activator, program SKEDACTV. SKEDULER is an interactive program although it is called primarily from macros. It is responsible for managing scheduled events, that is, events defined as either the execution of a single program or a sequence of programs that require synchronous activation. When utilized in a macro, all programs bounded by the call to the scheduler and either the end of the macro or another call to the scheduler are scheduled as an event. The event is scheduled according to the date and time specified in the call to the scheduler and is subject to change based on two additional parameters. The first parameter is a tolerance level that represents the time in seconds that the user is willing to "tolerate" and still consider the event as being activated on time. If the tolerance level cannot be met, a contingency plan is executed. The contingency plan is the second parameter and is parameterized as a code number. The user can choose from three plans: ignore the late start and proceed as normal, abort the scheduled event altogether,

or proceed somewhere other than at the beginning of the event.

Information regarding scheduled events is stored and managed by the scheduler in a disk-based schedule file. The advantage of having this information disk-based as opposed to memory-based is that scheduled events are retained if the computer system is down. The schedule file is sectioned into eight-word link listed logical records, one record for each scheduled event. The file accommodates 95 records (therefore 95 scheduled events) and can be expanded to support in excess of this number by recreating the file with a larger size. The first record of the schedule file, known as the head cell, always contains information associated with the next scheduled event or zeroes if nothing has been scheduled. Table 4a shows the structure of the head cell. The remaining 95 records contain parameters describing each event. These parameters are the scheduled year, day, and time, the name of the macro file, the position within the macro to start execution (does not necessarily have to be the beginning of the macro), tolerance, contingency, and a pointer to the link listed record. Table 4b shows the structure of logical records 1-95.

The other program that together with the scheduler provides the scheduling capability is the schedule activator, SKEDACTV. Unlike the scheduler, SKEDACTV is not available to the user as an interactive program. Instead, it is activated automatically by the scheduler as long as there is at least one outstanding scheduled event. After this initial activation, SKEDACTV utilizes the timer services of the MPX-32 operating system to activate itself once every ten seconds to compare the wall clock date and time with the date and time of the next scheduled event. If the two times are within the specified tolerance, SKEDACTV will activate the scheduler to initiate event processing.

TABLE 4a. SCHEDULE FILE HEAD CELL STRUCTURE

| Word | Description |
|---|---|
| 1. | Year and day of next scheduled event in YYDDD format |
| 2. | Time of next scheduled event expressed as 1/60ths of seconds past midnight |
| 3. | Next available record |
| 4-6. | Not used |
| 7. | Next record to process |
| 8. | Not used |

TABLE 4b. SCHEDULE FILE RECORD STRUCTURE

| Word | Description |
|---|---|
| 1. | Year and day of scheduled event in YYDDD format |
| 2. | Time of scheduled event expressed as 1/60ths of seconds past midnight |
| 3. | Event (macro) file name |
| 4. | Event (macro) chain pointer |
| 5. | Tolerance |
| 6. | Contingency |
| 7. | Next record in the link list |
| 8. | Not used |

SKEDACTV is also automatically activated after a system bootstrap.

3). Macro Expander

As designed, the scheduler requires that the execution of a macro be temporarily suspended when the scheduling capability is utilized. Although the MPX-32 operating system normally manages macro execution, it does not have the capability to satisfy the above requirement. Consequently, software was developed and implemented to handle this function. Program MACEXP is responsible for synchronous activation of programs that comprise a macro and under normal conditions performs much the same as its MPX-32 counterpart. However, in a manner that is transparent to the application, a program that has been activated via MACEXP will always signal that it has completed execution using the interprocess communication services of MPX-32. As part of the communications, the exiting program may optionally indicate to the macro expander that macro execution is to cease. In this case, MACEXP will not delete the macro file as it normally does after macro execution has been completed. MACEXP was written as an applications program and as such is available to be executed interactively, although typically it is program activated by macro generation programs or the scheduler.

4). Macro Generation Programs

With the completion of the above programs, macros could be developed to automate the ingest sequence and provide continuous automated ingest sequencing. Macros are created using macro generation programs. These are interactive programs that accept input from the user, build the command strings representing the programs to be executed, and assemble the command strings in a disk-based file known as a

macro file. User input is primarily the specification of parameters that represent the command parameters of the various commands comprising the macro. Thus, where appropriate, user input is inserted into the command strings. The last step of a macro generation program is to activate the macro expander, indicating the name of the macro to initiate and where within the macro to begin execution (known as the macro chain pointer).

The first macro generation program developed, GOESMACR, is designed to schedule an ingest sequence based on the current VAS mode of operations as detected by the real time ingest software. User input consists of specifying the channel(s) of interest, areal extent, reference point, and resolution for a particular satellite, date, and time. As shown in Table 5, GOES-VAS supports a number of operational modes each supporting a different number of data channels. Thus it is important that GOESMACR know the current mode of operations so that the correct type and number of channels can be accommodated and adjustments can be made if user input is incompatible with the current mode. The current mode is always available to the ingest software because the mode is identified in the broadcast data stream. Consequently, this software has the responsibility of recording the mode of operations after every ingest period.

The second macro generation program developed, GOESAUTO, utilizes GOESMACR to provide continuous automated ingest sequencing based on an initial satellite, date, and time and time interval specified by the user. As mentioned previously, scheduling events is only limited by the size of the schedule file. With the current default size a total of 95 events can be scheduled at any one time. To achieve continuous automated scheduling, GOESAUTO re-schedules itself to execute after the 94th scheduling of the ingest sequence.

TABLE 5. OPERATIONAL MODES FOR GOES-VAS

| Mode | Data Channels Available |
|---|---|
| VISSR | Visible<br>1 small detector IR |
| 2-stage MSI* | Visible<br>1 small detector IR<br>1 large detector IR |
| 3-stage MSI | Visible<br>1 small detector IR<br>2 large detector IR |
| 4-stage MSI | Visible<br>4 large detector IR |
| DS** | 12 IR, small and large detector |

* MSI is Multispectral Imaging
** DS is Dwell Sounding

## 5. Data Transmission

The primary requirement for access to GOES satellite data from a communications standpoint is the ability to request and receive user-defined subsets from the online archive residing on the groundstation computer from anywhere on the AIMS ethernet cluster. A secondary requirement is the ability to alter the scheduled ingest sequence to accommodate individual requirements (e.g., real time processing).

The system configuration of AIMS, as illustrated in Fig. 2, does not lend itself easily to accommodating satellite data from the GOES groundstation computer. This is because the computer, a Gould SEL 32/27, cannot be directly connected to the AIMS ethernet cluster, primarily because of hardware incompatibilities. Instead, the 32/27 and AIMS VAX 11/750 are linked with third-party hardware and software that provide high-speed parallel communications between DEC and Gould computers. The package is known as Q-LINK and was purchased with options to provide peripheral sharing, automatic file transfer/translation, remote log-on capability, and a DECnet capability know as Q-NET.

Based on the above requirements, the use of Q-LINK has been defined as a bi-directional link for the communication of control information and data transmitted and received asynchronously. Software design specifications will be based on current and anticipated requirements for data with emphasis on maximizing data throughput without sacrificing data integrity and minimizing the need for temporary mass storage. Of the software purchased with Q-LINK, Q-NET has been selected to take advantage of its network control and error recovery capabilities and also to maintain an

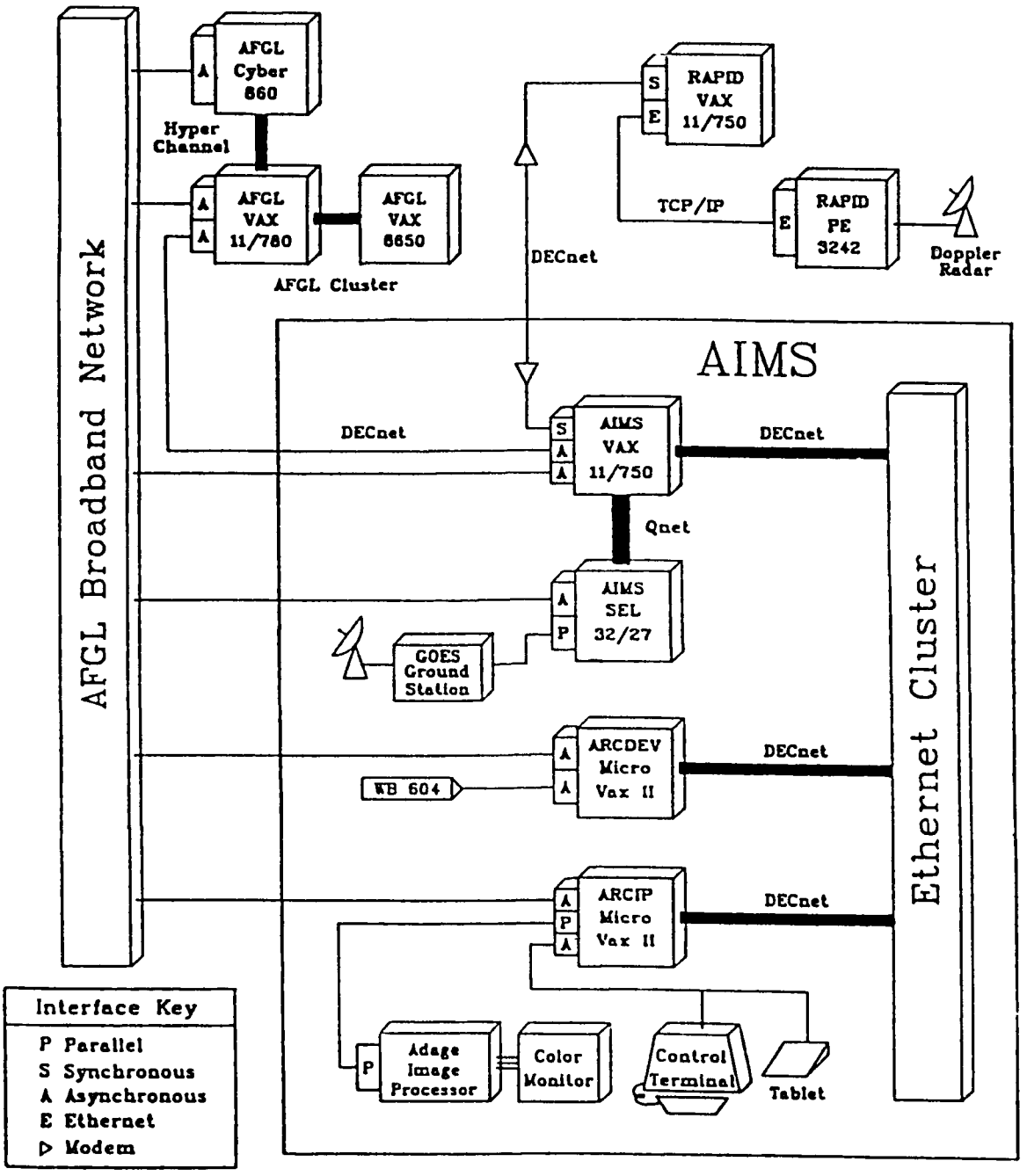



Fig. 2. Schematic Representation of the AFGL Interactive Meteorological System (AIMS).

integrated approach to software development consistent with AIMS.

As currently configured, the AIMS VAX 11/750, in addition to its primary responsibilities, also acts as a gateway between the point-to-point Q-LINK network and AIMS ethernet cluster. In order to maintain a consistent hardware and software base established by the cluster, STX proposed that the 32/27 essentially be "front-ended" by the gateway computer. This configuration has the added benefit that users, both local and remote, would have access to GOES satellite data and scheduling without the need to become familiar with either the 32/27 or the Q-LINK network. To support this configuration, both computers would require an autonomous process to field scheduling and data requests, queue future requests, communicate control information and data using a standard protocol format, and route requested data to the appropriate destination(s). This concept is under consideration.

## 6. Summary

A capability routinely to receive, extract, and store GOES sounding data (excluding data ingest), GOES multispectral imagery, navigation data, and grid information has been established for AIMS. The necessary software has been developed and implemented to achieve this capability. The software is based on functional specifications drawn from system requirements for a fully-functional GOES groundstation. Software design and development and the application of some new concepts have been directed in three broad categories: data storage, data ingest, and data transmission. Under data storage, data structures were defined for three data types: area directory, satellite, and navigation data. Based on these structures database management software was developed which in turn was utilized

as foundation software by applications to perform data-specific functions. The concepts of areas and groups of areas were applied to provide an interrelationship among the three data types that is characterized by fast and efficient single-point access to any data source. In addition, a utility was developed to provide user-friendly configuration of multi-disk systems for storage of satellite data. Under data ingest, the real time ingest sequence was automated to provide continuous acquisition of GOES satellite data without operator intervention. This capability results from the coordination of the ingest sequence, scheduling software, macro expander, and macro generation programs. Finally, system requirements and specifications were generated for the communications aspect of GOES satellite data as it relates to AIMS.

## 7. References


Komajda, R. J., 1987: An Introduction to the GOES I-M Imager and Sounder Instruments and the GVAR RetransmissionFormat. NOAA Technical Report NESDIS 33, U.S. Dept. Of Commerce, Washington, DC.

Gerlach, A. M., ed., 1986: Objective Analysis and Prediction Techniques - Final Report. AFGL-TR-87-0013, Contract F19628-82-C-0023, ST Systems Corporation, 140-148, ADA183450.

Dengel, R., and D. Santek, 1986: A generalized method for storing and processing digital satellite data. Preprints, Second International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Miami; Amer. Meteor. Soc., Boston, 133-138.

C. McIDAS/AIMS Engineering Support

STX has been responsible for day to day engineering support of the AFGL McIDAS and AIMS computer systems. Routine activities have included maintenance of the following electronic equipments: roof-mounted satellite antenna and down link system, Harris computer and peripherals, VAX computers and peripherals, video enhancement systems, archive system, all video display terminals, and all communication equipment required to keep McIDAS and AIMS in operating condition.

Components recently added to AIMS include Color Video Encoder Model 9840, Broadcast Sync Generator Model 2617, and RCA Video Cassette Recorder Model VKT650. Linked to an ADAGE Model 3000, these equipments change the standard RGB signal output to a video monitor to a NTSC composite signal. The NTSC composite signal is recorded on a standard VCR cassette and played through a standard television for briefings or demonstrations.

STX installed a Microwave Antenna on the roof of Building 1102C for the AFGL Lightning Monitoring System. Cables were fabricated and installed to connect the monitoring system to a Zenith Model Z248 computer. The antenna system replaced a costly telephone leased line that connected the system to the outside network.

STX also installed a Digital Mini-Exchange Box in the Atmospheric Sciences Division (LY). The Mini-Exchange Box interfaces two Decmate terminals and a NIU Box with a laser printer. Cables were fabricated and laid in the ceilings of two offices and a hallway. The Exchange Box increases usage and output of the laser printer.

The following major items of equipment required maintenance during the contractual year. The end item is listed, followed by the component(s) repaired or replaced to bring the equipment back to operating condition:

a. TI Printer Model 810
   Transistor Q4 (TIP41C)
   Transistor Q3 (MPS4356)
   Diode CR39 (1N5339)
   Fuse F2 (5 amp)

b. TI Terminal Model 765
   Resistor R332 (15 kΩ)
   Resistor R333 (15 kΩ)
   Diodes CR325, CR326, CR324, CR323, CR334
   SCR diode 9310
   Fuse F301 (2 1/2 amp)
   On/Off switch

b. Hazeltine Terminal Model 20
   CRT controller chip U44

d. Digital Video Storage System
   Cooling fans
   Crystal 22.184 MHZ
   IC's BA029 and BA030 (74LS169)

e. Tektronix Computer Display Terminal Model 4107
   Memory Chips U507 and U607 (4416)

f. Anadex Counter-Timer Model CF-600R
   Capacitor C102 (6200 μf)

g. Ann Arbor Terminal
   Memory chip M2(SY1403A)
   Capacitor C5 (10,000 μf)

h. ADM-24E Video Display Terminal
   Monitor board

i. Write Random Raster Read Memory (WRRRM)
   IC's U39 and U41 (74166)

j. McIDAS Antenna Polarization System
   Relays 2 ea. KHS17A11
   Reinstalled polarization motor

k. McIDAS Antenna Intercom System
   Replaced shorted wiring

l. McIDAS Channel 22 Digital Cursor Drawer
   IC number "D" (N8284)

m. McIDAS Channel 21 Red Enhancement Drawer
   15 volt regulator (MC1468)

n. Aydin PCM Bit Synchronizer Model 350
   Transistors Q1, Q2, Q5
   Resistors R1, R2, R3, R7, R22, R23, R21, R8, R18

o. Harris Laserfax Model 550
   Bearings 2 ea. PN FS1TP7
   Bearings 2 ea. PN 309444-001
   Idler Shaft PN 309716-003
   Spacers 2 ea. PN 309757-004
   Belt PN 6R6-208025
   Roller PN 423566-002

p. Wisconsin AAA Mode Frame Sync V
   Fabricated 26-pin cable with pins 5 and 18
   reversed and pins 13 and 26 reversed.

q. Harris Magnetic Tape Transport Interface Controller
   IC 4B (74170)

r. Harris Laserfax Model 850
   Roller Assembly PN 427446-GO1
   Transport belts 6 ea. PN 134010-002
   Foam Rollers 4 ea. PN 310787-001

## III. CLOUD/PRECIPITATION SYSTEMS: MORPHOLOGY AND MOTIONS

### Three-dimensional Cloud and Precipitation Mapping

#### 1. Introduction

The Ground-Based Remote Sensing Branch of AFGL (AFGL/LYR), with support from STX, is developing a hardware/software system to provide 0 - 30 min forecast locations of cloud/precipitation. These forecasts are intended to assist in assessing the quality of those satellite-to-ground communication links that are adversely affected by the presence of hydrometeors in the atmosphere.

The system that has been developed, the Remote Atmospheric Probing Information Display (RAPID) System, has several components:

* Hardware Development and Acquisition (AFGL)

* Data and Memory Management (STX)

* Support Software (STX, AFGL)

* Analysis (STX)

* Prediction (AFGL)

Organizational development responsibilities are indicated for each component. Much of the development to date has been described in two reports produced jointly by AFGL and STX. Sadoski et al. (1988) gives a fairly complete description of the first three components while Bohne et al. (1988) describes the analysis and prediction techniques that have been tested and ultimately adopted. In this report we will discuss some of these RAPID components. In particular,

the next section will present a very brief hardware overview, identify STX contributions to RAPID through brief summaries and references to previous reports, and describe in detail developments since those reports. Only a very brief description of the hardware is included, and no discussion of the prediction techniques is provided since these two areas are the responsibilities of AFGL personnel. The reader is referred to the above mentioned reports for details concerning these efforts. The following section contains a description of the RAPID software package that has been developed to implement the various analysis components.

## 2. Components of RAPID

### a. Hardware

RAPID is a total hardware and software system that ingests satellite and radar data, applies analysis techniques to the data, and outputs analysis and forecast products, often in the form of color displays. Throughout the development of RAPID the hardware configuration has undergone, and is still undergoing, considerable change. This evolution is well documented in Sadoski et al. (1988). Currently, RAPID consists of a Digital Electronics Corporation (DEC) MicroVAX computer with a powerful auxiliary ADAGE 3000 Image Processor, plus assorted peripherals such as hard disks, color monitors, and color camera.

RAPID is not a stand-alone entity but must interact with other systems to ingest the required data. Satellite imagery is obtained from the AFGL Interactive Meteorological System (AIMS) while radar imagery, in rectangular Cartesian format, is obtained from the AFGL/LYR processing hardware. This has required the establishment of reliable

communication hardware/software linkages among these systems, all of which are described in Sadoski et al. (1988). The main elements in these linkages are DECNet and a "cheapernet."

b. Data and Memory Management

For the forecasting problem it is necessary to process many large images (e.g., as many as 12 volumes of data, each volume containing 3 planes, and each plane containing about 0.125 million elements of 8-bit data--for a total of 4.5 Mbytes of data). To facilitate this processing, images are retained in the ADAGE image memory. This allows more options for the processing of the images; also, it allows the processing to be accomplished with the much faster processor of the ADAGE. To further facilitate access to these data, several management procedures were implemented:

* Each image is stored as a separate file on the VAX disk, with each file containing a detailed file header (Table 1).

* ADAGE memory was subdivided into regions 32 bits deep. These subdivisions were then assigned for specific image types, i.e., radar reflectivity, satellite infrared, satellite visible. See Fig. 1 for a map of this subdivision and Table 2 for allocation details.

* A "file cabinet" that contains information about each of the images stored in the ADAGE memory was established and retained in ADAGE memory (Fig. 1). Information contained in this cabinet includes a subset of the file header depicted in Table 2 as well as information concerning the location of the image within

TABLE 1. **RAPID Image Data Common Header Format**

| Startline Byte | Length (bytes) | Mnemonic | Description |
|---|---|---|---|
| 0 | 2 | ITYPE | Image type; S/R/P/O |
| 2 | 2 | IOPT | Data type; VI/IR/RE/VE |
| 4 | 2 | DTYPE | Data type; IM/GR/PP/RH/CC |
| 6 | 2 | SENSOR | sensor ID |
| 8 | 2 | SATID | satellite ID |
| 10 | 2 | YEAR | YY; year of data |
| 12 | 2 | DAY | DD; julian day of data |
| 14 | 2 | HOURS | HH; hour of data |
| 16 | 2 | MINUTES | MM; minute of data |
| 18 | 2 | SECONDS | SS; second of data |
| 20 | 2 | ULL | upper left corner coordinate, line number |
| 22 | 2 | ULE | upper left corner coordinate, element number |
| 24 | 2 | NLINES | size of image in lines |
| 26 | 2 | NELES | number of elements per image line |
| 28 | 2 | ULATDEG | DDD; ° latitude of ULE |
| 30 | 2 | ULATMIN | M; MM; minutes latitude of ULE |
| 32 | 2 | ULATSEC | SS; seconds latitude of ULE |
| 34 | 2 | ULONDEG | DDD; ° longitude of ULL |
| 36 | 2 | ULONMIN | MM; minutes ° longitude of ULL |
| 38 | 2 | ULONSEC | SS; seconds ° longitude of ULL |
| 40 | 2 | LRES | line resolution, y-scale, km |
| 42 | 2 | ERES | element resolution, x-scale, km |
| 44 | 2 | NUMSCALE | scale value numerator |
| 46 | 2 | DENOMSCALE | scale value denominator |
| 48 | 2 | SHIFT | linear shift factor |
| 50 | 2 | PRF | radar PRF factor |
| 52 | 2 | WAVELENGTH | radar wavelength (scaled) |
| 54 | 2 | CONSTANT | radar constant (scaled) |
| 56 | 2 | LEVEL | radar level |
| 58 | 2 | LATLOCDEG | sensor location, ° latitude |
| 60 | 2 | LATLOCMIN | sensor location, minutes latitude |
| 62 | 2 | LATLOCSEC | sensor location, seconds longitude |
| 64 | 2 | LONLOCDEG | sensor location, ° longitude |
| 66 | 2 | LONLOCMIN | sensor location, minutes longitude |
| 68 | 2 | LONLOCSEC | sensor location, seconds longitude |
| 70 | 58 | (reserved for future use) | |
| 128 | 128 | COMMENT | comments |

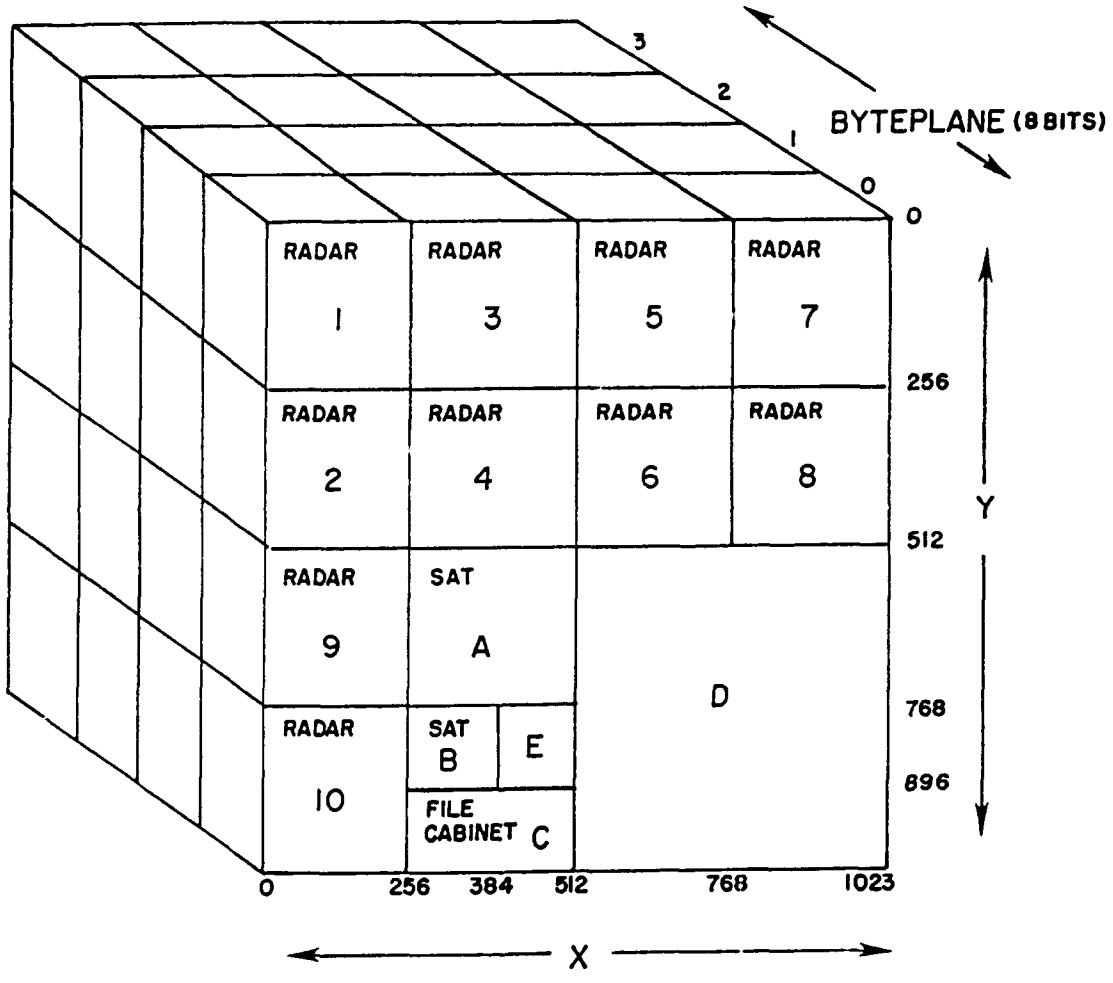



Fig. 1. Three-dimensional Map of First Page of ADAGE 3000 Memory. Table 2 Describes Contents of the Various Storage Areas.

TABLE 2. **ADAGE Display Memory Allocations**

| | |
|---|---|
| 1-10 | are the storage areas for ten 256x256 radar images, reflectivity or velocity. Each byteplane (0, 1, and 2) will store a different height for the time represented by its particular frame. |
| A | is storage for three 256x256 visible satellite images, a different time scan being stored in each byteplane of the quadrant. |
| B | is storage for three 128X128 IR satellite images, similar to A. |
| C | contains all "file cabinet" information. |
| D | may be used for high resolution or large-scale data displays, e.g. as 512X512 km visible satellite data. |
| E | has been reserved for future use. |

the ADAGE and the type of processing that the data have undergone. Table 3 depicts the file cabinet structure.

For more details on these procedures see Sadoski et al. (1988).

c. RAPID Support Software

Support software is defined as software developed by local users or obtained from outside groups that performs a wide range of functions from general data analysis to data handling and display. In general, these are routines that may be utilized by a number of users or functions. As such, they must be easily accessed, user-friendly, thoroughly debugged, and well documented. To facilitate their use, support software has been organized according to function, e.g., display tools, analysis routines for the VAX, etc., and stored in libraries. Detailed discussion of the libraries can be found in Sadoski et al. (1988).

d. Analysis Techniques

One of the major goals of the STX effort is to develop mapping techniques that ultimately provide useful parameters for the forecasting of the location of precipitation and/or cloud. The approach is to extract contours from the files being monitored, e.g., radar reflectivity or satellite-derived cloud brightness or infrared temperature. From these contours characteristic parameters are derived that lend themselves to simple forecasting techniques.

To perform the required analysis, there are several steps that are required, namely:

* Data smoothing

TABLE 3. **RAPID File Cabinet Organization**

| Starting Byte | Mnemonic | Description |
|---|---|---|
| 0-18 | | (Same as in RADAR/Satellite Data File Header) |
| 20 | LCORNER | upper left corner coordinate, line number relative to original image data array |
| 22 | ECORNER | upper left corner coordinate, element number relative to original image data array |
| 24 | LRANGE | size of displayed image in pixel lines |
| 26 | ERANGE | number of elements per displayed line |
| 40 | LRES | image resolution, y-length scale in km |
| 42 | ERES | image resolution, x-length scale in km |
| 44 | XWINDOW | element origin within display memory |
| 46 | YWINDOW | line origin within display memory |
| 48 | INUMBER | nth image number of this series |

* Contour extraction

* Contour filtering

* Motion/evolution estimation.

All of these steps, with the exception of contour filtering, have been thoroughly discussed in Bohne et al. (1988) and will only be briefly summarized here. However, a more detailed discussion of the contour filtering step will follow.

Data smoothing is required to eliminate noise due to missing lines (a common occurrence with satellite imagery), spurious data points, and "insignificant" small scale fluctuations. This type of editing is required to facilitate contour extraction. Several techniques have been tested and documented (Bohne et al., 1988).

For contour extraction, several representations were investigated (Bohne et al., 1988) and the Freeman Chain Code representation was the one adopted. This code consists of a Cartesian coordinate pair for the first element of the contour and a series of directional codes that point between contour locations. This representation adequately describes the contour while markedly reducing the amount of data to be processed. Subsequent analysis techniques all utilize this contour representation.

Once the contours are extracted and stored as Freeman Chain Codes, it is often necessary to perform filtering operations, even if the data have been previously filtered. A number of techniques for filtering have been explored, including:

* Simple averaging of the directional codes of the Freeman Chain Code

* Weighted averaging of directional codes where diagonal codes are weighted more than vertical or horizontal codes

* Averaging of Cartesian coordinates derived from chain codes.

None of these techniques produced the desired results, namely, smoothing the small scale fluctuations while retaining the larger scale features intact. As a consequence, a more complex pattern recognition scheme was adopted. Basically, the technique constructs the longest straight lines that will maintain the integrity of the overall shape of the contour. To accomplish this a three-pass filtering scheme is applied. On the first pass, a simple vector addition of two directional codes is performed. This results in one or two new "averaged" codes, depending on the vectors being added, to replace the two being added. This pass removes any point-to-point fluctuations. The second pass then performs vector additions of three or more directional codes to produce a new series of "averaged" codes, ones where the scale of fluctuation being removed is increased with the number of codes being added. The result of this type of averaging is a contour consisting of straight line segments with lengths of at least that corresponding to the maximum number of codes being averaged. It should be noted that the filtering in these two passes is basically a vector addition process. There are occasions where adjustments to the resultant vectors are required to ensure passage through grid points. In an attempt to further increase the length of straight line segments, the resultant codes are adjusted by merging short segments with the same directional codes and with

separations of no more than some threshold, such as 6 codes. This type of consolidation results in longer straight line segments in general while maintaining the integrity of the original overall contour.

The net effect for this filtering scheme is displayed in Figs. 2a and b. In Fig. 2a are plotted two contours of infrared temperatures derived from data that have been filtered with a 5 x 1 median filter to eliminate line "noise" and a 3 x 3 median filter to eliminate point "noise." There is still significant small scale structure in these contours, too much to be monitored with any precision. Application of the unique filtering process described above results in the contour in Fig. 2b. Here the larger scale features of the contour are retained while the small scale is eliminated. These contours (Fig. 2b) will lend themselves much more readily to the contour matching routines that are used in the motion/evolution step of analysis.

Once the contours are extracted and smoothed for at least two successive data sets, an assessment of motion and evolution may be made. This is accomplished by breaking down the contours into segments and then associating segments in successive scans. Basically, two approaches were adopted; namely, to break down the contours into straight line segments and to divide contours into segments with the same number of elements. Once breakup is achieved the segment matching is performed either through simple association with regard to segment orientation and length for straight line segments or through a correlation analysis based on least squares fitting of two segments. These techniques are described in detail in Bohne et al. (1988).

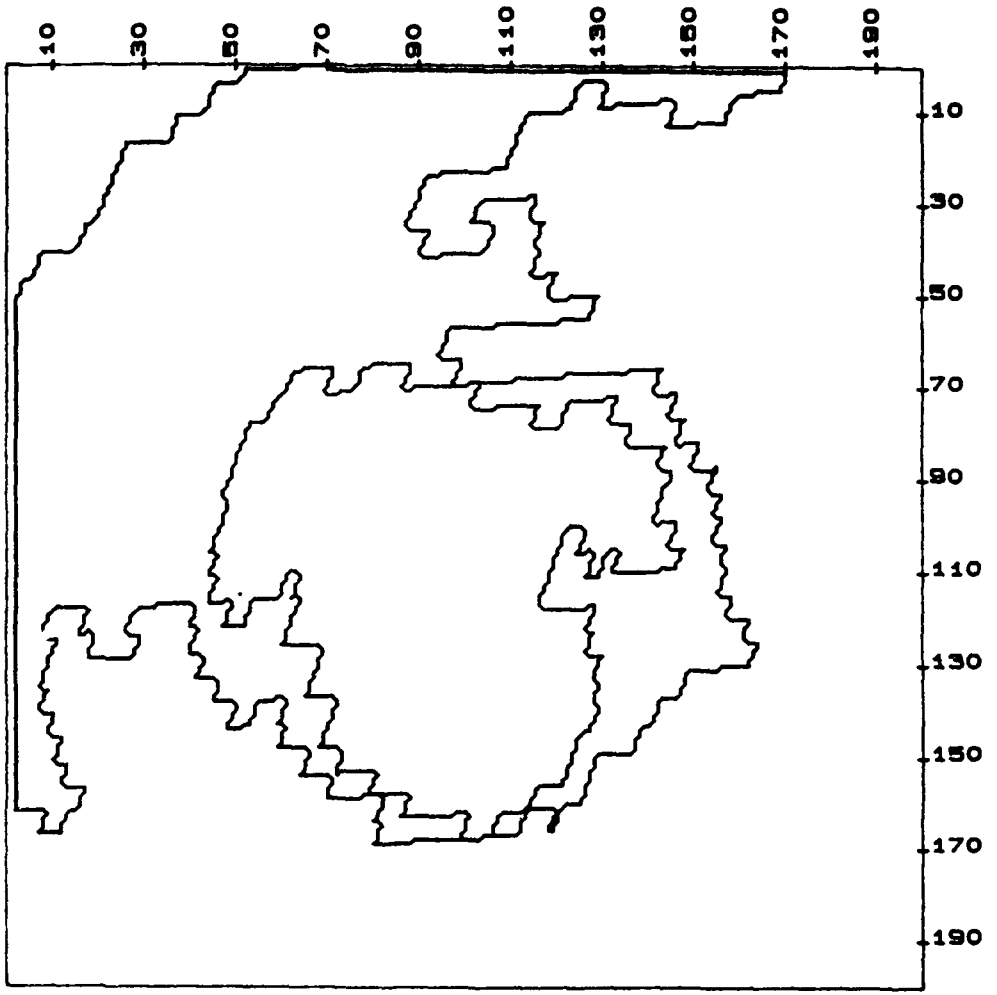



Fig. 2a. Contours of Infrared Temperature for Two Values for 1500 EST 25 September 1985.

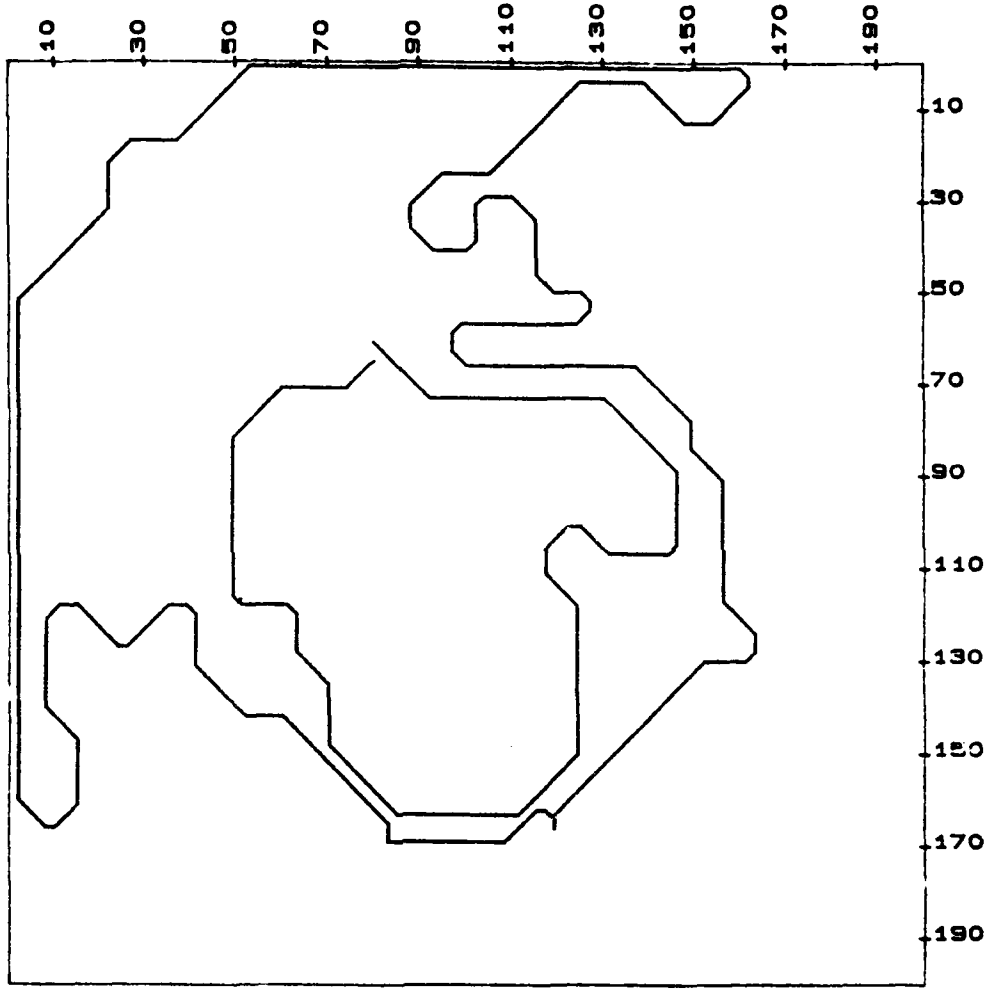



Fig. 2b. Smoothed Contours of Infrared Temperature for Two Values for 1500 EST 25 September 1985.

## 3. RAPID Software Package

An extensive interactive software package has been developed that utilizes the elements outlined above; namely, RAPID support software, data/memory management, and analysis techniques, with provision to easily add the AFGL-developed forecasting modules. The elements of this package are depicted in Fig. 3. This package was derived from the prototype FTEST package described very briefly in Sadoski et al. (1988). Software is written in FORTRAN and C with some routines in ICROSS, the C-subset language used for the ADAGE. The main driver for this software package is the module RAPID that interactively determines the type of processing that the user desires. Control is then transferred to the requested module, which then queries the user for parameters and then proceeds with the required processing. Upon completion of this processing, control is passed back to the RAPID module. Each module accesses and creates data in the ADAGE memory or the VAX disk as needed. Following is a brief description of the type of processing that occurs in each module for which STX has responsibility. The Prediction and Contour Reconstruction modules are being developed by AFGL and will not be discussed here.

### a. Ingest

This module selects the data to be processed and transfers them to the ADAGE memory. It automatically extracts the file header information and stores the required elements along with the position elements listed in Table 2. In addition, data already stored in the ADAGE memory may be saved to VAX disk, moved to a new byteplane, replicated, and/or reduced in size (i.e., decreased in terms of number of data points per image). This module must be accessed before any image data processing is performed regardless of whether the processing software resides in the VAX or ADAGE.

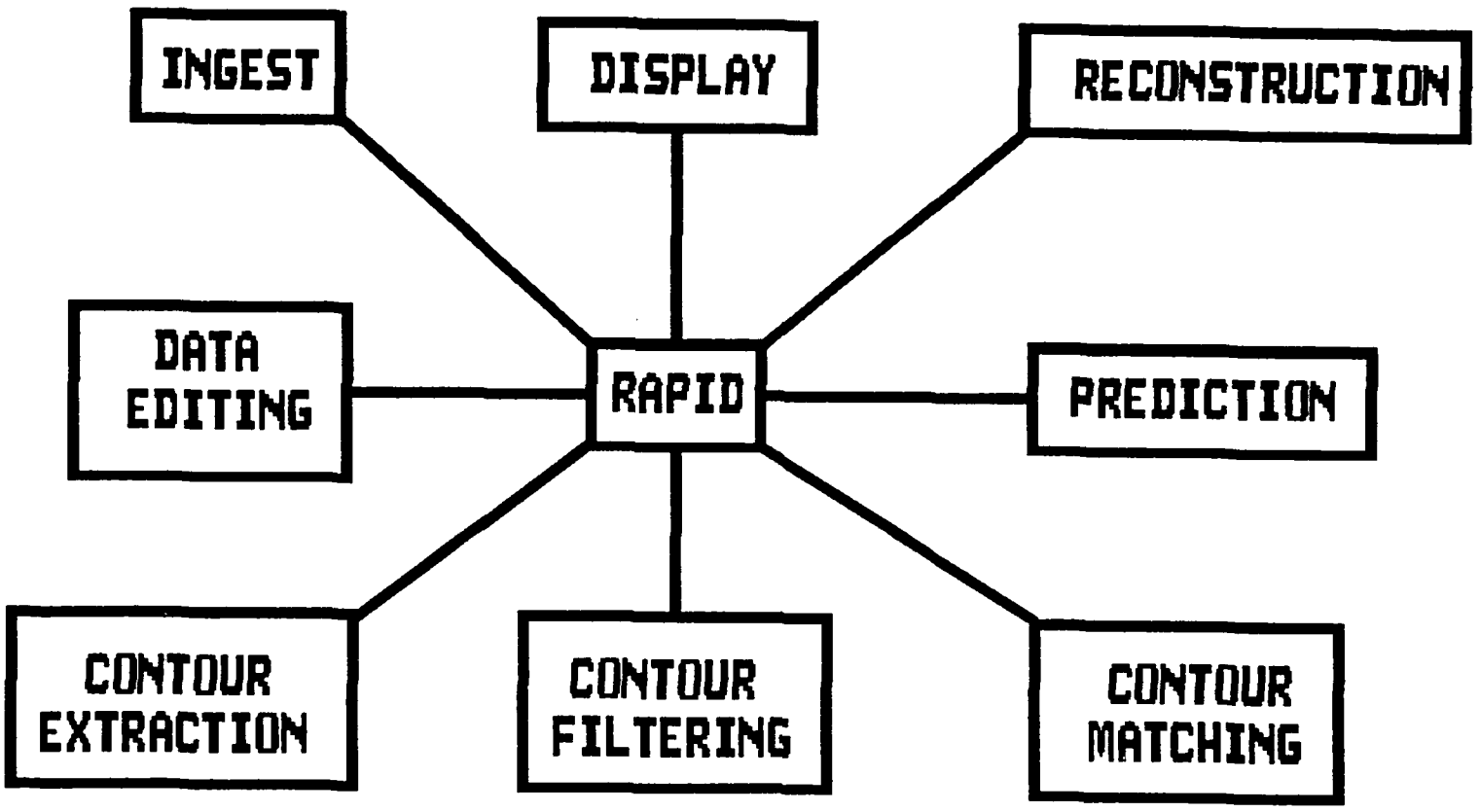



Fig. 3. Functional Diagram of Processing Components in RAPID Software Package. Solid Lines Connecting Perimeter Boxes with RAPID Indicate Lines of Two-way Communication.

b. Data Editing

In this module "noise" and small scale fluctuations are removed from the data using filters discussed in Bohne et al. (1988). All processing is interactive, such that one may vary the filters, the parameters within the filters, and the number of times each filter is applied. For this operation, all data must be resident in ADAGE memory.

The filters that have been implemented are:

* Median 5 point line filter
* Median 9 point line filter
* Median 3 x 3 box filter
* Median 5 x 5 box filter
* Low pass filter
* LaPlace filter.

The software for these filters operates on the ADAGE, the most efficient processor for this type of operation.

c. Contour Extraction

Contours for user-specified data values are extracted from images stored within the ADAGE. The Freeman Chain Code formulation for contours as discussed in Bohne et al. (1988) has been adopted for this processing. Single or multiple valued contours may be extracted from the entire image or from a windowed region within the image. The resultant codes are stored as data sets within the ADAGE that may later be downloaded to the VAX disk for long-term storage.

d. Contour Filtering

Software has been implemented to perform the contour filtering scheme described in the Analysis Techniques subsection above (2d). This routine allows the user to specify the maximum number of points (<8) that will be considered in the smoothing process.

e. Contour Matching

Software to break up the contours into segments and to perform segment matching for the assessment of motion and evolution is included in the Contour Matching module of Fig. 3. As in the other modules, the software is implemented interactively and assumes that the Freeman Chain Codes for the contours are already stored on the ADAGE. Output from this module is in the form of another array stored in the ADAGE memory that includes:

* Segment identifier
* Segment length
* Segment mean orientation
* Segment starting point
* Distance from image reference point to segment starting point
* Segment identifier for associated segment in next image.

These attributes are stored on the ADAGE but may also be dumped to the VAX disk. This file, or collection of files if a number of images have been processed, and the files containing the chain codes are then available for processing in the Prediction and Contour Reconstruction modules. These latter two modules are to be provided by AFGL.

f. Display

At all stages software may be displayed on the color monitor available for the ADAGE. Images may also be transferred to the Dunn camera that is connected to the output of the ADAGE. In this way hard copy (35mm film) may be obtained of images. For contour plots and other non-image analysis displays, plots may also be generated on the Hewlett-Packard pen plotter.

4. Summary

STX has derived a number of analysis techniques that will operate on images of radar or satellite data and has implemented them in a software analysis system to provide estimates of the motion and evolution of cloud/precipitation regions. This software package emphasizes user interaction as opposed to speed. However, the modular nature of the software and the expansive use of the memory within the ADAGE image processor should allow a relatively easy transition to a real time analysis system.

5. References

Bohne, A. R., F. I. Harris, D. R. Egerton, P. A. Sadoski, 1988: Short Term Forecasting of Cloud and Precipitation. AFGL-TR-88-0032, Air Force Geophysics Laboratory, Hanscom AFB.

Sadoski, P. A., A. R. Bohne, D. R. Egerton, F. I. Harris, 1988: The Remote Atmospheric Probing Information Display (RAPID) System. AFGL-TR-88-0036, Air Force Geophysics Laboratory, Hanscom AFB.

# IV. AUTOMATED GLOBAL CLOUD CLIMATOLOGY

## Automated Global Cloud Climatology

### 1. General

#### a. Introduction

The goal of this task was to develop a methodology such that the climatology of total sky cover can be accurately and concisely stored on a computer and rapidly retrieved for a variety of applications. With the Burger database (Burger, 1985) as a starting point, other databases were obtained and archived. A brief list of these databases is given in Table 1. Details of these databases are described in Willand (1988). Additional information on this task can be found in the prototype program CloudZ written for the Zenith Z-100 computer and available on a floppy disk.

#### b. Burger Distribution

Total sky cover is a mixed distribution; that is, it consists of both discrete points of probability (clear and overcast) and continuous probability density between clear and overcast. Although the probability is given as, for example, 2/10, what is really meant is the probability of sky cover between 0.15 and 0.25. The probability of exactly 0.2000... is infinitesimal--a probability density. On the other hand, there is a true probability of clear. Most attempts to approximate sky cover, for example, the beta distribution, flounder on this point.

The Burger distribution (Burger, 1985) provides true probabilities at clear and overcast while providing probability densities between these two values. The parameters of the Burger distribution are the mean and the

## TABLE 1. SKY COVER DATABASES

Burger data set: (Jan Apr Jul Oct at four local times)
- RUSSWO 266 sites; mostly 10-year POR
- NIS 1846 sites
- NAV Atlas 222 areas
- WHOI N&S Atlantic $10^{o}$ by $10^{o}$ Areas, 1942-1972
- SSMO 443 locations near ports
- World Survey 125 sites in Soviet Union

DOE/NCAR Global Cloud Clim. $5^{o}$ by $5^{o}$ areas, 1971-1981
- FGGE $2.5^{o}$ by $2.5^{o}$; 1979

DATSAV Clusters: 65 sites, hourly obs for correlation

3DNeph 1/4 mesh smoothed: satellite/surface obs 1977-1982

## TABLE 2. CORRELATION FUNCTIONS APPROX. EXP (-x) NEAR THE ORIGIN

| MODEL | $s = d / r$ FUNCTION $\sigma \leq 1$ | $\sigma > 1$ | $\sigma = s / m$ MODEL FACTOR m |
|---|---|---|---|
| Model B | | | |
| | $(2/\pi)\{\arccos(\sigma)-\sigma\,SQRT(1-\sigma^2)\}$ | 0 | 128 |
| Sawtooth Models | | | |
| 2D | $1 - (12/\pi)\,\sigma + 3\,\sigma^2$ | $+(24/\pi)\{\arccos(1/\sigma)-SQRT(\sigma-1)\}$ | 340 |
| 3D | $1 - 3\,\sigma + 2\,\sigma^2$ | $-6(\sigma-1)^2/\sigma$ | 260 |
| 4D | $1 - (8/\pi)\,\sigma + 3/2\,\sigma^2$ | ? | 217 |

d is a separation distance between points
r is a scale distance - a function of the element being modeled
d and r must be in the same units, e.g., km
s is a dimensionless standardized distance (1=1 scale distance)
m is a dimensionless scaling factor for each model
$\sigma$ is a model dependent dimensionless distance used within a model

scale distance. The mean is self-explanatory. The scale distance parameter requires some explanation.

The Burger distribution is derived from multiple simulations of the Boehm Sawtooth Wave (BSW) model. The BSW consists of a number, i.e., 12, of sawtooth fields. Each field has its phase and orientation randomly selected. The amplitude of each wave is set to one, so that the height at any arbitrary location can be any value between zero and one. Since the sawtooth has a uniform slope, the probability of any height is uniform [0,1]. The summation of heights from several waves is equivalent to the summation of uniform random variables. This sum quickly approaches a normal distribution. Thus any point in a BSW simulation has a value that approximates a value chosen from a normal distribution.

In addition, any two close points will have almost the same value so that there is, when the simulation is repeated many times, a high correlation between the values at the two close points. For two points farther apart, the correlation will be lower. The value of correlation for various separation distances specifies a correlation function. This correlation function resembles EXP(-x) near the origin. Several weather elements (but not all) have been observed to have this shape of correlation drop off (Bertoni and Lund, 1964). The sawtooth correlation function needs a scaling factor to match the observed correlation functions. This scaling factor is designated as scale distance--the separation distance at which correlation drops to 0.99 approximately. The approximation is necessary since several models are involved, including the older non-sawtooth Model B (Gringorten, 1979). The exact formulas for sawtooth correlation are given in Table 2.

Scale distance was originally defined for Model B as the distance correlation equals 0.99. The model factor m was rounded from 127.3+ to 128. This value has continued to be used. Other model factors were determined by minimizing the maximum difference between each model's correlation and Model B in the range 1 to 0.1.

In Model B, $\sigma = 1$ is the radius of the averaging disk. In the sawtooth models, $\sigma = 1$ is the sawtooth wavelength. In terms of correlation dropoff, $\sigma$ is not equivalent in different sawtooth models but s is.

c. Burger Areal Algorithm (BAA)

Burger and Gringorten (1983) generated numerous BSW simulations and kept track of the coverage of various sized areas for several mean values. These were plotted and then painstakingly fitted with algebraic algorithms.

To obtain flexibility, Burger followed Gringorten's use of a standardized area using the dimensionless variable s:

$$s = \sqrt{A}\ /r \qquad (1)$$

where A is the area and r the scale distance. s can be thought of as the side of square that has the equivalent standardized area. Alternately, Z, a variable equal to the logarithm of s to the base 2, is used:

$$Z = \log_2 s = \ln s/\ln 2. \qquad (2)$$

For a constant r, an increase of Z by one (e.g., Z = 3 to Z = 4) means that the length of a side of an equivalent square is doubled.

The BAA originally was fitted to results from 2D sawtooth simulations. Burger also tested the BAA against 3D sawtooth simulation results and could find no significant difference. This result was expected because of the similarity of their correlation functions. The slight difference in their correlation functions suggests that there should be a slight difference in their coverage distributions. However, the change is apparently too small to be detected by simulation.

2. Errors in Using the Burger Distribution

a. Sky Dome As a Flat Surface

Gringorten (1982) made use of the notion that the sky dome could be approximated as a flat surface. His estimate of a 2424 $km^2$ sky dome was based on several references in the literature giving the radius of the sky dome as 15 nautical miles. Clearly such a concept has a certain arbitrariness to it: How close to the horizon can clouds be observed? What are the heights of the clouds? Nevertheless, when Burger (1985) fitted observed sky cover distributions using the Burger Areal Algorithm - BAA, he found a remarkably good fit.

The good fit does not, however, imply that 2424 $km^2$ is a good estimate. The true parameters of the BAA are the mean and the parameter Z. In fitting an empirical distribution, Z is found first. Next, Eq. (2) is used to determine a scale distance given an area. If the area is in error so will be the scale distance. However, when the erroneous scale distance is to be tested, it is put into Eq. (2) where its error is cancelled by error in area and the proper Z is calculated. Thus while the BAA gives a good fit to sky cover distributions, the resulting scale distance is

suspect in any application other than total sky cover as seen by a surface observer.

b. Distorted and Bad Data Effects on Scale Distance

Sky cover data, particularly summarized data, come in many forms:

a. In tenths, 11 categories: 0-.05,.05-1.5, . . ., .95-1

b. In eighths, 9 categories: 0-.0625,.0625-.1875, . . .,.9375-1

c. Airways, CLR,SCT,BKN,OVC; 0-.05,.05-.55, .55-.95,.95-1

d. NAVAIR Atlas, probability $\leq$ 2/8 and probability $\geq$ 5/8

e. DOE/NCAR Atlas, mean and standard deviation

f. 3DNeph Summary, 21 categories: clr,0-.05,.05-1, . . .,.95-1 with 9 point smoother applied to 1/8 mesh (appr. 25 nm on a side) satellite data taken at various look angles.

For each a different algorithm has to be used to estimate scale distance. Every algorithm fails under some conditions. Small data samples and obviously bad data are responsible for many of the failures. One expedient method of dealing with small samples is simply to delete them from the analysis. But the fact remains that none of the algorithms is robust with respect to small samples.

Surprisingly, the standard deviation algorithm fails when the standard deviation is small. Other algorithms fail with what appear to be acceptable sky cover distributions.

These failures can be traced to two causes. First, the algorithms were designed with only the asymptotic (infinite size sample) distribution in mind. Second, the BAA algorithm itself has faults that are acceptable when it is used to find a probability but cause considerable problems when it is used in an inverse fashion to find scale distance.

Given these weaknesses, it was decided to delay any further scale distance processing until a reliable algorithm was developed.

## 3. Objective Analysis

### a. Purpose

Specification of the climatological probability of sky cover anywhere on the globe, any time of day and time of year, is a complex problem. The problem is first simplified by specifying sky cover with two parameters--the mean and scale distance. At a point these parameters are each compactly specified by a two-dimensional Fourier analysis. Next the Fourier coefficients are analyzed over the globe by a mixture of Fourier/Legendre analysis.

In calculating this large a number of coefficients, high order matrices are inverted, and great care must be taken with numerical calculations. Much of the following section is concerned with methods that minimize numerical instability.

b. Mean and Scale Distance

The mean and scale distance are orthogonal; that is, knowing one gives no information about the other. This orthogonality means that they can be fitted separately. That is not true of the parameters of many other distributions. For example, the parameters of the Weibull distribution interact, and great care must be taken when fitting them so that a small error in one parameter is not greatly magnified by a small error in the other.

Even in the Burger distribution, or in any double-bounded distribution, the mean and the standard deviation are related by an inequality. For a variable with a range limited to zero to one, the standard deviation cannot be larger than $\sqrt{[\mu(1-\mu)]}$ where $\mu$ is the mean. Thus if the standard deviation and the mean are calculated separately, an invalid combination could result.

The valid range of the mean is zero to one. With regression an invalid mean is possible. If the equivalent normal deviate (END) of the mean is used, all values are valid, and this problem is avoided. Similarly, a negative scale distance is invalid. If Z as defined in Eq. (2) is used, all values are possible, and invalid results cannot occur. Further, since both the END of the mean and the value Z are order 1 in magnitude, numerical roundoff is minimized.

4. Fourier Analysis in Time of Day and Time of Year

Time of Day (TOD) and Time of Year (TOY) are circular variables, which makes them ideal for Fourier analysis. Since the mean and scale distances are handled separately but identically, only the scale distance will be considered here.

Specifically, what is analyzed as explained above is the value Z.

Situation 1: Data for all months and all hours of the day are available.

First Z for a given location is analyzed for all the hours of one month at a time. The result for January (subscript 1) is:

$$Z_1(h) = A_{01} + A_{11}\cos(h) + A_{21}\sin(h) + A_{31}\cos(2h) + A_{41}\sin(2h) + A_{51}\cos(3h) + A_{61}\sin(3h) + \ldots + A_{12\ 1}\cos(6h), \quad (3)$$

where h is the time of day in radians, h = (hour/24)*2π, A's are determined from data as described below, or in a more compact notation:

$$Z_1(h) = \Sigma\ A \quad F\ (h) \qquad i = 0 \text{ to } 23 \quad (4)$$

$$\text{where } F\ (h) = 1 \qquad \text{if } i = 0 \quad (5)$$
$$= \cos[((i+1)/2)h] \qquad \text{if } i \text{ odd}$$
$$= \sin[(i/2)h] \qquad \text{if } i \text{ even and} > 0.$$

The coefficients A are determined by making use of the orthogonal property of sines and cosines:

$$A_{i1} = \Sigma\ Z_{ik}\ F_i\ [(k/24)*2\pi], \qquad k = 0 \text{ to } 23. \quad (6)$$

$Z_{10}$ is the Z value of the observed scale distance for January at 0000L. In Eq. (6), $Z_{1k}$ refers to the value specified by data. In Eq. (3) or (4), $Z_1(h)$ is the calculated value which will be identical (rounding error is very small) to the observed $Z_{1k}$ on the hour. In Eq. (4) h need not equate to a value on the hour; h can vary

continuously to allow the scale distance to be calculated at any time, e.g., 0841L.

In the same way the 24 values of $A_{12}$ for February are calculated and so on for the other months.

Next the coefficients themselves undergo a Fourier analysis. The $A_{oj}$, j = 1 to 12 are fitted with 12 Fourier coefficients:

$$B_{oj} = \Sigma\ A_{ok}\ F_j\ [\{(k-.5)/12\}*2\pi], \qquad k = 1 \text{ to } 12. \qquad (7)$$

Similarly, $B_{1j}$ is fitted using the $A_{1j}$ coefficients, $B_{2j}$ using the $A_{2j}$ coefficients, etc.

The end result is the 288 B coefficients, instead of the 288 (12 months times 24 hours) data points. The advantage is that Z can now be calculated for any time, not just on the hour and for any day of the month. In addition, not all 288 coefficients need be used to obtain an accurate fit. Indeed, by truncating the higher order Fourier coefficients smoothing is introduced which can actually improve the estimates. See Fig. 1.

If data are not available for every hour but are in <u>regularly spaced</u> intervals, i.e, every three hours, Eq. (5) can be used by limiting k to only those values for which there are data.

<u>Situation 2</u>: Data for some hours or some months are missing.

When some data are missing, the Fourier functions are no longer orthogonal and the simple summation formulas Eqs. (6) and (7) can not be used. In this case multiple

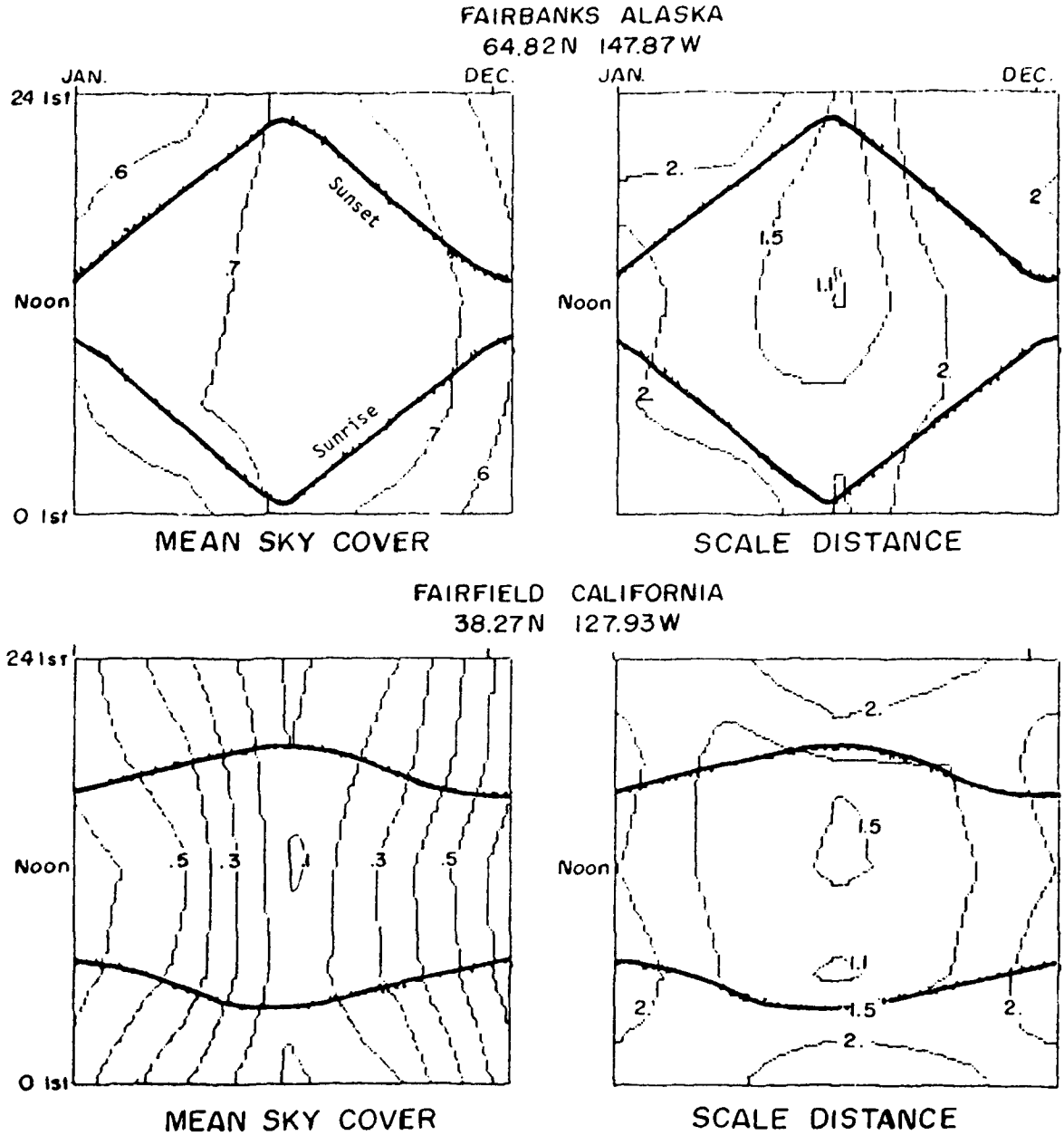



Fig. 1. Mean Sky Cover and Scale Distance Analyzed by Two-dimensional Fourier Analysis. Heavy Lines Are Times of Sunrise and Sunset. Note Strong Relation between Length of Day and Small Values of Scale Distance.

regression is used to find the B coefficients. Predictor points P for the multiple regression are given by:

$$P_{ij} = F_{j \text{ MOD } M}[(h_i/24)*2\pi] * F_{\text{INT}(j/M)}[(J_i/365)*2\pi], \quad (8)$$

where i is an index given to each data point that is available,

j = 1 to the total number of predictors (which cannot exceed the total number of data points)
M is the number of Fourier functions in time of day
$h_i$ is hour of the ith data point
$J_i$ is the Julian day (e.g., 1 = 1 Jan, 32 = 1 Feb, 365 = 31 Dec).

For monthly data, the day half way through the month is used for J. But Eq. (7) allows the flexibility of using bi-monthly, semi-monthly, or any grouping of the data. Leap years are neglected in this scheme.

The predictand values $Z_i$ are the observed values of Z at $h_i$ and $J_i$. The coefficients $B_j$ are given by the matrix equation:

$$\underline{B} = \underline{Z}(\underline{P}^t\underline{P})^{-1} \quad (9)$$

The Choleski method with standardization is used to solve Eq. (8) numerically. This method is described further in the next section. Once $\underline{B}$ is found, Z can be calculated for any time of day and for any Julian day even though data are missing for that particular time/day.

## 5. Fourier/Legendre Analysis (A6060)

### a. Selecting an Analysis Scheme

Analysis on the globe presents certain problems. One way to do such an analysis is to use spherical harmonics which use a combination of Fourier and Legendre functions. However, spherical harmonic analysis requires 2N data points to calculate N coefficients (Colombo, 1981). This number is in contrast to the two-dimensional time of day, time of year Fourier analysis described above which only takes N data points to fit N coefficients.

In addition, sky cover data in the polar regions differ from lower latitude data in three ways. First, surface observations are very sparse. Second, in some months there is no sunrise nor sunset. So that if length of day or time of sunrise is to play a part in synthesizing diurnal effects, these regions have to be handled separately. Third, satellite cloud discrimination data suffer from the constant snow and ice background and from the extreme inversions.

It thus appears practical to analyze the polar areas separately. This discussion will be constrained to analysis between latitudes 60 South and 60 North. The resulting analysis scheme is called A6060.

### b. Numerical Description of A6060

Since as above the mean and scale distance are handled identically but separately, only the analysis of Z, the surrogate for scale distance, will be described.

First consider a set of N data points $Z_i$ for i = 1 to N. These points may be located on a grid or at arbitrary

locations $X_i$ for longitude and $Y_i$ for latitude in degrees. These are converted to Fourier ($-\pi$ to $\pi$) and Legendre (-1 to 1) ranges by:

$$x_i = (X_i/180)*2\pi, \text{ and} \tag{10}$$

$$y_i = \sin[Y_i/60)*\pi/2]. \tag{11}$$

The reason for the sine is explained below. Here just note that taking the sine approximates an equal area projection in x and y.

The predictor points are given by:

$$P_{ij} = F_{INT(j/M)}[x_i] * L_j \bmod M[Y_i], \quad j = 1 \text{ to } N_p \tag{12}$$

where F is the Fourier function as in Eq. (5)

M is the order of the Legendre polynomial. There are M+1 Legendre terms, counting the constant term.

$N_p$ is the total number of predictor terms. If $N_f$ is the number of Fourier terms, then $N_p = (M+1)N_f$

$L_j$ [y] is the jth Legendre polynomial:

$$L_o = 1,\ L_1 = y,\ L_2 = (3y^2 - 1)/2,$$
$$L_3 = (5y^3 - 3y)/2,\ L_j = yL_{j-1} - L_{j-2}$$
$$+ yL_{j-1} - (yL_{j-1} - L_{j-2})/j. \tag{13}$$

The exact sequence of the recurrence Eq. (13) is for numerical stability. The matrix equation for the calculation of coefficients is as in Eq. (9). However, some details of the matrix solution are in order.

c. Matrix Inversion for A6060

First the variables are standardized; that is, for each predictor and for Z, the mean is subtracted and the result divided by its standard deviation. To prevent numerical instability, the mean is found first. Then the mean is subtracted from the variable and the subtractand summed and divided by N to calculate the standard deviation. If the short form of calculating standard deviation is used, numerical roundoff often causes a negative square root error.

Since the variables are standardized, the term ($\underline{P}^t\underline{P}$) is the correlation matrix $\underline{C}$ between the predictor terms. This $N_p$ by $N_p$ square matrix is symmetrical with ones along the diagonal. Calculating this matrix is the longest part of the computation, longer even than the actual matrix inversion operation. As with most simultaneous linear equation solutions, the correlation matrix inverse is not directly found. Instead, the lower triangular equivalent matrix T is calculated via the Choleski square root algorithm (Westlake, 1968). Then the inverse of the lower triangular matrix is calculated, followed by a two-part back substitution.

For standardized variables the resulting coefficients are sometimes called beta coefficients and their magnitude is related to the importance of a particular predictor. To be used in synthesizing the Z field, the beta coefficients are converted to regression coefficients by multiplying by the standard deviation of the predictand (Z) and dividing each by its predictor standard deviation.

Given the regression coefficients, Z can be calculated for any point between 60 South and 60 North.

d. Error Analysis

A series of simulation runs was made to test the accuracy and robustness of the A6060 algorithm. First, synthetic Z values were generated from a known field. Next, these values were given to the A6060 algorithm. The resulting coefficients were then used to make a calculated field which was then compared to the known field.

The first series of simulations showed an interesting phenomenon. While the calculated field fitted the simulated data points well enough, it showed too much variation for high latitudes. An analysis of the Legendre functions showed that they have much higher first derivatives near their end points (60 South and 60 North in the case of A6060) than near the center of their range (the equator). Several transformations were tried, including the cosine which is often used with Legendre functions. The cosine transformation only made the change in variability from center to end worse. The sine transformation as shown in Eq. (11) equalized the variation in the Legendre polynomials quite well.

When the data locations were spread out over the globe (60 South to 60 North) the results were amazingly accurate. For over 100 predictors, the values were calculated to 7 decimals of accuracy.

As data locations were restricted to smaller and smaller areas, the accuracy fell off; for example, calculating values for Europe given only data for North America. Under severe data limitations, the algorithm would fail--usually in the Choleski square root section. This kind of failure indicates that the correlation matrix is singular, or more to the point, a given set of data locations can only support this type of analysis up to a

certain number of predictors. The values at the locations do not limit the algorithm, which will give good results even for outlandish observed values. The limitation is in the spread of locations or, more specifically, in having a good spread of predictor values in Eq. (12).

e. Examples of A6060 Analysis

Data for January 0000L were analyzed for the mean and scale distance. An example of the analysis is shown in Fig. 2. This example was programmed for the Zenith Z-100 as a prototype of a complete sky cover climatology. The prototype rapidly produces an estimate of the sky cover distribution for any location between 60 South and 60 North. Other sky cover related statistics, such as the probability of a cloud-free line-of-sight, are also calculated.

When the output of the prototype is compared with observed sky cover distributions, the result is quite good. See Fig. 3.

6. Summary

An effective methodology for analyzing sky cover has been developed. The analysis makes use of the Burger distribution with its two parameters, mean and scale distance, to specify any value of the distribution. The mean and the scale distance are analyzed in time of day and time of year by a two-dimensional Fourier analysis. They are analyzed in space by a Fourier/Legendre analysis. The resultant scheme can synthesize a complete sky cover distribution for any location between 60 South and 60 North.

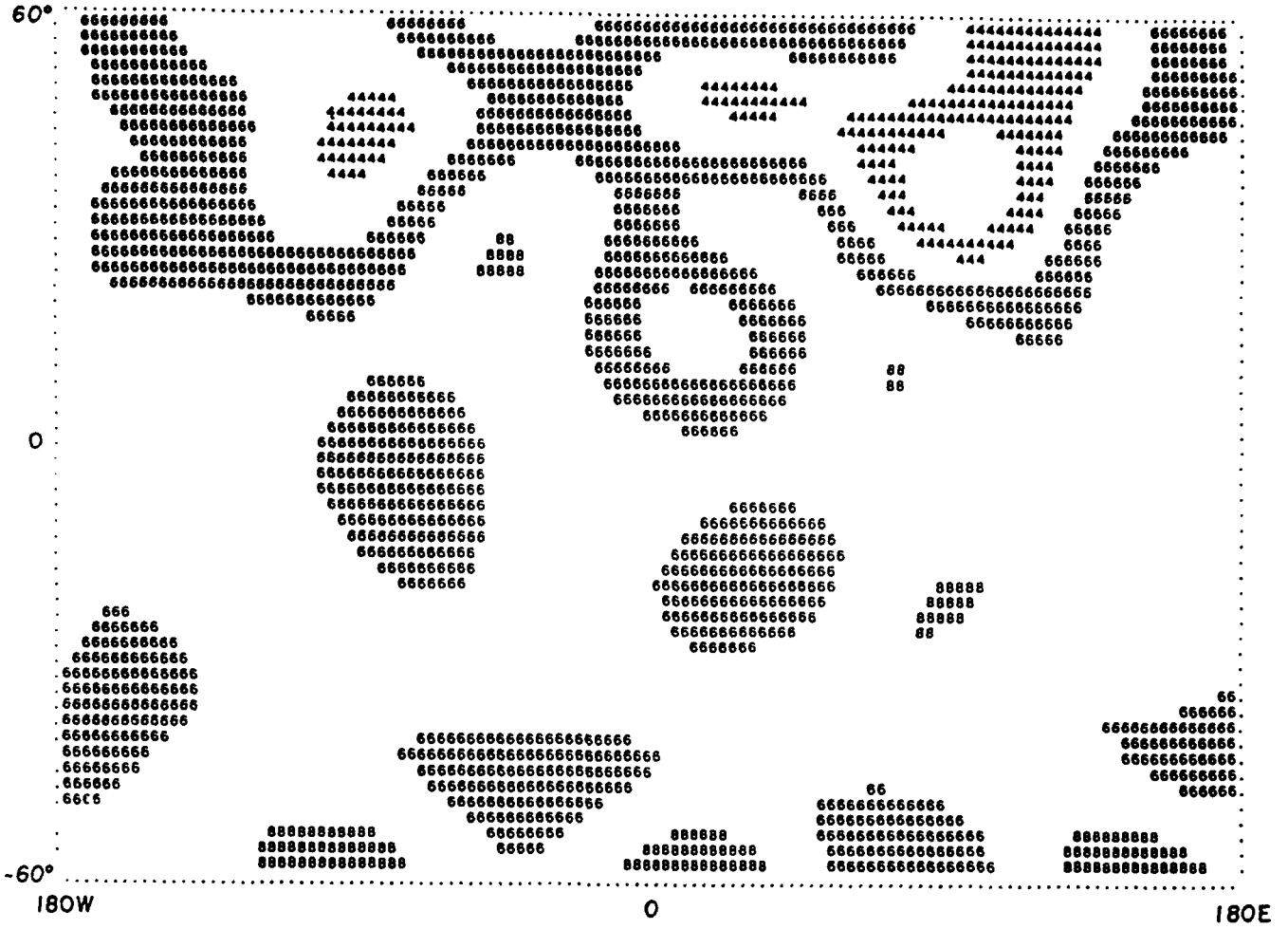



Fig. 2. Example of A6060 Analysis of Scale Distance. Numbers Indicate Calculated Value of Z. Map Is Equal Area; i.e., Vertical Coordinate Is Sine of Latitude.

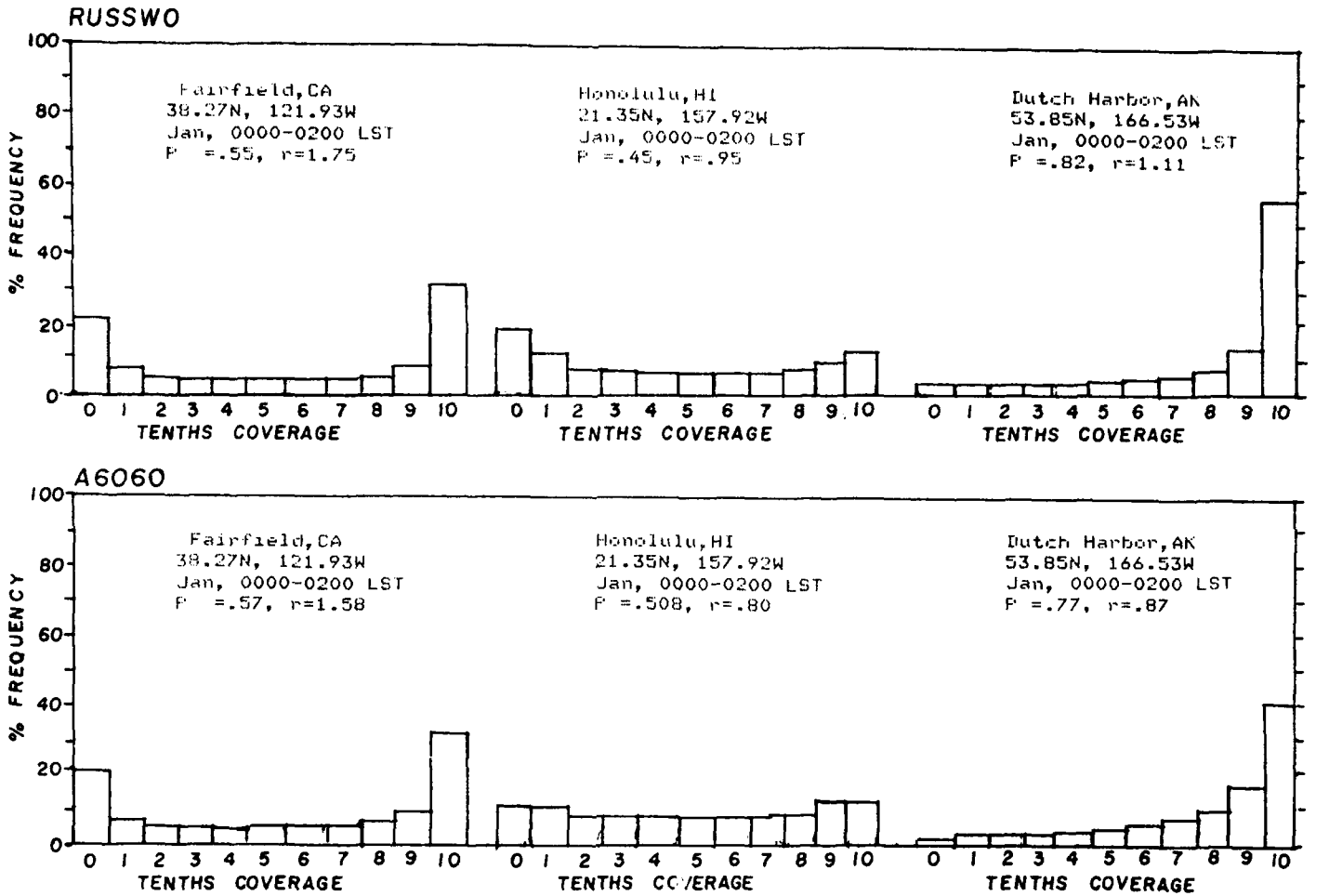



Fig. 3. Comparison of Distributions Calculated by A6060 Analysis Scheme and Distributions from Observations (RUSSWO - Revised Uniform Summary of Surface Weather Observations).

## 7. References


Bertoni, E. A., and I. A. Lund, 1964: Winter Space Correlations of Pressure, Temperature and Density to 16 km. AFCRL-64-1020. Air Force Cambridge Research Laboratories, Hanscom, Envir. Res. Papers No. 75, AD611002.

Burger, C. F., 1985: World Atlas of Total Sky Cover. AFGL-TR-85-0198, Air Force Geophysics Laboratory, Hanscom AFB, ADA170474.

_____ and I. I. Gringorten, 1983: Lineal and areal probabilities of weather conditions. Preprints, Eighth Conference on Probability and Statistics in Atmospheric Sciences, Hot Springs, AK; Amer. Meteor. Soc., Boston, 39-44.

Colombo, O., 1981: Numerical Methods for Harmonic Analysis on the Sphere. AFGL-TR-81-0038, Ohio State University, ADA104178.

Gringorten, I. I., 1979: Probability models of weather conditions occupying a line or an area. J. Appl. Meteor., 18, 957-977.

_____, 1982: Climatic Probabilities of the Vertical Distribution of Cloud Cover. AFGL-TR-87-0078, Air Force Geophysics Laboratory, Hanscom AFB, ADA 118753.

Westlake, J., 1968: A Handbook of Numerical Matrix Inversion and Solution of Linear Equations. John Wiley & Sons, Inc., New York.

Willand, J. H., 1988: Users Manual for C Cloud S Database. AFGL-TR-88-0060, Contract F19628-87-C-0046, ST Systems Corporation.

## V. ATMOSPHERIC TRANSPORT AND DIFFUSION

### Atmospheric Transport and Diffusion

AFGL has been working for several years to perfect a mathematical model that predicts surface layer wind-adjusted toxic hazard corridors resulting from chemical spills near Air Force installations. WADOCT (wind and diffusion over complex terrain) is a model that combines major aspects of the AFGL toxic chemical diffusion model, AFTOX, with AFWIND, the AFGL surface layer windflow model.

During most of 1987 and early 1988, STX concentrated efforts on upgrading Zenith-100 versions of AFTOX and AFWIND into user-friendly field operational streamlined software for the Zenith-248 microcomputer. This accomplished, the basic functions of each were blended into WADOCT software. Appropriate color graphic representations of results were developed for both computer screen and plotter.

An enhanced method of calculating the toxic chemical plume meandering derived in AFTOX was incorporated into the Zenith-248 version. As part of the AFTOX validation testing, STX wrote software to display any x-y data on a Z-248 monitor and to provide a line-of-best-fit and correlation coefficient of these points. Up to 500 x-y log or linear values can be shown. Similar displays can be produced on the Graphtec plotter. These methods were employed to produce scatter plots of measured atmospheric concentrations of released chemicals vs. concentrations predicted by AFTOX. Hazard distances observed and predicted were similarly compared. Known Air Force databases including Ocean Breeze, Dry Gulch, Prairie Grass, and Green Glow served as controls for the validation.

Field usage and test data runs made during the year prompted revision to make both the windflow and diffusion models as accurate and responsive to field needs as possible. WADOCT, the combined model, benefitting from the intense scrutiny and testing of both AFWIND and AFTOX, combines the essential features of each. The spill scenarios covered by WADOCT do not produce the detailed output of AFTOX but coupled with the windflow provide unique combinations. WADOCT includes the option to run only the windflow or only the diffusion model routines if desired.

To run WADOCT an input file of terrain heights must be prepared describing a domain of up to 10 x 10 km with grid spacing up to 500 m. Further enhancement of these values by the addition of roughness or vegetation is needed. Additionally required is a data file containing the chemical properties of some 56 substances including their molecular weights, exposure limits, etc. Other chemicals may be introduced if necessary information is supplied at run time.

An initialized wind field is hypothesized based on wind information measured at one or more sites. If multiple observational inputs are used, an objective analysis scheme to derive non-homogeneous conditions for initialization is employed. The windflow model portions of WADOCT perform a variational analysis of surface layer winds in the x-y plane to induce an initial wind field to conform to the constraints of topography, buoyancy forces, advection of momentum, and conservation of mass. The driving equations for this system attempt to minimize a volume integral relating momentum advection to buoyancy forces. When the minimum value is attained, the system is said to be in quasi-steady balance between constraints (Lanicci and Weber, 1986).

A chemical spill occurring at some x-y point within the domain is introduced. Spill size, rate, height above ground, liquid or gas, and instantaneous or continuous release are inputs requested by the model. Also input are the elapsed time since the start of the spill, the concentration averaging time, and whether an atmospheric inversion exists.

Based on a Gaussian puff scheme, maximum concentration areas and downwind distance limits are predicted and up to five contours representing differing concentrations are calculated. Points within the domain outlining the toxic hazard plumes are stored.

As described above, the windflow is "relaxed" over several steps until minimum is reached. Following solution of the windflow, the chemical plume outline is adjusted to conform to the changes made. Output files are generated to produce displays of the calculated windflow and toxic plume.

STX produced software to display the wind vectors and plume in contrasting colors for the Zenith-248 microcomputer screen or for output to a color plotter. Allied software produced a terrain contour file that can be summoned to underlie the wind vectors and chemical plume on either the screen or plotter (Fig. 1). The entire domain or any selected window can be shown.

The Raj heavy gas model was not delivered to AFGL until late December, at which time it was decided not to combine it with the dispersion model.

A replacement system for archiving data from weather sensors under the control of a Zenith-248 and backup tape unit at the AFGL Weather Test Facility (Otis ANGB) is still being installed and tested; thus no data analysis by STX was

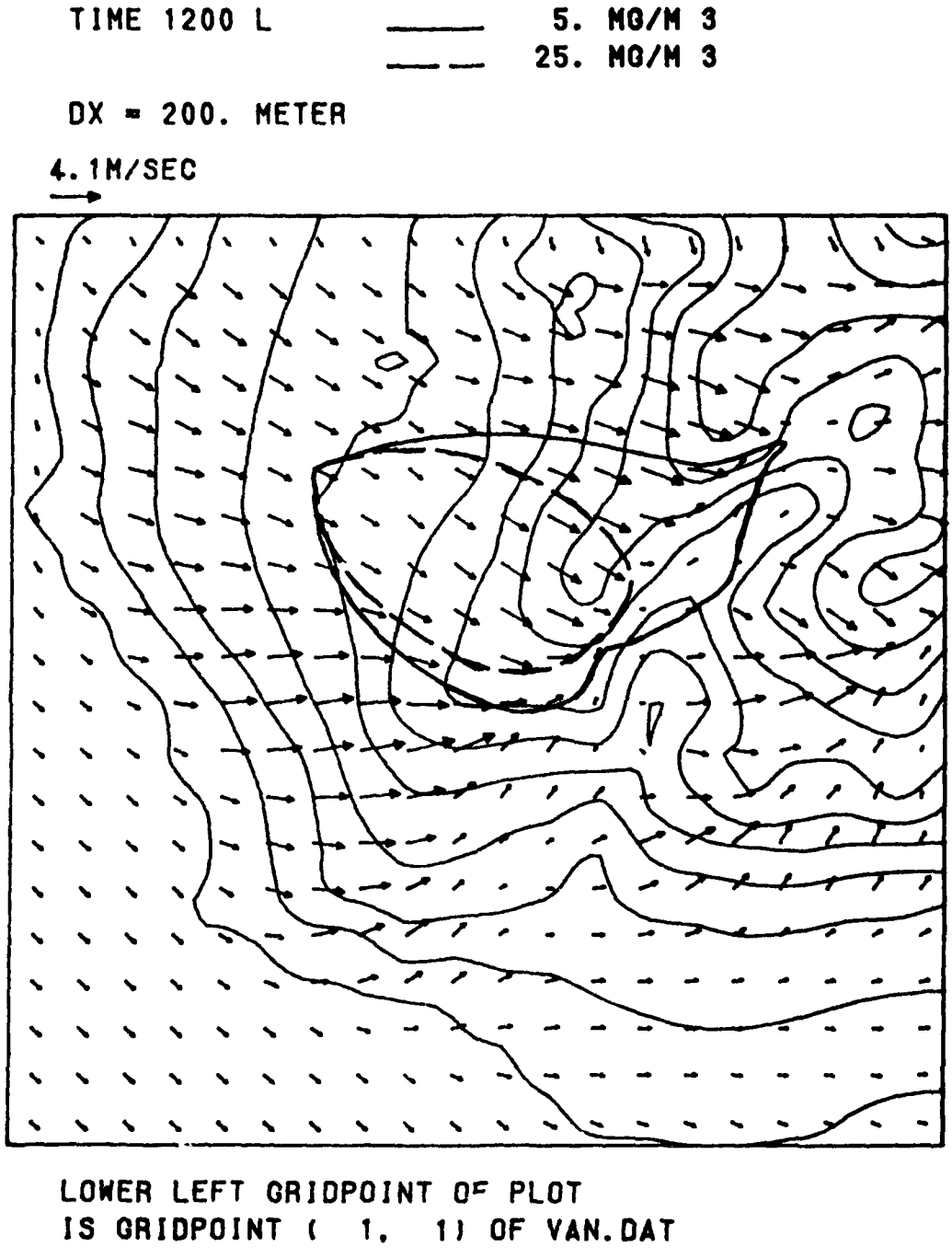



Fig. 1. Sample WADOCT and Contour Program Output Showing Predicted Plumes of Two Concentrations Resulting from Spill of Unknown Chemical at Gridpoint (7, 15) of South Vandenberg AFB Data Array.

possible. The MAWS system continues as the basis for sensor data archiving at Otis. STX maintained the processing software and made occasional revisions and upgrades during the year as required.

Reference

Lanicci, J. M., and H. Weber, 1986: Evaluation of a surface-layer windflow model for complex terrain using meteorological tower data from Vandenberg AFB, CA. Preprints, Fifth Joint Conference on Applications of Air Pollution Meteorology, Chapel Hill, NC; Amer. Meteor. Soc., Boston, 265-268.