Memory Efficient Solution of the Primitive Equations
for Numerical Weather Prediction on the CYBER 205

James J. Tuccillo

Systems and Applied Sciences Corporation

5809 Annapolis Road

Hyattsville, MD 20784

1. INTRODUCTION

Numerical Weather Prediction (NWP), for both operational and research purposes, requires not only fast computational speed but also large memory. In this paper I will discuss a technique for solving the Primitive Equations for atmospheric motion on the CYBER 205, as implemented in the Mesoscale Atmopsheric Simulation System (MASS) (Kaplan et. al., 1982), which is fully vectorized and requires substantially less memory than other techniques such as the Leapfrog or Adams-Bashforth Schemes. The technique to be presented uses the Euler-Backard time marching scheme.

Also to be discussed will be several techniques for reducing the CPU time of the model by replacing "slow" intrinsic routines by faster algorithms which use only hardware vector instructions.

2. MODEL BACKGROUND

2.1 Description

MASS is a hydrostatic primative equation model which is run over a limited area. The model forecast the 3-dimentional structure of wind, pressure, temperature and moisture. The actual domain of coverage, along with the horizontal distribution of grid points, is depicted in Fig. 1. The characteristics of the model are listed in Table 1.

2.2 Uses and Support

The model has been applied primarily to the problem of forecasting the atmospheric environment within which severe local storms (severe thunderstorms and tornadoes) are likely to develop. It has also been applied to the problems of forecasting and investigating east coast cyclogenesis, upper level turbulence and shear, and boundary layer transport. Support for the model development has been provided by NASA/Goddard using the computational facilities of NASA/Langley (CYBER 203) and NASA/Goddard (CYBER 205)

2.3 History

The original version was implemented on a 500K word CDC STAR 100 Vector Processor at NASA/Langley in the late 70's using 64-bit FORTRAN. The availability of the SL/1 programming language at Langley, which permitted easy access to the 32-bit instruction set on the STAR 100, resulted in an effective doubling of the memory and the model was recoded with larger vectors. This allowed for an increase in the area over which the model was run while maintaining the same horizontal and vertical resolution.
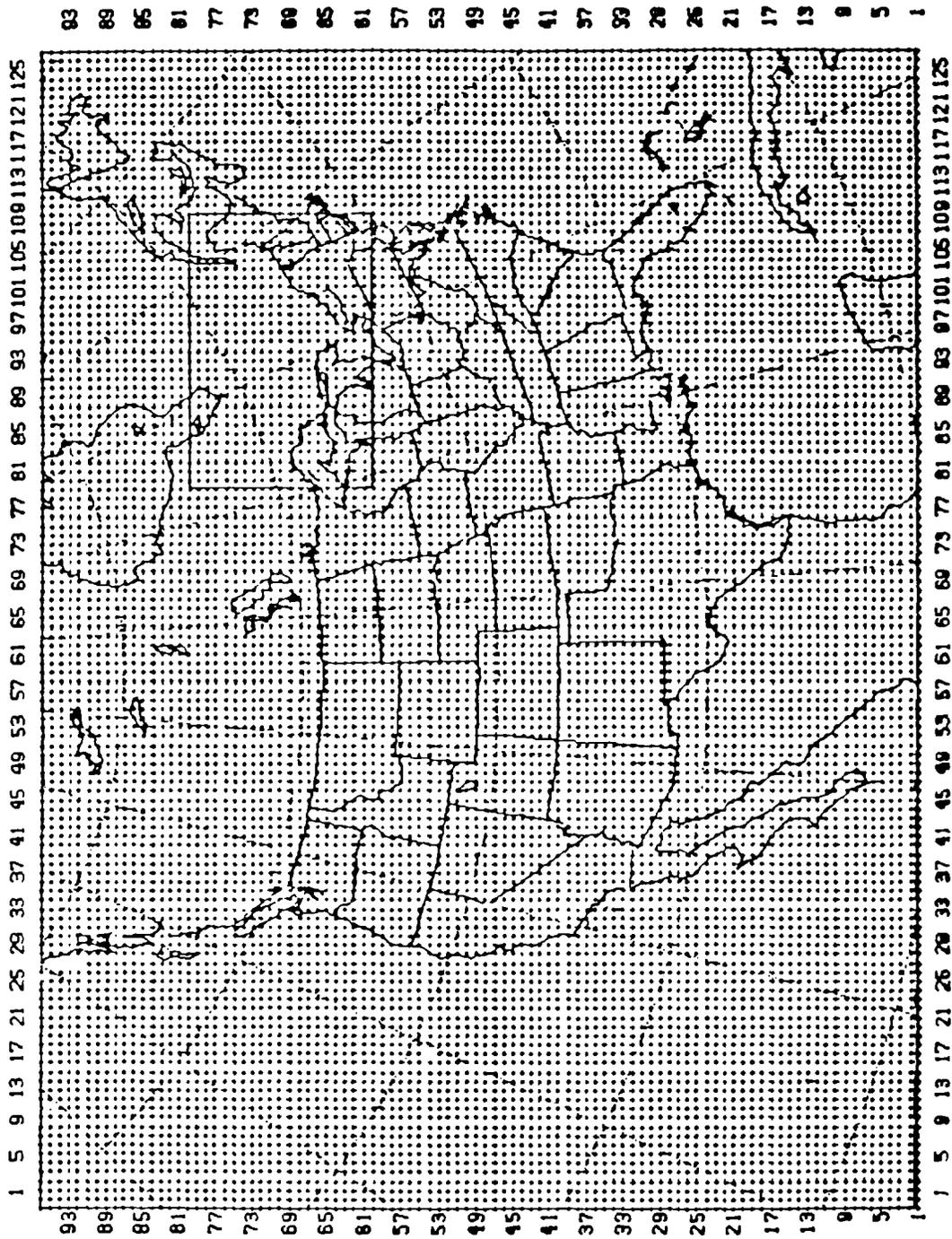
FIG. 1 DOMAIN OF COVERAGE BY MASS MODEL AND THE HORIZONTAL
OF GRID POINTS DISTRIBUTION

37

TABLE 1 CHARACTERISTICS OF MASS MODEL

MASS (DESCRIPTION)

o   HYDROSTATIC PRIMITIVE EQUATIONS

o   TERRAIN FOLLOWING SIGMA-P COORDINATE

o   LIMITED AREA DOMAIN

o   CARTESIAN GRID ON A POLAR STEREOGRAPHIC MAP (ARAKAWA "A" GRID)

o   4TH ORDER ACCURATE HORIZONTAL SPACE DIFFERENCING

o   2ND ORDER ACCURATE VERTICAL SPACE DIFFERENCING

o   2ND ORDER ACCURATE TIME DIFFERENCING

o   INITIAL DATA IS DERIVED FROM THE LFM ANALYSIS PLUS RAWINSONDES

o   INITIALIZATION IS BASED ON THE CALCULUS OF VARIATIONS

o   PHYSICS

        – LARGE SCALE PRECIPITATION

        – PLANETARY BOUNDARY LAYER

        – DRY CONVECTION

        – MOIST CONVECTION (UNDER DEVELOPMENT)

o   50 KM GRID SPACING AT 45°N

o   19 EQUALLY SPACED LAYERS

o   128 X 96 COMPUTATIONAL DOMAIN

o   TIME DEPENDENT BOUNDARY CONDITIONS

o   COMPREHENSIVE INTERACTIVE DIAGNOSTIC PACKAGE ON THE FRONT END

        – VERTICAL PROFILES

        – VERTICAL CROSSECTIONS

        – CONSTANT PRESSURE SURFACES

        – TIME HISTORY

        – TRAJECTORIES

        – VERIFICATION STATISTICS

In the spring of 1980, the STAR 100 was upgraded to a 1m word CDC CYBER 203. The new machine effectively had twice the memory of the STAR 100. The area over which the model is run was again expanded and the vertical resolution was increased from 12 to 14 vertical layers.

In the spring of 1983, the model was transferred to the NASA/Goddard CYBER 205. The model was recoded in CDC FORTRAN 2.0 using 32-bit arithmetic. After being successfully benchmarked against the Langley version, the vertical resolution was again increased from 14 to 19 layers. The Goddard version of MASS on the CYBER 205 executes approximately 3 times faster than the Langley version on the CYBER 203. This can be explained by

1) Reduction in cycle time from 40 to 20 NS.
2) Linked triad instruction on the CYBER 205.
3) Faster gather/scatter instruction.
4) Coding differences.

3. EQUATION SET

The model utilizes a standard primitive equation set cast in a terrain following σ_p coordinate system. As indicated earlier, the forecasted variables are the 3-D distribution of wind, pressure, temperature and moisture. The basic prognostic equations are given below where u and v are x and y coordinate momentum, T is temperature, q is the moisture mixing ratio and π is the pressure at the terrain minus the pressure at the top of the model.

39

$$\frac{\partial u}{\partial t} = -um\frac{\partial u}{\partial x} - vm\frac{\partial u}{\partial y} - \dot{\sigma}\frac{\partial u}{\partial \sigma} - m\frac{\partial \phi}{\partial x} - \sigma\alpha m\frac{\partial \pi}{\partial x} + fv + X\nabla^4 u + \frac{\partial u}{\partial t}\Big|_{Physics}$$

$$\frac{\partial v}{\partial t} = -um\frac{\partial v}{\partial x} - vm\frac{\partial v}{\partial y} - \dot{\sigma}\frac{\partial v}{\partial \sigma} - m\frac{\partial \phi}{\partial y} - \sigma\alpha m\frac{\partial \pi}{\partial y} - fu + X\nabla^4 v + \frac{\partial v}{\partial t}\Big|_{Physics}$$

$$\frac{\partial T}{\partial t} = -um\frac{\partial T}{\partial x} - vm\frac{\partial T}{\partial y} - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \frac{\alpha}{c_p}\omega + X\nabla^4 T + \frac{\partial T}{\partial t}\Big|_{Physics}$$

$$\frac{\partial \mathfrak{z}}{\partial t} = -um\frac{\partial \mathfrak{z}}{\partial x} - vm\frac{\partial \mathfrak{z}}{\partial y} - \dot{\sigma}\frac{\partial \mathfrak{z}}{\partial \sigma} + X\nabla^4 \mathfrak{z} + \frac{\partial \mathfrak{z}}{\partial t}\Big|_{Physics}$$

$$\frac{\partial \pi}{\partial t} = \int_1^0 m\frac{\partial u\pi}{\partial x} + m\frac{\partial v\pi}{\partial y}\, d\sigma$$

NOTE: α = specific volume
f = coriolis parameter

Three diagnostic equations close the system and are given below where $\dot{\sigma}$ is the

vertical velocity, ϕ is the geopotential energy and ω is the vertical velocity in pressure

coordinates.

$$\dot{\sigma}_{k+\frac{1}{2}} = \dot{\sigma}_{k-\frac{1}{2}} + \left[\frac{\partial \pi}{\partial t} + m\frac{\partial u\pi}{\partial x}\Big|_k + m\frac{\partial v\pi}{\partial y}\Big|_k\right]\frac{\Delta\sigma}{\pi}$$

$$\phi_{k+\frac{1}{2}} = \phi_{k-\frac{1}{2}} + RT_k\left(1 + 0.61\mathfrak{z}_k\right)\ln\left(P_{k-\frac{1}{2}}/P_{k+\frac{1}{2}}\right)$$

$$\omega_k = \frac{\pi}{2}\left[\dot{\sigma}_{k+\frac{1}{2}} + \dot{\sigma}_{k-\frac{1}{2}}\right] + \sigma_k\left[\frac{\partial \pi}{\partial t} + um\frac{\partial \pi}{\partial x}\Big|_k + vm\frac{\partial \pi}{\partial y}\Big|_k\right]$$

The boundary conditions are

$$\dot{\sigma}_1 = \dot{\sigma}_0 = 0$$

$$\phi_{\frac{1}{2}} = \text{TERRAIN HEIGHT}$$

and the definitions for and π are

$$\sigma = \frac{P - P_{top}}{\pi} \qquad \pi = P_{sur} - P_{top}$$

the remaining variables are

40

$m =$ mapscale grid transformation factor

$c_p =$ specific heat at constant pressure

$R =$ gas constant for dry air

$P_{sur} =$ pressure at the terrain

$P_{top} =$ pressure at the top of the model

$\chi =$ horizontal eddy diffusivity

4. GRID SYSTEM

The technique for solving the differential equations is to discretize the equations into finite difference form and solve them on a 3-D grid. The horizontal grid employed is the Arakawa "A" grid where all dependent variables are defined at all grid points. The vertical grid is staggered so that u, v, T and q represent layer averages defined at the mid-point of each layer and and are held at the layer interfaces. The third diagnostic variable, w, is held with u, v, T and q. This structure is represented in Fig. 2.

5. NUMERICAL TECHNIQUE

5.1 Horizontal Space Derivatives

The fourth order accurate finite difference approximation to an x-direction space derivative for an arbitrary variable ψ is given below

$$\left.\frac{\partial \psi}{\partial x}\right|_i \approx \frac{1}{12\Delta x}\left[8\left(\psi_{i+1} - \psi_{i-1}\right) - \left(\psi_{i+2} - \psi_{i-2}\right)\right] + O\left(\Delta x^4\right)$$

where i is a horizontal index. An analogous formula is used for y – direction derivatives.

41

$$\bullet$$
$$\bullet$$
$$\bullet$$

— — — U, V, T, q, ω — — — $K + 1$

———————— $\dot{\sigma}, \phi$ ———————— $K + \frac{1}{2}$

— — — U, V, T, q, ω — — — K

———————— $\dot{\sigma}, \phi$ ———————— $K - \frac{1}{2}$

— — — U, V, T, q, ω — — — $K - 1$

$$\bullet$$
$$\bullet$$
$$\bullet$$
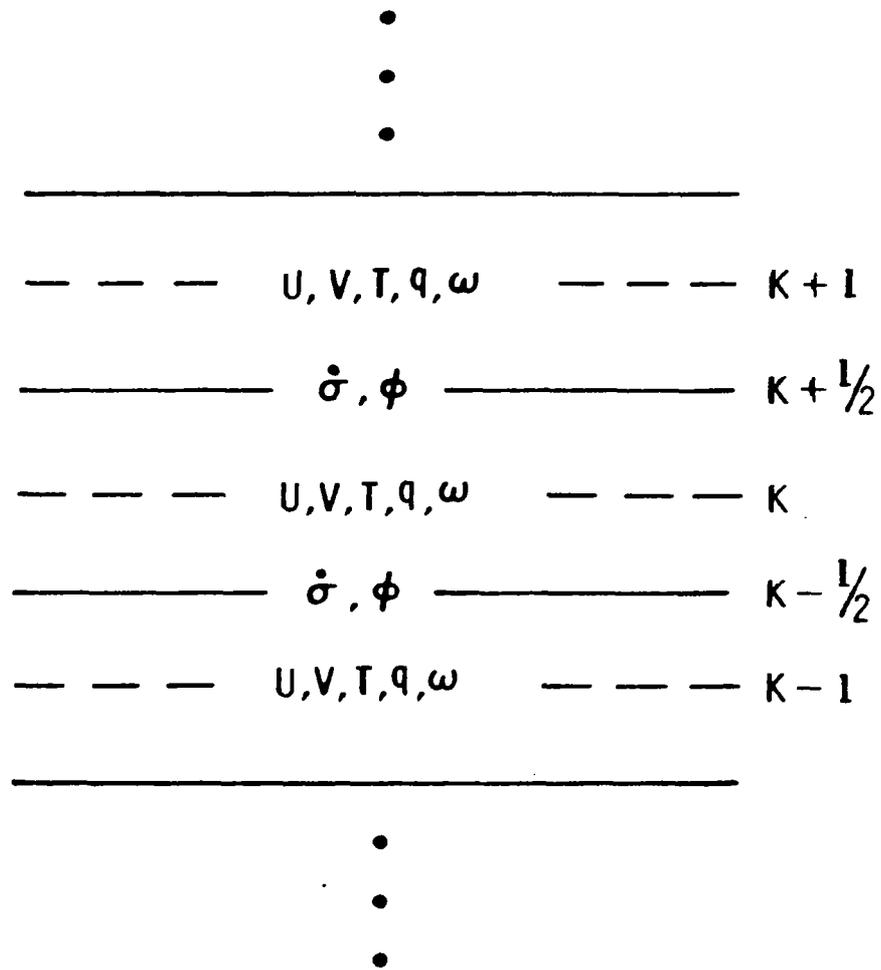
Fig. 2 Vertical grid system of MASS

5.2. Vertical Space Derivatives

A second order accurate finite difference formula is used to approximate the vertical advection terms of the u,v, T and q prognostic equations. The representation, for an arbitrary variable ψ, is given below

$$\dot{\sigma}\frac{\partial \psi}{\partial \sigma}\bigg|_k \approx \frac{1}{2\Delta\sigma}\left[\dot{\sigma}_{k+\frac{1}{2}}\left(\psi_{k+1}-\psi_k\right)+\dot{\sigma}_{k-\frac{1}{2}}\left(\psi_k-\psi_{k-1}\right)\right]$$

where k is a vertical index.

5.3 Time Derivatives

A second order accurate approximation to the time derivatives is used. The Euler-Backward Technique has the properties of frequency dependent damping and no computational mode. For an arbitrary variable ψ the finite difference representation is given as

$$\psi^* = \psi^n + \frac{\partial \psi^n}{\partial t}\Delta t \qquad \text{Prediction}$$

$$\psi^{n+1} = \psi^n + \frac{\partial \psi^*}{\partial t}\Delta t \qquad \text{Correction}$$

where n is a time level index and * refers to a intermediate time level.

This scheme requires the storage of only one time level of information (time level n) whereas other explicit schemes such as the Leapfrog Scheme requires the storage of at least two time levels (n and n-1). The penalty is that twice the computational work is required as compared with the Leapfrog scheme.

6.   BASIC MEMORY REQUIREMENTS

As mentioned earlier, the Euler-Backward scheme for time marching the prognostic equations for the 3-D structure of wind, pressure, temperature and moisture requires the storage of only one time level of information. The * 'ed time level is an intermediate time level and only needs to be as deep (with respect to the vertical) as is required to solve the equations at a layer. It should be noted that only the vertical advection terms couple the model layers together and that to solve the equations at layer k requires the dependent variables at layers k+1, k and k-1. Therefore, the * 'ed time level only needs to be 3 deep (it holds the prediction values to be used during the correction step) and can be reused for the solution of each layer.

Given that the 19 model layers contain 128 x 96 grid points each, the basic memory required is

u (128, 96, 19)

v (128, 96, 19)

T (128, 96, 19)

q (128, 96, 19)

pi (128, 96)

ustar (128, 96, 3)

vstar (128, 96, 3)

tstar (128, 96, 3)

qstar (128, 96, 3)

pistar (128, 96)

If an additional layer were to be added only the u, v, T and q arrays would be increased. The ustar, vstar, tstar and qstar arrays are always dimensioned 3 deep and this is a function of the vertical advection terms which require 3 layers of storage to solve the equations.

In contrast, the Leapfrog scheme would require 2 sets of arrays dimensioned 128 x 96 x 19, therefore, there is a considerable memory savings with the Euler-Backward Scheme. A technique developed by Tuccillo (1983) shows some promise in reducing the computational work by increasing the premissable timestep.

7.   METHOD OF SOLUTION

The method of solution is depicted in Fig. 3 and shows the sequence of steps required to solve the equations at all layers. Prediction is the step that advances the solution from the n to the * time level and correction is the step that advances the solution from the * to the n+1 time level. It there are NZ layers then there are 2*NZ number of steps required to advance the solution one time step. The number above each line represents the order of solution where the first step is to perform prediction for layer 1, the second step is prediction at layer 2, the third step is correction at layer 1 and so on. After correction (the 2*NZ step) at layer NZ is finished the solution has been advanced one time step.

LAYER

NZ $\underline{\quad 2*NZ-2 \quad}$ $\underline{\quad 2*NZ \quad}$

NZ-1 $\underline{\quad 2*NZ-4 \quad}$ $\underline{\quad 2*NZ-1 \quad}$

$\underline{\quad 2*NZ-3 \quad}$

·   ·
·   ·
·   ·

3         $\underline{\qquad 4 \qquad}$

2         $\underline{\qquad 2 \qquad}$ $\underline{\qquad 5 \qquad}$

1         $\underline{\qquad 1 \qquad}$ $\underline{\qquad 3 \qquad}$

PREDICTION   CORRECTION

$$\psi^n \rightarrow \psi^*$$ $$\psi^n \rightarrow \psi^{n+1}$$
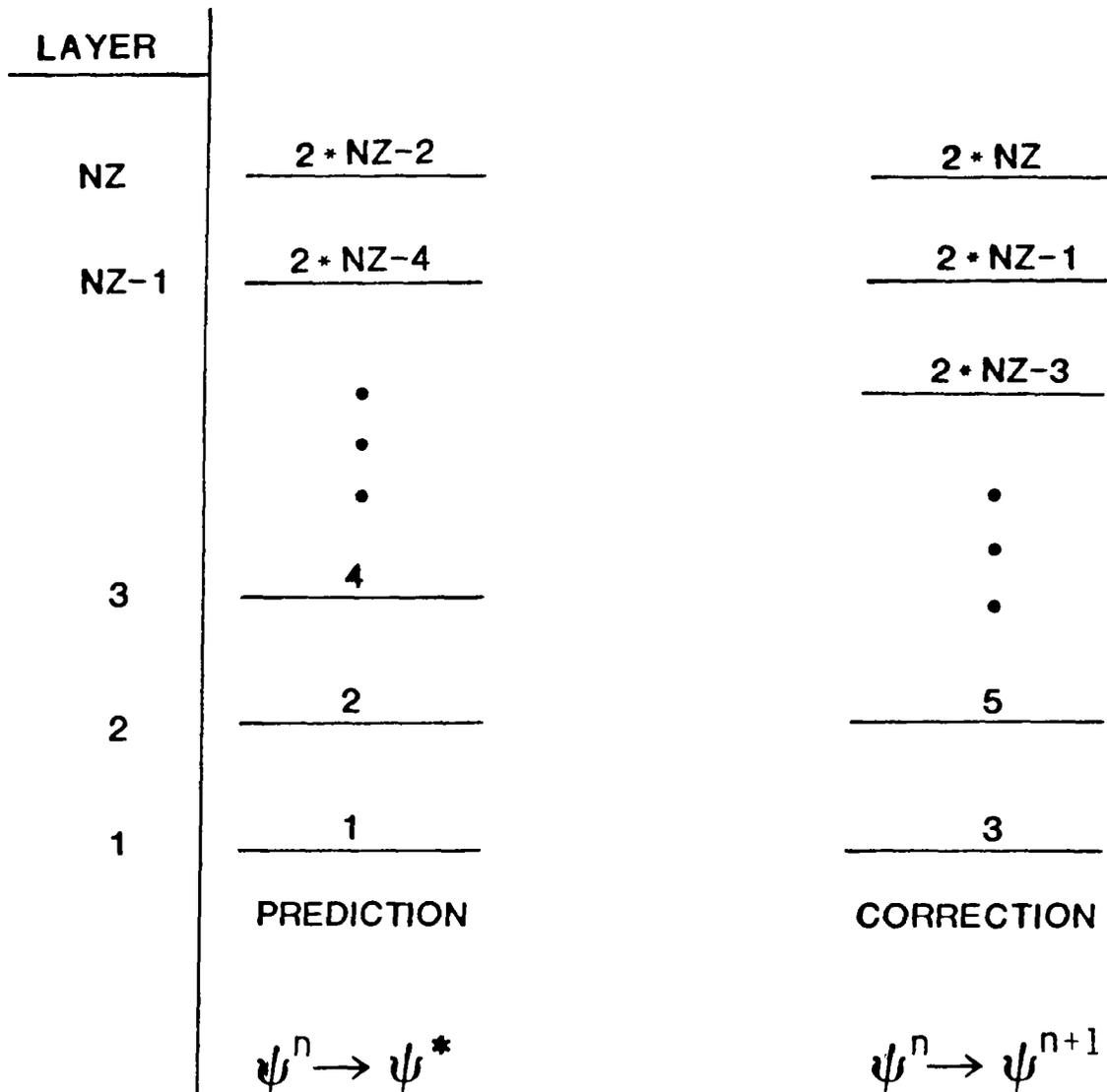
FIG. 3 SEQUENCE OF STEPS TO ADVANCE THE SOLUTION
ONE TIMESTEP

The *'ed arrays are reused for each layer and the calculations for each layer are fully vectorized where the vector lengths are NX*NY or 12288. For this vector length the machine is computing at about 98% of its maximum rate.

8. BOUNDARY CONDITIONS

Since MASS is a limited area model, as opposed to a global model, the solution at the horizontal boundaries needs to be specified. The technique for specifying the boundary conditions consist of blending externally calculated values using a weighted average formula which is represented by

$$\frac{\partial \psi}{\partial t} = W \frac{\partial \psi}{\partial t}\bigg|_{INTERIOR} + (1-W)\frac{\partial \psi}{\partial t}\bigg|_{EXTERIOR}$$

where   W = 0 on outer column and row

W = 0.333 on first column and row in

W = 0.666 on second column and row in

W = 1.0 on third column and row in

It should be pointed out that this technique produces an overspecification at the boundary and higher horizontal diffusion is required near the boundaries to control noise generation.

This technique is vectorized by holding the externally specified boundary tendencies in a vector and using the scatter instruction to expand them into the correct positions prior to computing the weighted average. This technique minimizes to amount of storage required.

9. PROGRAMMING TECHNIQUES

The code is completely vectorized in the horizontal. The average vector length is about 12000 which represents the number of horizontal grid points. There is a loop over the vertical layers.

Some specific techniques used during the coding are

o   32-bit arithmetic

Sensitivity tests have indicated that 32-bits provides enough precision. Using 32-bits effectively doubles the real memory and halves the execution time.

o   Explicitly Vectorized

The code does not depend on automatic vectorization by the compiler. All descriptors are set up with DATA and ASSIGN statements. Special Q8 calls are used where required.

· o   Diadic and Triatic Structure

All vector statements are written in a diadic structure (triadic when linked triads are created) to minimize compiler generated dynamic space which may cause paging.

o   Subroutines are kept small enough so that the Register File is not overflowed.

Subroutines which have more local variables then the size of the register file (approximately 200) can be inefficient since loads from memory must be executed. All subroutines are kept small enough so that the swap instruction can load all necessary local variables at entry.

48

o   Parameter Statement Used for Vector Dimensions

        Vector dimensions are easily changed by changing parameter values..

o   Factoring of Equations to yield Linked Triads

        The sequence of instructions have been arranged to yield the maximum number of linked triads.

o   Run Only in Real Memory

        No page faults are generated during the interative time marching.

o   Vectors are Grouped on Large Pages

        All large vectors are placed in common and grouped on large pages using loader options.

o   Bit Vectors vs. Gather/Scatter

        For those situations where control store or gather/scatter can be applied, an analysis using the nominal performance figures for each instruction was performed and the most CPU or memory efficient techniques was applied.

10. TECHNIQUES FOR REDUCING CPU TIME

A 24-hour simulation with the model requires 1312 timesteps. Each timestep requires the evaluation of 2*NZ natural logs (for 12288 grid points). This required approximately 22 mins of CPU time using the 32-bit FORTRAN VHALOG function. Since the range of arguments for the natural log function was known, a more efficient

technique was incorporated where the natural log was approximated with a series factored using Horner's Rule. The evaluation requires 11 vector instructions, nine of which are linked triads, and runs approximately 40 times faster than the FORTRAN intrinsic function. This technique reduced the CPU time spent evaluating natural logs to 30 secs.

Other techniques for reducing CPU time consist of approximating the ** FORTRAN function with series of square roots (square root in a hardware instruction) and inverting scalars to generate vector multiplies instead of vector divides.

The version of MASS implemented on the CYBER 205 at NASA/Goddard requires 13 large pages of memory and 15 minutes of CPU time (same as wall time) for a 24 hour simulation over the area depicted in Fig. 1.

11. EXAMPLE OF OUTPUT

MASS at Goddard features a comprehensive postprocessing system to produce output from the model for interpretation. The post processing system runs interactively and produces hard copies on a GOULD electrostatic plotter. Future versions of the postprocessing system will likely feature interactive color graphics which should greatly improve the usability of the modeling system as a research tool for studying atmospheric processes. Figs. 4-12 are examples of the output from three of the six postprocessing programs currently available.
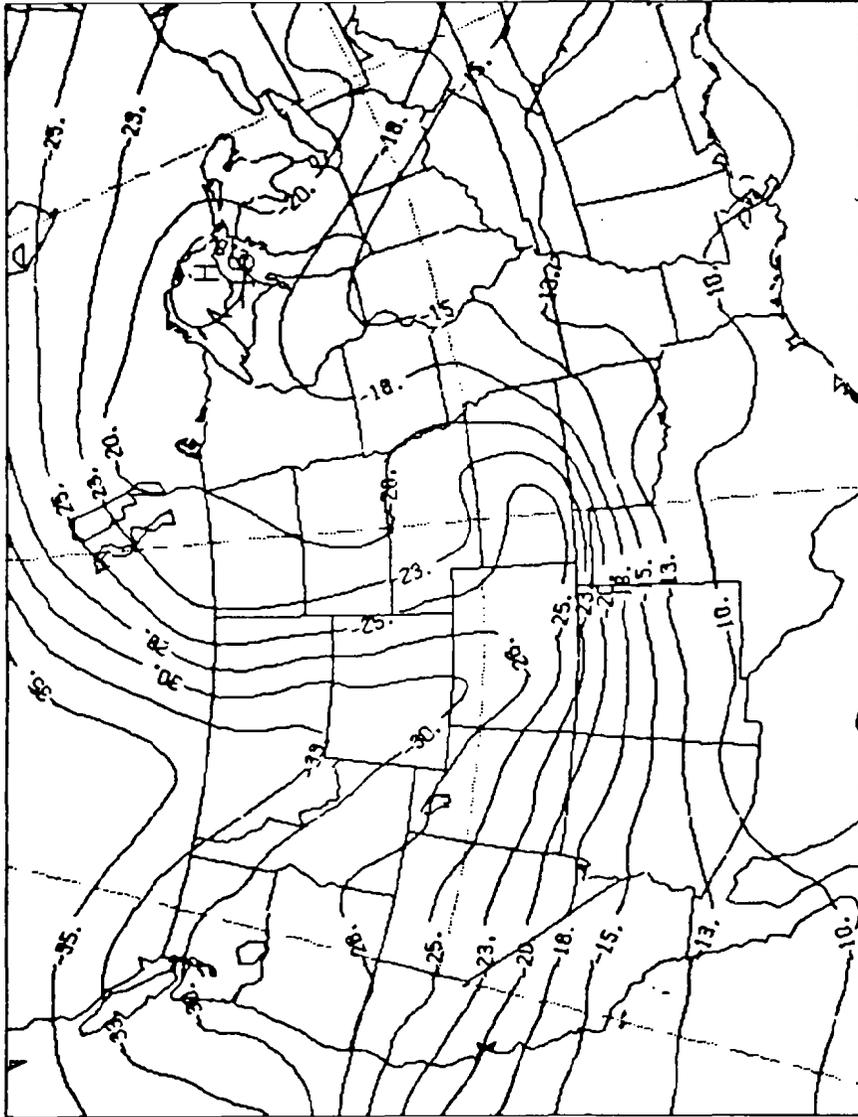
500 T VT 2100 GMT 04/02/82 INIT 1200 GMT 04/02/82

MASS 2.0

FIG. 4 MASS FORECASTS 500 MB TEMPERATURES (DEGREES CELCIUS)
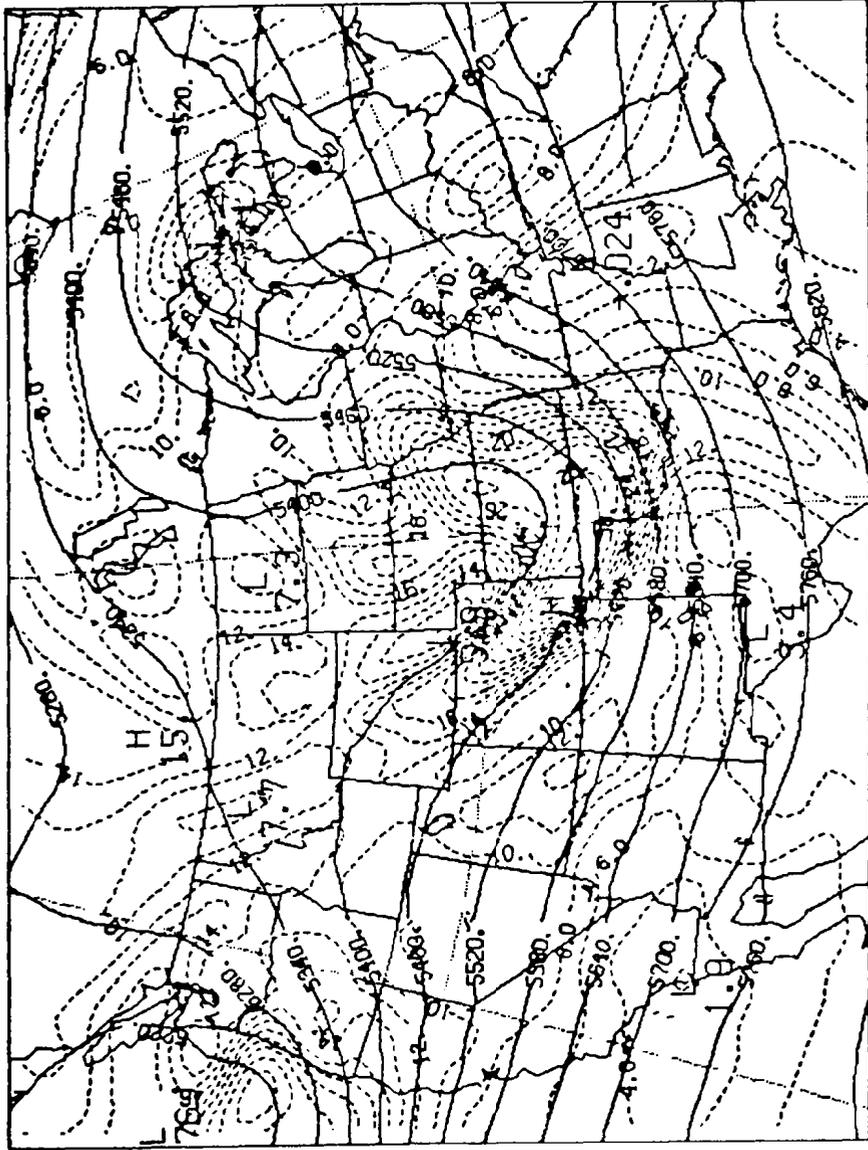
MASS 2.0

500 Z   VT 2100 GMT 04/02/82   INIT 1200 GMT 04/02/82

FIG. 5 MASS FORECASTED 500 MB HEIGHTS (METERS) AND VORTICITY (PER SECOND)
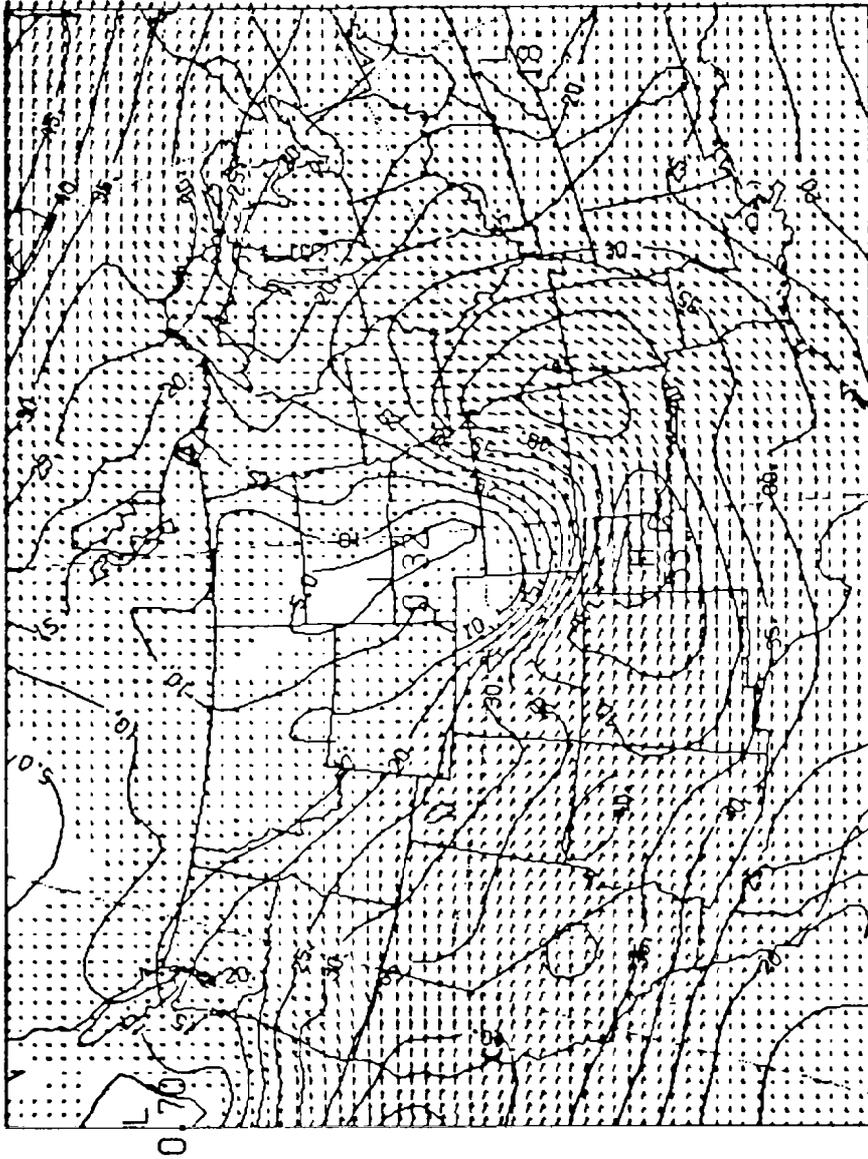
MASS 2.0

500 WND   VT 2100 GMT 04/02/82   INIT 1200 GMT 04/02/82

FIG. 6 MASS FORECASTED 500 MB WIND VECTORS AND ISOTACHS
(METERS PER SECOND)
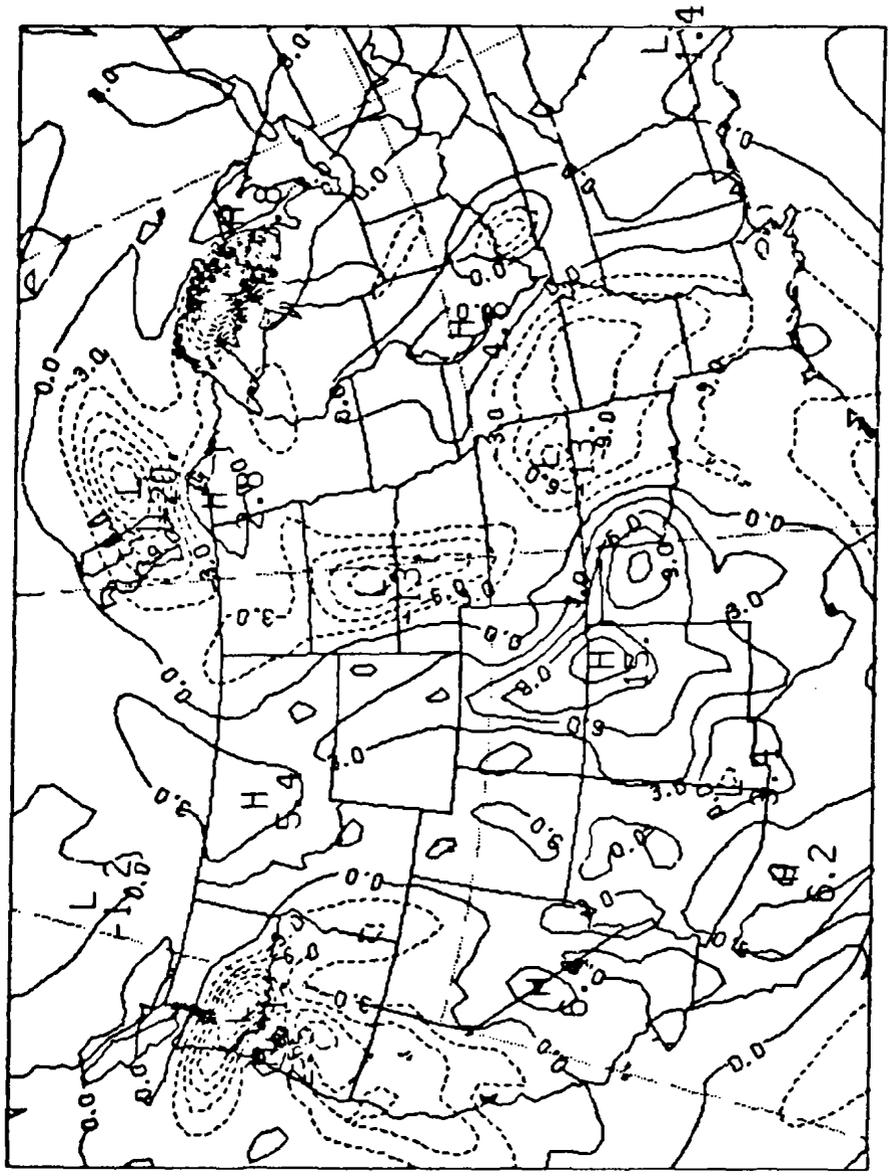
MASS 2.0

500 ØME   VT 2100 GMT 04/02/82 INIT 1200 GMT 09/02/82

FIG. 7 MASS FORECASTED 500 MB VERTICAL VELOCITY
(MICROBARS PER SECOND)
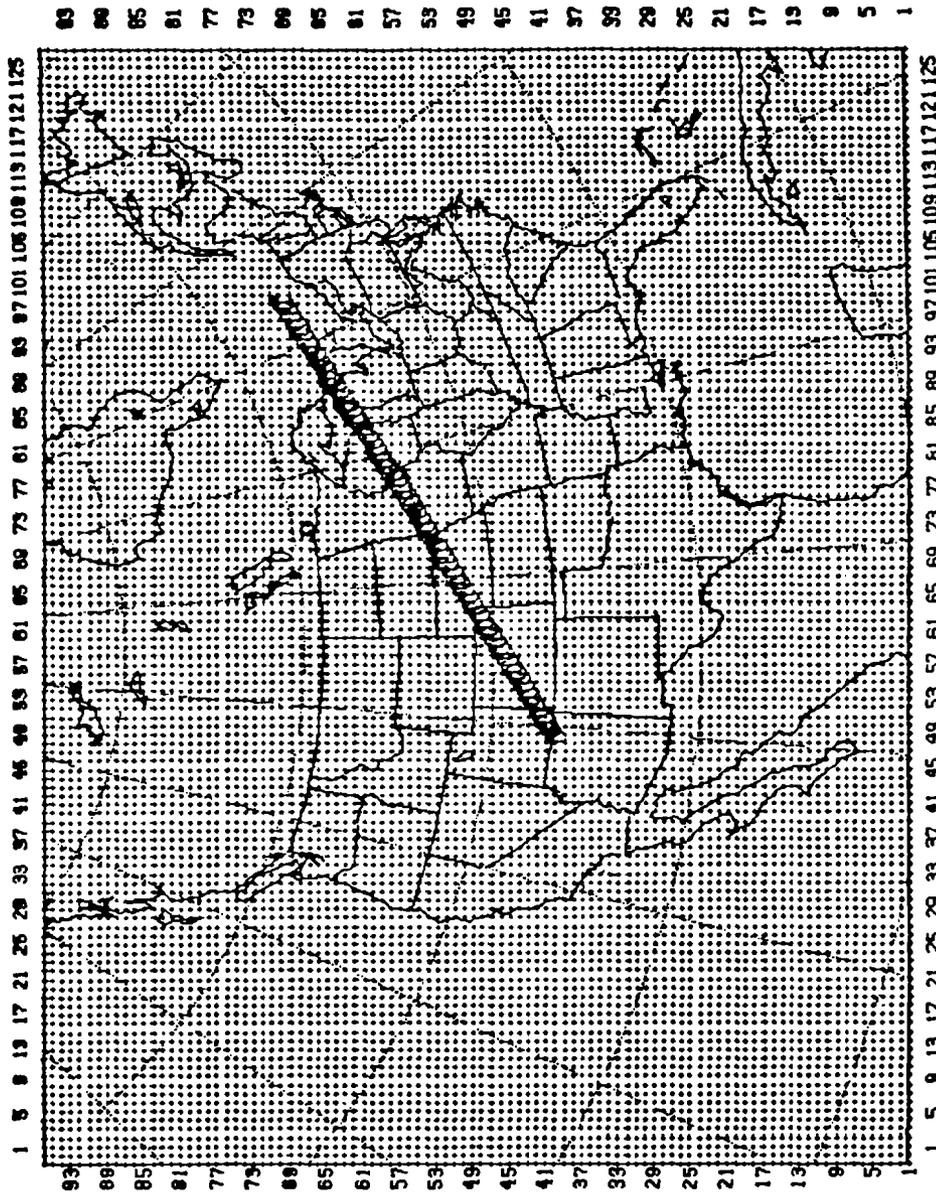
FIG. 8 VERTICAL CROSS-SECTION LOCATOR MAP
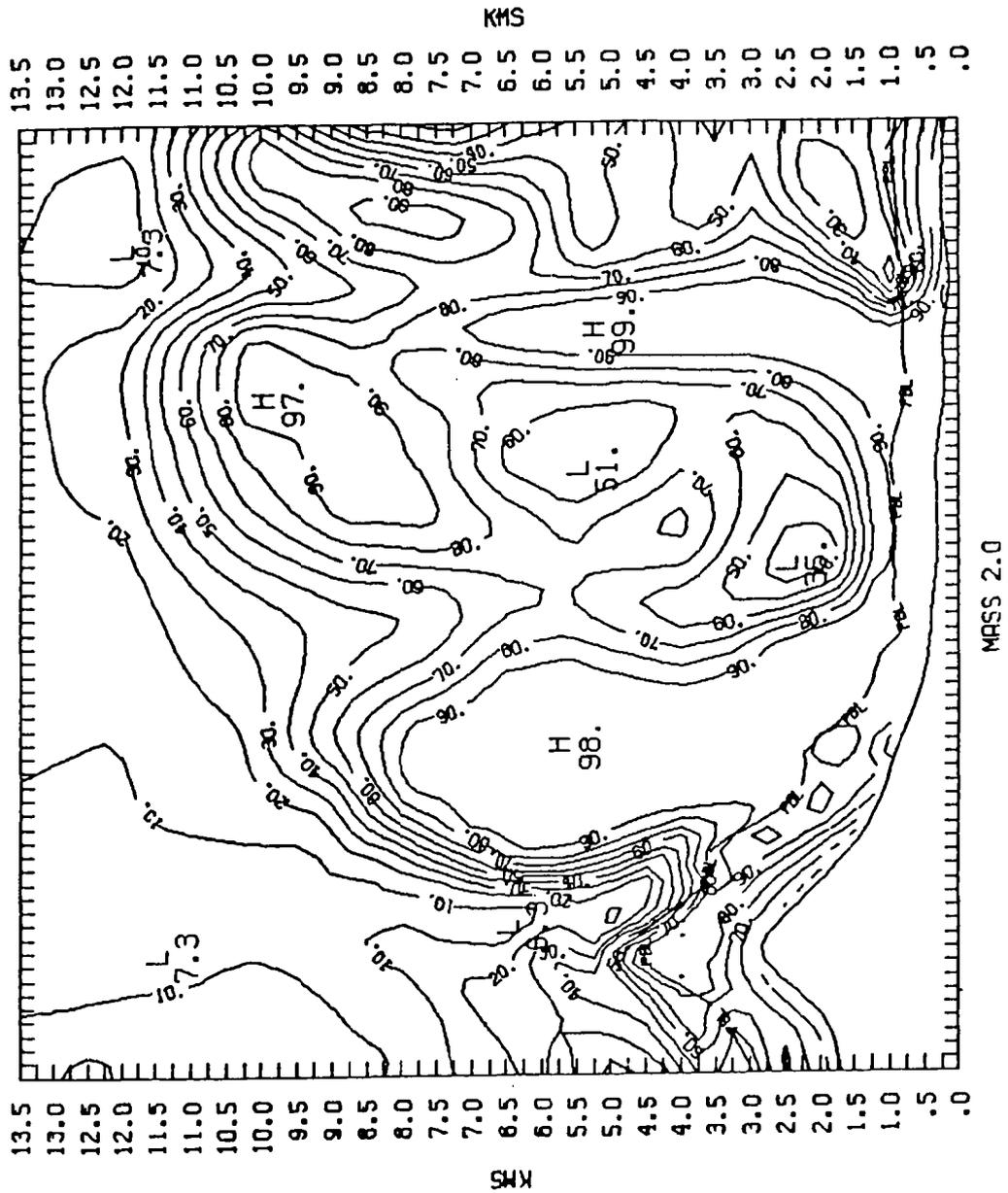
FIG. 9 MASS FORECASTED VERTICAL CROSS-SECTION OF RELATIVE HUMIDITY (PERCENT)

RH   VT 2100 GMT 04/02/82   INIT 1200 GMT 04/02/82

MASS 2.0

MASS 2.0

VV   VT 2100 GMT 04/02/82   INIT 1200 GMT 04/02/82

FIG. 10 MASS FORECASTED VERTICAL CROSS-SECTION OF VERTICAL VELOCITY
(MICROBARS PER SECOND)

SOUNDING LOCATIONS
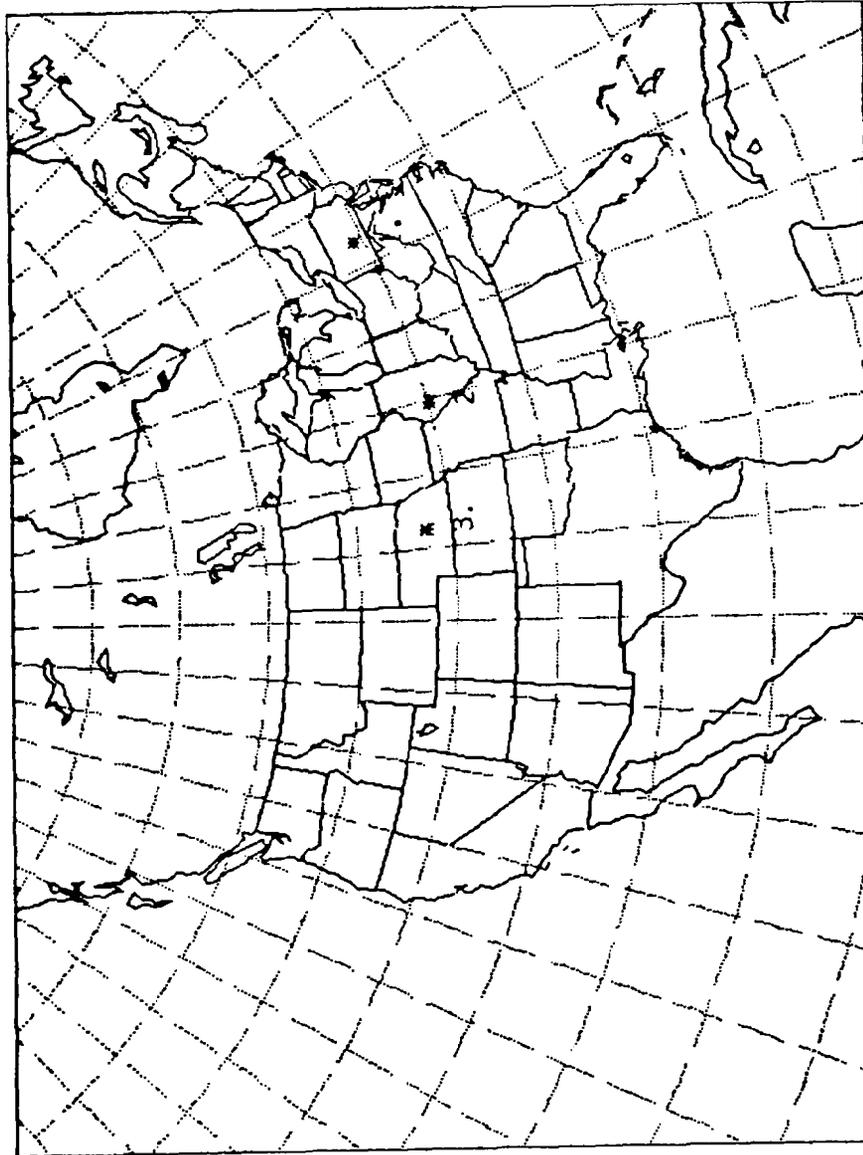
FIG. 11 SOUNDING LOCATOR MAP
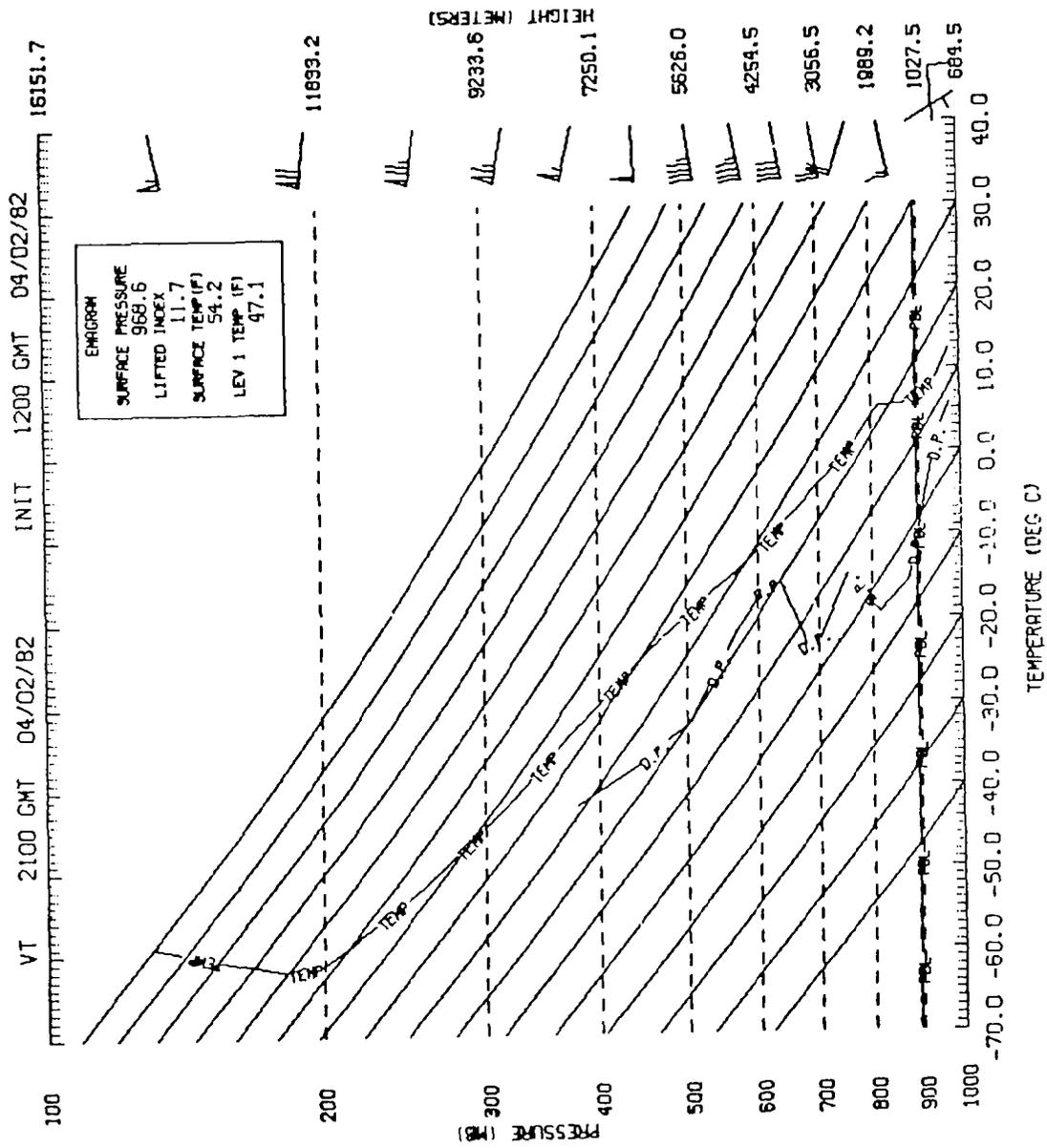
Fig. 12 MASS FORECASTED SOUNDING

12. REFERENCES

Kaplan, M.L., J.W. Zack, V.C. Wong, and J.J. Tuccillo, 1982: Initial Results from a Mesoscale Atmospheric Simulation System and Comparisons with the AVE-SESAME I Data Set. Mon. Wea. Rev., 110, 1564-1590.

Tuccillo, J.J., 1983: The Application of Pressure Gradient Force Averaging to the Euler-Backward Scheme. M.S. Thesis, Old Dominion University.