Initialization of the
PSU/NCAR Mesoscale Model

Ronald M. Errico
# Table of Contents

1. Introduction .................................................. 1

Part 1. MATHEMATICAL DEVELOPMENT

2. General Model Description .................................. 4
   2.1 Model equations ........................................ 8
   2.2 The balance condition .................................. 13
   2.3 Vertical modes .......................................... 17
   2.4 Transformed horizontal equations in terms of vertical modes .................................................................. 21
   2.5 Determination of surface pressure changes ......................... 22

3. Solution Procedure ............................................ 24

4. Characteristics of the PSU/NCAR Mesoscale Model .................. 26
   4.1 Vertical discretization ...................................... 26
   4.2 Horizontal discretization .................................... 30
   4.3 Horizontal boundary conditions ............................. 36

Part 2. VERTICAL MODE SOFTWARE

5. Introduction to Vertical Mode Software .......................... 38

6. Matrix Form of Vertical Discretization of the MM4 .................. 40

7. The Program VMODES ............................................ 42

Part 3. INITIALIZATION SOFTWARE

8. Introduction to the Initialization Software ........................ 46

9. Instructions in Job Control ..................................... 48

10. The Program INITAL ............................................. 51
   10.1 Variable names ............................................. 51
       10.1.1 Parameter definitions ................................ 52
       10.1.2 Namelist variables .................................. 53
       10.1.3 Variable arrays ....................................... 55
## Contents

10.2 Program list ........................................ 61
10.3 File formats ........................................ 65

11. The FSUB and QSUB Routines ..................... 67
   11.1 Subroutine F1SUB .................................. 67
   11.2 Subroutine F2SUB .................................. 68
   11.3 Subroutine F3SUB .................................. 69
   11.4 Subroutine QSUB .................................. 71

Part 4. USER’S GUIDE

   12.1 The JCL file ..................................... 74
   12.2 VMODES UPDATE file ............................ 77
   12.3 INITIAL UPDATE file ............................ 79
   12.4 MM4 UPDATE file ................................ 79
   12.5 NAMELIST data input files .................... 82
   12.6 The choice of NMODES and J1 .................. 84
   12.7 Explanation of printed diagnostics .......... 86
   12.8 Use of subroutine PRINTF ..................... 88

Part 5. SOFTWARE DEMONSTRATION

13. Vertical Modes of the MM4 ..................... 89
14. Convergence of Initialization Algorithm .......... 96
15. Initialized Modifications to Fields ............. 101
16. Comparison of MM4 Forecasts .................. 109

References ............................................. 118
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>List of mathematical symbols</td>
<td>5</td>
</tr>
<tr>
<td>10.1</td>
<td>List of variable symbols, pointer names, values of pointers, and references</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>for how the variables are calculated</td>
<td></td>
</tr>
<tr>
<td>10.2</td>
<td>Equivalencing of fields</td>
<td>60</td>
</tr>
<tr>
<td>13.1</td>
<td>Specified $\sigma_k$-levels, derived $\sigma_{k+\frac{1}{2}}$-levels,</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>standard atmospheric temperatures at data levels, and calculated equivalent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>depths for the 10-level MM4</td>
<td></td>
</tr>
<tr>
<td>13.2</td>
<td>Specified $\sigma_k$-levels, derived $\sigma_{k+\frac{1}{2}}$-levels,</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>standard atmospheric temperatures at data levels, and calculated equivalent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>depths for the 15-level MM4</td>
<td></td>
</tr>
<tr>
<td>13.3</td>
<td>The hydrostatic matrix for the MM4</td>
<td>93</td>
</tr>
<tr>
<td>13.4</td>
<td>Inverse of the hydrostatic matrix for the MM4</td>
<td>93</td>
</tr>
<tr>
<td>13.5</td>
<td>The matrix of vertical modes of the MM4</td>
<td>93</td>
</tr>
<tr>
<td>13.6</td>
<td>Inverse of the matrix of vertical modes of the MM4</td>
<td>93</td>
</tr>
<tr>
<td>15.1</td>
<td>The rms differences between initialized and non-initialized fields at each</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>$\sigma$-level at the initial time</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Fig. 4.1: The vertically discrete structure of the MM4 as it pertains to the initialization procedures. ........................................ 27

Fig. 4.2: The domain of the $46 \times 61$, 80 km grid of the MM4 centered over the United States. A portion of the grid structure, denoted by cross ($\times$) and dot (●) points is also presented. ........................................ 31

Fig. 12.1: The CRAY JCL file for initializing a dataset for the MM4. .............................. 75

Fig. 12.2: The file of CRAY UPDATE mods for the program that determines vertical modes of the 10-level MM4. ........................................ 78

Fig. 12.3: The file of CRAY UPDATE mods for the initialization program for the specified resolution. ........................................ 80

Fig. 12.4: The file of CRAY UPDATE mods for the adiabatic initialization of the MM4 with indicated resolution. ........................................ 81

Fig. 12.5: Data input files for a starting iteration of the program INITAL, the MM4, and a nonlinear iteration in the program INITAL. ........................................ 83

Fig. 13.1: The structures of the 10 vertical modes of the 10-level MM4, with $\sigma$-spacing and $\tilde{T}$ defined in Table 13.1. ........................................ 94

Fig. 13.2: The structures of the 15 vertical modes of the 15-level MM4, with $\sigma$-spacing and $\tilde{T}$ defined in Table 13.2. ........................................ 95

Fig. 14.1: Sample output from the program INITAL during execution of the first nonlinear iteration. ........................................ 97
Fig. 14.2: The rms time tendencies of vorticity, divergence, and pseudo-geopotential as a function of iteration number and vertical mode index.

Fig. 14.3: The rms adjustments (Δ) of vorticity, divergence, pseudo-geopotential, and surface pressure as a function of iteration number and vertical mode index.

Fig. 15.1: The domain of the cross grid for the demonstration experiment.

Fig. 15.2: (a.) The hydrostatic thickness between the surface and the 0.25 σ-level determined from the non-initialized analysis at the initial time. (b.) The difference in that field produced by the initialization.

Fig. 15.3: (a.) The u field at the 0.25 σ-level of the non-initialized analysis at the initial time. (b.) The difference in that field produced by the initialization.

Fig. 15.4: (a.) The T field at the 0.75 σ-level of the non-initialized analysis at the initial time. (b.) The difference in that field produced by the initialization.

Fig. 15.5: (a.) The T field at the 0.75 σ-level of the non-initialized analysis at the initial time. (b.) The difference in that field produced by the initialization.

Fig. 16.1: Time series of $p_s$, u, T, and $\omega$ (the last three at the 0.45 σ-level) at the point $\lambda=28$, $\phi=42$ (near Detroit, Michigan) for 15-hour forecasts begun from the initialized and non-initialized datasets for OZ 22 April 1981.

Fig. 16.2: (a.) The difference of the fields of hydrostatic thickness as forecast at hour 12 starting from initialized and non-initialized datasets. (b.) The error in the hydrostatic thickness field (non-initialized forecast minus analyzed field at verification time). The hydrostatic thickness fields are determined between the surface and the 0.25 σ-level.

Fig. 16.3: (a.) The difference of the u fields at the 0.25 σ-level as forecast at hour 12 starting from initialized and non-initialized datasets. (b.) The error in the u field
Fig. 16.4: (a.) The difference of surface pressure as forecast at hour 12 starting from initialized and non-initialized datasets. (b.) The error in the surface pressure field (non-initialized forecast minus analyzed field at verification time).

Fig. 16.5: (a.) The difference of the $T$ fields at the 0.75 $\sigma$-level as forecast at hour 12 starting from initialized and non-initialized datasets. (b.) The error in the $T$ field at the 0.25 $\sigma$-level (non-initialized forecast minus analyzed field at verification time).
This report describes one contribution by Ronald Errico of the Large-Scale Dynamics Section of NCAR’s Atmospheric Analysis and Prediction Division (AAP) to the Acid Deposition Modeling Program at NCAR. The intention of this program is to determine statistics of acid deposition in the United States through simulations as well as to produce software designed to forecast such deposition. The participation by AAP in this project regards the development of a weather forecasting model capable of providing high-resolution (spatial and temporal) wind and moisture data (forecast or simulated). The primary purpose of this contribution by Ronald Errico is to provide more realistic initial data for use by that model. This software has been designed for use with the Pennsylvania State University/NCAR mesoscale model designated MM4, although it is easily adaptable to other models. In addition to its function with initial data, it may be used as a diagnostic tool to investigate the model’s behavior. It was designed for those investigators who want software easy to both use and modify.
ACKNOWLEDGMENTS

The author wishes to thank the several people who helped make this work possible. The MM4 was originally developed at the Pennsylvania State University by Richard Anthes and Thomas Warner. The version referenced in this report was provided by Eirh-Yu Hsie and Ying-Hua Kuo. Marina Skumanich acted as liaison between the author and ADMP. Both she and Ying-Hua Kuo reviewed the manuscript and offered helpful suggestions. Although this project was in progress before the author saw the paper on mesoscale initialization by Bourke and McGregor, that paper was very encouraging since it demonstrated the success of the initialization technique.

This work was partially supported by the U.S. Environmental Protection Agency under Interagency Agreement DW930144-01-1, but has not been subject to EPA review procedures.
In this report, software designed to initialize a particular mesoscale model is described. That model is the one developed for the Acid Deposition Modelling Project (ADMP) of the National Center for Atmospheric Research (NCAR). It was developed jointly by scientists at NCAR and the Pennsylvania State University (PSU). This model has been designated MM4. The primary purpose of the initialization is to dynamically balance initial data for the MM4 so as to remove unrealistic inertial-gravitational oscillations from its forecasts. Since these oscillations are caused by propagating waves, hereafter they simply will be referred to as gravity waves. The initialization software will be designated by INIT.

The MM4 is described in Anthes et al. (1986). It is based on the model of Anthes and Warner (1978). Its domain is a limited area. Several options are available for specification of the lateral boundaries (INIT works with all). The grid is staggered using a Lambert conformal mapping. The vertical coordinate is $\sigma$, specified at discrete levels.

The initialization procedure used in INIT is based on the work of Errico (1979, 1982a), Leith (1980), and Bourke and McGregor (1983). Essentially, analyzed data is modified so as to satisfy a dynamic balance condition. That condition is described by a system of nonlinear equations resembling a quasi-geostrophic $\omega$-equation coupled with a nonlinear balance equation and a potential vorticity constraint. Specifically, a linearized potential vorticity is not changed by the initialization procedure. This constraint is consistent with the theories of geostrophic adjustment and nonlinear normal-mode initialization.
Although vertical structure functions are determined in order to easily invert relevant three-dimensional integral-differential operators, no normal modes are determined.

The INIT is written in FORTRAN-77 for the CRAY-1 computer. It can be easily adapted to other machines. The bulk of this software is stored in a CRAY UPDATE program library named READMP, and can easily be edited as such. The instructions in job control language (JCL) are in terms of the CRAY operating system (COS) version 1.13. This JCL allows for job-stepping loops.

The INIT includes distinct programs for deriving vertical structure functions for the MM4 and for performing the remaining tasks for nonlinear initialization. It can also be used to perform a variety of analyses of MM4 input or output. For these latter tasks, the INIT likely would need some minor modification.

The INIT is written in modular form. That is, distinct tasks are performed in distinct subroutines. This provides for easy modification and debugging as well as the potential for easy output of intermediate results. The code itself is extensively documented with internal comments. Most variable names are mnemonic. Many are described within the code itself.

Since the nonlinear initialization procedure requires calculation of the time tendencies of the dynamic fields, and since the procedure is iterative, the JCL is written using a loop structure. This is described in Section 9. The time tendencies are calculated by inputing data into the MM4, forecasting ahead one time step, outputing the forecast data, and then computing time tendencies by a forward difference approximation. The former
steps are done within the MM4. The last step is done in the INIT, where the tendencies are then used to compute appropriate modifications to the model input. The MM4 and INIT codes remain separate. They are alternately loaded and executed within the JCL loop. Datasets containing grid-point fields in the format of input or output MM4 files are passed between the two codes. In this way, many changes made to one code need not affect the other.

This technical report is divided into 5 parts. The initialization procedure is described mathematically in Part 1. This description is somewhat general, so that the presented procedures can be applied to almost any mesoscale model. However in this part of the report, the finite differencing used in the MM4 is also described in particular. Then in Part 2, the software for computing the vertical modes of the MM4 is presented. The part of the INIT that deals with the actual initialization of data is described in Part 3. A quick user's guide appears in Part 4. The user interested in only running the appropriate programs may just read this section. Finally, an example of the initialization of some real analyzed fields is presented in Part 5. Some vertical structure functions determined for the MM4 are presented there also.
Part 1 consists of Sections 2-4. In Section 2, the relevant equations are described rather generally, as they would apply to many primitive equation models. A step by step procedure that could be used to determine the initialized fields is presented in Section 3. Then, in Section 4, relevant details of the MM4 are presented as they apply to the initialization procedure. In particular, the vertical and horizontal discretization used in that model are described in detail.

Section 2. General Model Description

This section consists of five parts. First the model equations are presented in a form most suitable for describing the balance condition. Then the balance condition itself is presented. The method for obtaining the model's vertical modes is presented next. Their use simplifies the procedure for solving the balance equations. Both the model and balance equations are then described in terms of the vertical modes. Finally, a procedure for initializing the surface pressure is presented.
Table 2.1 List of mathematical symbols used in this report. Numbers in parentheses indicate section where variable is introduced.

- **A**: thermodynamic operator or matrix (2.1)
- **B**: hydrostatic operator or matrix (2.1)
- $C_p$: specific heat of air at constant pressure (2.1)
- $d$: discrete form of differential increment in hydrostatic equation (4.1)
- $f$: coriolis parameter (2.1)
- $f_0$: representative value of $f$ within region (2.1)
- $f$ with double subscripts indicates a generic field (4.2)
- $g$: acceleration of gravity (2.1)
- $g$ (with subscripts): the right hand side of a Helmholtz equation (4.2)
- $\hat{h}$: pseudo-geopotential as function of vertical-mode index (2.3)
- $\Delta h$: a modification to the $h$ field (2.2)
- $H$: an equivalent depth (2.3)
- **I**: the identity (product) matrix (2.3)
- $m^X$: a map-scale factor defined on cross grid (4.2)
- $m^*$: a map-scale factor defined on dot grid (4.2)
- $N$: a sum of nonlinear terms in either the $\zeta$, $\delta$, or $h$ prognostic equation (2.1)
- $N$ with index $I$ or $J$ is number of grid points in particular direction (4.1)
- $P$: linearized potential vorticity (2.2)
- $p$: pressure (2.1)
- $p_S$: surface pressure (2.1)
- $p_T$: fixed pressure at the model top (2.1)
- $p^* = p_S - p_T$ (2.1)
- $\bar{p}$: a single-valued, specified value of $p^*$ (2.1)
Table 2.1 (continued)

$\Delta p$ a change in $p^*$ due to one iteration of initialization procedure (2.5)
$Q$ specific humidity (8)
$R$ the gas constant (2.1)
$t$ time (2.1)
$T$ temperature (2.1)
$\bar{T}$ a specified mean temperature on $\sigma$ surfaces (2.1)
$T' = T - \bar{T}$ a (linearized) perturbation temperature (2.1)
$T_0$ a standard surface temperature (7)
$\bar{T}$ a standard atmospheric temperature (7)
$T_x$ a standard stratospheric temperature (7)
$\Delta T$ change in $T$ due to one iteration of initialization procedure (2.5)
$u$ eastward component of velocity (2.1)
$v$ northward component of velocity (2.1)
x horizontal coordinate, increasing eastward (2.1)
$\Delta x$ the grid spacing
$y$ horizontal coordinate, increasing northward (2.1)
z a component of a vertical mode (2.3)
$\hat{z}$ a vertical-mode structure function or vertical mode (2.3)
$z_0$ a standard atmospheric height (7)
$z_x$ a standard tropopause height (7)
$Z$ a matrix whose columns are eigenvectors (2.3)
$\delta$ horizontal divergence (2.1)
$\hat{\delta}$ horizontal divergence as a function of vertical-mode index (2.3-4)
$\delta^N$ a new iterated value for the balanced divergence (2.2)
$\delta^O$ a former iterated value for the balanced divergence (2.2)
$\Delta \zeta$ a modification to the divergence field (2.2)
Table 2.1 (continued)

$\zeta$ relative vorticity on $\sigma$-surfaces (2.1)

$\dot{\zeta}$ relative vorticity as a function of vertical-mode index (2.3-4)

$\zeta^N$ a new iterated value for the balanced vorticity (2.2)

$\zeta^O$ a former iterated value for the balanced vorticity (2.2)

$\Delta \zeta$ a modification to the vorticity field (2.2)

$\bar{\theta}$ potential temperature determined from $\bar{T}$ (4.1)

$\kappa = R/C_p$ (2.1)

$\lambda$ square of inverse of deformation scale (2.4)

$\sigma = (p - p_T)/p^*$ vertical coordinate (2.1)

$\dot{\sigma} = d\sigma/dt$ (2.1)

$\tau$ the “tau” operator or matrix (2.1)

$\phi$ the (hydrostatic) geopotential (2.1)

$\phi_s$ geopotential of surface topography (2.1)

$\chi$ velocity potential (2.1)

$\psi$ stream function (2.1)

$\omega = dp/dt$ (2.1)

$\nabla^2$ 2-dimensional (horizontal) Laplacian (2.1)
2.1 Model equations

The adiabatic primitive equations may be written as:

\[
\begin{align*}
\frac{\partial u}{\partial t} &= -u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} - \dot{\sigma} \frac{\partial u}{\partial \sigma} + f v - \frac{\partial \phi}{\partial x} - R T \frac{\sigma}{\sigma + p T / p^*} \frac{1}{p^*} \frac{\partial p^*}{\partial x}, \\
\frac{\partial v}{\partial t} &= -u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y} - \dot{\sigma} \frac{\partial v}{\partial \sigma} - f u - \frac{\partial \phi}{\partial y} - R T \frac{\sigma}{\sigma + p T / p^*} \frac{1}{p^*} \frac{\partial p^*}{\partial y}, \\
\frac{\partial T}{\partial t} &= -u \frac{\partial T}{\partial x} - v \frac{\partial T}{\partial y} - \dot{\sigma} \frac{\partial T}{\partial \sigma} + \frac{\kappa T}{\sigma + p T / p^*} \frac{\omega}{p^*}, \\
\frac{\partial p^*}{\partial t} &= -\int_0^1 \left( \frac{\partial p^* u}{\partial x} + \frac{\partial p^* v}{\partial y} \right) d\sigma, \\
\dot{\phi} &= \phi_t + R \int_1^\sigma \frac{T}{\sigma + p T / p^*} d\sigma, \\
\dot{\sigma} &= \frac{\sigma}{p^*} \int_0^1 \left( \frac{\partial p^* u}{\partial x} + \frac{\partial p^* v}{\partial y} \right) d\sigma - \frac{1}{p^*} \int_0^\sigma \left( \frac{\partial p^* u}{\partial x} + \frac{\partial p^* v}{\partial y} \right) d\sigma, \\
\frac{\omega}{p^*} &= \dot{\phi} + \frac{\sigma}{p^*} \frac{\partial p^*}{\partial t} + \frac{\sigma}{p^*} \left( u \frac{\partial p^*}{\partial x} + v \frac{\partial p^*}{\partial y} \right),
\end{align*}
\]

where

- \(x, y\) are horizontal coordinates,
- \(t\) is time,
- \(u\) is the eastward component of velocity,
- \(v\) is the northward component of velocity,
- \(T\) is temperature,
- \(\phi\) is the (hydrostatic) geopotential,
- \(p\) is pressure,
\[ \sigma = (p - p_T)/p^* \] is the vertical coordinate,

\[ p^* = p_S - p_T, \text{ with} \]

subscript \( S \) referring to a surface value,

subscript \( T \) referring to value at the top of the model,

\[ \dot{\sigma} = d\sigma/dt, \]

\[ \omega = dp/dt, \]

\( f \) is the coriolis parameter,

\( R \) is the gas constant,

\[ \kappa = R/C_p, \text{ where } C_p \text{ is the specific heat of air at constant } p, \]

and \( g \) is the acceleration of gravity.

Map scale factors have been excluded from the equations in order to simplify the presentation. They will be reintroduced in conjunction with the discussion of finite differences in Section 4.2. The boundary conditions at both \( \sigma = 0 \) and \( 1 \) is that \( \dot{\sigma} = 0 \). The model also requires specification of lateral boundaries, which need not be discussed in this report.

Most details of (2.1.1)-(2.1.7) are unimportant for explanation of the initialization procedure. Therefore, at this point the notation is simplified by distinguishing a set of relevant linear terms from the remaining terms. These linear terms are those which remain after (2.1.1)-(2.1.7) are linearized about a state of rest \( (u = 0, v = 0) \) with \( T = \bar{T} \) (a
being some representative nonzero value for the model domain. The linearized equations are

\[
\begin{align*}
\frac{\partial u}{\partial t} &= f_0 v - \frac{\partial}{\partial x}(\phi' + RT\frac{\sigma}{\sigma + pT/\bar{p}} p'), \\
\frac{\partial v}{\partial t} &= -f_0 u - \frac{\partial}{\partial y}(\phi' + RT\frac{\sigma}{\sigma + pT/\bar{p}} p'), \\
\frac{\partial T'}{\partial t} &= -\frac{\rho}{\bar{p}} \frac{\partial T}{\partial \sigma} + \frac{\kappa T}{\sigma + pT/\bar{p}} \frac{\omega}{\bar{p}}, \\
\frac{\partial p'}{\partial t} &= -\bar{p} \int_0^1 \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, d\sigma, \\
\phi' &= \phi_s + R \int_1^\sigma \frac{T'}{\sigma + pT/\bar{p}} \, d\sigma, \\
\dot{\sigma} &= \sigma \int_0^1 \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, d\sigma - \int_0^\sigma \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, d\sigma, \\
\frac{\omega}{\bar{p}} &= \dot{\sigma} - \sigma \int_0^1 \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, d\sigma,
\end{align*}
\]

where \( T'(x,y,t,\sigma) = T(x,y,t,\sigma) - \bar{T}(\sigma) \), and \( p'(x,y,t) = \bar{p}'(x,y,t) - \bar{p} \).

It is easier to develop the initialization procedure in terms of vorticity \( \zeta \) and divergence \( \delta \) defined as

\[
\begin{align*}
\zeta &= \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \\
\delta &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}.
\end{align*}
\]

These are related to the stream function \( \psi \) and velocity potential \( \chi \) by

\[
\begin{align*}
\nabla^2 \psi &= \zeta, \\
\nabla^2 \chi &= \delta.
\end{align*}
\]
where

\[ \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \]  

(2.1.19)

is the two-dimensional Laplacian, operating on a sigma-surface. In terms of \( \psi \) and \( \chi \), the velocity components are

\begin{align*}
    u &= \frac{\partial \chi}{\partial x} - \frac{\partial \psi}{\partial y}, \\
    v &= \frac{\partial \chi}{\partial y} + \frac{\partial \psi}{\partial x}.
\end{align*}

(2.1.20, 2.1.21)

In terms of \( \zeta \) and \( \delta \), (2.1.8) and (2.1.9) are transformed to

\[ \frac{\partial \zeta}{\partial t} = -f \zeta, \]  

(2.1.22)

\[ \frac{\partial \delta}{\partial t} = f \phi' - \nabla^2 (\phi' + RT \frac{\sigma}{\sigma + p_{T}/p} \frac{p'}{\bar{p}}). \]  

(2.1.23)

The only prognostic variable upon which the linearized \( \partial T'/\partial t \) depends is \( \delta \).

Therefore, (2.1.10) may be written as

\[ \frac{\partial T'}{\partial t} = A \delta, \]  

(2.1.24)

where \( A \) is an integral-differential operator (in \( \sigma \)) which may be determined by substituting (2.1.13) and (2.1.14) into (2.1.10). This operator is also a function of \( \bar{T} \) and \( p_{T}/\bar{p} \). In a similar fashion, (2.1.12) can be written as

\[ \phi' = RB T' + \phi_s, \]  

(2.1.25)

where \( B \) is another integral operator, also in \( \sigma \). Neither of these operators will be described further here, but their analogs for the discrete vertical grid of the MM4 will be presented in detail in Section 6.
The $\zeta$ tendency is independent of both $T'$ and $p'$. The $\delta$ tendency depends on those variables only in the form of $\nabla^2$ of

$$h = \phi' + R\tau \left( \frac{\sigma}{\sigma + p_T/p} \right) \frac{\nu'}{\bar{p}}.$$  

(2.1.26)

That this quantity represents the pressure force is a result of using $\sigma$ as the vertical coordinate. Here, $h$ will be called the pseudo-geopotential. Its time tendency is determined by combining (2.1.24), (2.1.25) and (2.1.11) to yield

$$\frac{\partial h}{\partial t} = -\tau(\delta),$$  

(2.1.27)

where $\tau$ is the differential-integral $\sigma$-coordinate operator

$$\tau(\ ) = -RBA(\ ) + \frac{RT\sigma}{\sigma + p_T/p} \int_0^1 (\ )d\sigma.$$ 

(2.1.28)

The complete nonlinear equations may be written as

$$\frac{\partial \zeta}{\partial t} = -f_0\delta + N_z,$$  

(2.1.29)

$$\frac{\partial \delta}{\partial t} = f_0\zeta - \nabla^2 h + N_\delta,$$  

(2.1.30)

$$\frac{\partial h}{\partial t} = -\tau\delta + N_h,$$  

(2.1.31)

together with (2.1.4). All the terms not explicitly described appear as $N$ with a suitable subscript referring to the equation in which each appears. In particular, the $N$'s include all nonlinear terms and terms involving horizontal variations of $f$. In the forthcoming development, the form of the $N$'s need not be considered explicitly.
2.2 The balance condition

If solutions to the primitive equations are to be characterized by slow behavior,
then initially
\[
\frac{\partial \delta}{\partial t} = 0, \quad (2.2.1)
\]
and
\[
\frac{\partial^2 \delta}{\partial t^2} = 0, \quad (2.2.2)
\]
are required (Phillips, 1960; Daley, 1980; Leith, 1980; Errico, 1982a; and Bourke and McGregor, 1983). These equations are equivalent to
\[
f_0 \zeta - \nabla^2 h = -N \delta, \quad (2.2.3)
\]
\[
(f_0^2 r^{-1} - \nabla^2) \delta = r^{-1}(f_0 N \delta - \nabla^2 N h + \frac{\partial}{\partial t} N \delta), \quad (2.2.4)
\]
respectively. The last term in (2.2.4) may usually be neglected.

Another way of interpreting this balance condition is in terms of the model’s normal modes and geostrophic adjustment theory. These modes are the solutions of the linearized model (2.1.8)–(2.1.14). The modes which propagate in time are called gravity waves. At any particular horizontal scale, the amplitude of such a wave may be determined as a linear combination of the right hand sides of (2.2.3) and (2.2.4). This linear combination actually involves linear differential operators, and not simply algebraic sums of the fields.

Instead of deriving the balance equations from the conditions (2.2.1) and (2.2.2), they can be derived from the condition that the time tendencies of the amplitudes of
gravitational waves are negligible compared with other terms in the prognostic equations for those amplitudes. This condition replaces (2.2.2) with

$$\frac{\partial}{\partial t} (f_0 \zeta - \nabla^2 h) = 0,$$  \hfill (2.2.5)

which is identical to (2.2.4) with the $\partial N_\delta/\partial t$ term omitted.

As a gravity wave propagates, energy is exchanged between the $f_0 \zeta - \nabla^2 h$ and $\delta$ fields. If only one of (2.2.1) or (2.2.5) is imposed in an initialization scheme, that scheme will likely be ineffective since the unbalanced portion of the wave amplitude will quickly act to unbalance the initialized portion through this energy exchange. Therefore both (2.2.1) and (2.2.5) must be satisfied simultaneously. For further description of these ideas regarding balance see Leith (1980), Errico (1981, 1982a, 1982b), or Bourke and McGregor (1983).

The system (2.2.1) and (2.2.5) is not closed since there are only two equations for the three unknown fields $\zeta$, $\delta$, and $h$. Therefore, another relation is needed. The one chosen here is that the linearized potential vorticity $P$ is not changed by the initialization procedure, where

$$P = \zeta - f_0 \tau^{-1} h.$$  \hfill (2.2.6)

This condition is consistent with the constraint that the rotational modes, as used in non-linear normal-mode initialization (Machenhauer, 1977; Leith, 1980; Errico, 1982a; Bourke and McGregor, 1983), are unaffected by the initialization procedure. The rotational modes are equivalent to geostrophic Rossby waves when the latitudinal variation of the coriolis parameter is vanishingly small.
The complete set of equations for the balance condition is

\begin{align}
(f_o^2 r^{-1} - \nabla^2) \zeta &= -f_o r^{-1} N_\delta - \nabla^2 P, \quad (2.2.7) \\
(f_o^2 r^{-1} - \nabla^2) \delta &= \tau^{-1} (f_o N_\zeta - \nabla^2 N_h), \quad (2.2.8) \\
h &= \tau f_o^{-1} (\zeta - P). \quad (2.2.9)
\end{align}

These equations are highly nonlinear since the \( N \)'s depend on \( \zeta, \delta, \) and \( h \) in a complicated way. Therefore, these equations are usually solved iteratively. A simple procedure is to assume a first "guess" for the three fields, compute the \( N \)'s consistent with those fields, and then compute new fields by solving (2.2.7)-(2.2.9) with the \( N \)'s given by the previous step (i.e., fixed). The process is repeated, using the previously determined fields as the next "guess", until changes between one iteration and the next are small.

Computationally, there is a simpler, but equivalent procedure. Values for the new iterate (superscript \( N \)) and previous (or old) iterate (superscript \( 0 \)) of the fields are related by

\begin{align}
\zeta^N &= \zeta^O + \Delta \zeta, \\ \delta^N &= \delta^O + \Delta \delta, \\ h^N &= h^O + \Delta h, 
\end{align}

where the changes (designated by \( \Delta \)) to the fields are the solutions of

\begin{align}
(f_o^2 r^{-1} - \nabla^2) \Delta \zeta &= -f_o r^{-1} \frac{\partial \delta}{\partial t}, \\
(f_o^2 r^{-1} - \nabla^2) \Delta \delta &= \tau^{-1} (f_o \frac{\partial \zeta}{\partial t} - \nabla^2 \frac{\partial h}{\partial t}), \\
\Delta h &= \tau f_o^{-1} \Delta \zeta.
\end{align}
The time tendencies appearing on the left hand sides of (2.2.13)-(2.2.14) are those determined from the old iterate of the fields; e.g.,

$$\frac{\partial \zeta}{\partial t} = -f_0 \delta^0 + N_i^0.$$ (2.2.16)

Equations (2.2.13) – (2.2.14) require horizontal boundary conditions to determine $\Delta \zeta$ and $\Delta \delta$ from the respective right hand sides. Boundary conditions are also required for determination of $\Delta \psi$ and $\Delta \chi$ from $\Delta \zeta$ and $\Delta \delta$ using (2.1.17) and (2.1.18), respectively. Since the model is actually described in terms of $u$ and $v$ these boundary values should be related to desired conditions for $\Delta u$ and $\Delta v$ at those boundaries. The boundary values for $\Delta T$ and $\Delta p$, and therefore $\Delta h$ must also be considered when choosing a condition for $\Delta \zeta$ because of the relation (2.2.15).

In practice, choice of boundary conditions should be determined after consideration of several aspects. It may not be possible to apply conditions for $\Delta u$, $\Delta v$, $\Delta T'$ and $\Delta p$ consistently to (2.1.13)-(2.2.15) so that solutions exist and can be determined by some not-too-difficult computational means. This problem becomes more acute when the discrete model grid is considered. It must be remembered that the purpose of the initialization is to remove unrealistic gravity wave “noise” from model forecasts. Therefore application of a particular boundary condition is successful only if the desired result of removing such noise is attained. Finally, any changes produced at the boundaries due to application of the initialization procedure must be examined with respect to resulting changes in the model forecasts. Depending on the lateral boundary conditions used by the model, changes at the boundaries may affect forecasts more significantly than changes in the interior (Errico
and Baumhefner, 1986; Anthes et al., 1985). The question of which boundary conditions are most appropriate is still open and a subject of continuing research.

2.3 Vertical modes

In order to solve (2.2.13) and (2.2.14) it is necessary to invert the three-dimensional integral-differential operator \((f_0^{-2}r^{-1} - \nabla^2)\). Fortunately, since \(r\) is independent of \(x\) and \(y\), and \(\nabla\) is independent of \(\sigma\), solutions to (2.2.13)-(2.2.14) are separable in the horizontal and vertical coordinates. In particular, the equations may be simplified by using a transformation in terms of the eigenvectors of \(r\). These eigenvectors are called the vertical structure functions or vertical modes of the linearized primitive equations (2.1.8)-(2.1.14).

In this section, the procedure for determining the vertical modes is first described for the domain where \(\sigma\) is continuously defined in the interval [0,1], and then for a vertically discretized domain where the fields are defined at only a finite number of \(\sigma\)-levels. The latter case applies to almost all numerical models, and the resulting eigenvalue problem is simply described in terms of matrices (rather than integral-differential linear operators for the continuous case).

The eigenvectors \(z_i\) of \(r\) are determined as the solutions to

\[
(gH_i - r)z_i = 0,
\]

(2.3.1)
where the $H_i$ are corresponding eigenvalues. There are an infinite number of such solutions. Values of $H_i$ are called equivalent depths, for reasons which will be made apparent later.

A set of eigenvectors $\bar{z}_i$ can also be determined for the adjoint problem

$$ (gH_i - \tau)^{-1} \bar{z}_i = 0, $$

(2.3.2)

where

$$ \int_0^1 \bar{z}_i \bar{z}_j d\sigma = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} $$

(2.3.3)

Both $\bar{z}_i$ and $z_i$ are independent of $x$ and $y$.

Under certain conditions, when the specified field of $\bar{T}(\sigma)$ is statically stable, all the $H_i$ can be shown to be positive valued, with a maximum value $H_1$, called the equivalent depth of the external mode. In this case, the modes can be ordered in terms of decreasing values of $H_i$; i.e.,

$$ H_i > H_j \quad \text{for} \quad i < j. $$

(2.3.4)

Also in this case, each successively so-ordered $z_i$ has one more zero crossing. The external mode has no zero-crossings; i.e., it is nearly barotropic. Further discussion of the general properties of this eigenvalue problem are beyond the scope of this report. However, an example solution for the vertically-discrete MM4 is presented in Section 13.

The three-dimensional fields of $\zeta$, $\delta$, and $h$ can each be expanded in terms of the $z_1$ by utilizing (2.3.3). For example, for $h$,

$$ h(x, y, \sigma, t) = \sum_{i=1}^{\infty} z_i(\sigma) \hat{h}_i(x, y, t), $$

(2.3.5)
where
\[ \hat{h}_i(x,y,t) = \int_0^1 z_i(\sigma) h(x,y,\sigma,t) d\sigma. \] (2.3.6)

The \( \hat{h}_i \) may be considered as the amplitudes of the vertical modes \( z_i \) at the location \( x, y \) and time \( t \).

For the model which uses fields defined on a finite number \( K \) of discrete \( \sigma \)-levels, \( \sigma_1, \ldots, \sigma_K \), the operator \( \tau \) is appropriately defined as a \( K \times K \) matrix (the matrix for the MM4 will be described in Section 6). The normal modes \( z_i \) are the \( K \)-dimensional eigenvectors satisfying
\[ (gH_i - \tau)z_i = 0. \] (2.3.7)

There are \( K \) independent solutions. If \( Z \) is the \( K \times K \) matrix whose columns are the eigenvectors \( z_i \), and \( H \) is the diagonal matrix of correspondingly-ordered \( H_i \), then
\[ gHZ = \tau Z. \] (2.3.8)

The 3-dimensional fields of \( \zeta, \delta, \) and \( h \) can be represented as one-dimensional vectors of two-dimensional fields. For example,
\[ h(x,y,t) = (h(x,y,\sigma_1,t), h(x,y,\sigma_2,t), \ldots, h(x,y,\sigma_K,t))^T, \] (2.3.9)
where superscript \( T \) indicates a vector or matrix transpose. A field in terms of discrete \( \sigma \)-coordinates can be transformed into a field in terms of vertical-mode amplitudes, analogous to (2.3.6) and (2.3.5) by
\[ \hat{h} = Z^{-1}h, \] (2.3.10)
\[ h = Zh, \] (2.3.11)
and similarly for the fields of $\zeta$, $\delta$, and the $N$'s. The columns of $Z^{-1}$ correspond to the $z_i$ described in (3-2) for the continuous-$\sigma$ case, so that

$$I = ZZ^{-1},$$

(2.3.12)

where $I$ is the diagonal (identity) matrix whose nonzero components all equal 1.

In practice, the results are not very sensitive to the reasonable specification of $\bar{T}$, and they are independent of the value of $\bar{p}$. Therefore, the matrices $\tau$, $Z$, $Z^{-1}$, and the eigenvalues $H_i$ may be computed once, and then stored.
In terms of the vertical mode amplitudes, (2.1.29)–(2.1.31) are replaced by

\[
\frac{\partial \xi_i}{\partial t} = -f_0 \delta_i + \hat{N}_\xi_i, \tag{2.4.1}
\]

\[
\frac{\partial \delta_i}{\partial t} = f_0 \xi_i - \nabla^2 h_i + \hat{N}_{\delta_i}, \tag{2.4.2}
\]

\[
\frac{\partial h_i}{\partial t} = -g H_i \delta_i + \hat{N}_{hi}, \tag{2.4.3}
\]

where the subscript \(i\) refers to a particular vertical mode. Since the \(N\)'s are nonlinear, the equations with distinct values of \(i\) are fact coupled (e.g., nonlinear vertical advection acts to couple the modes). If only one value of \(i\) is considered, (2.4.1)–(2.4.3) is a system of shallow water equations for a mean depth \(H_i\).

In the case of the continuously defined \(\sigma\), there are an infinite number of such equations. However the linearized form of each such equation includes only two-dimensional (horizontal) operators or fields. Thus, a finite number (3) of three-dimensional equations have been replaced by an infinite number of two-dimensional equations. However, the real advantage of the transformed system (2.4.1)–(2.4.3) is obtained when \(\sigma\) is discrete. Then the number of equations remains the same (3 for each \(\sigma\)-level), but the linear terms are now decoupled in \(\sigma\).

In terms of the vertical modes, the three dimensional operator \((f_0^2 \tau^{-1} - \nabla^2)\) is replaced by a set of two-dimensional operators \((\lambda_i - \nabla^2)\) where

\[
\lambda_i = f_0^2/(g H_i) \tag{2.4.4}
\]
is the squared inverse of the Rossby radius of deformation corresponding to the depth H.

There is one value of \( \lambda \) for each discrete equivalent depth. Thus, the balanced fields are determined by iterating

\[
(\lambda_a - \nabla^2) \Delta \delta_i = -\frac{\lambda}{f_0} \frac{\partial \delta_i}{\partial t},
\]

\[
(\lambda_i - \nabla^2) \Delta \delta_i = \frac{\lambda}{f_0^2} (f_0 \frac{\partial \delta_i}{\partial t} - \nabla^2 \frac{\partial \delta_i}{\partial t}),
\]

\[
\Delta \delta_i = f_0 \lambda_a^{-1} \Delta \delta_i.
\]

In practice, it is simplest to compute the time-tendencies by using the relation

\[
\frac{\partial \delta_i}{\partial t} = \frac{\zeta(t) - \zeta(t = 0)}{\Delta t},
\]

where \( \Delta t \) is a short time. Similar expressions are used for the other time-tendencies.

If the numerical model uses a forward step initially, and if \( \Delta t \) is the time step, then (2.4.8) is exactly the correct expression. In that case, it is advantageous to use (2.4.8) if the numerical model directly outputs fields at various times, rather than time tendencies.

Of course, the model software can instead be modified to output tendencies, but in the remainder of this report, the former method will be assumed.

### 2.5 Determination of surface pressure changes

The balance condition described in Section 2.2 is defined in terms of \( h \) which is a function of both \( T' \) and \( p' \). Since the latter are the model fields, it is necessary to invert (2.1.26) to determine those balanced fields from that of \( h \). Specifically, since field changes
are computed in the initialization procedure, it is necessary to determine $\Delta T$ and $\Delta p$ from $\Delta h$. Note that there are more unknown fields than known fields. However, Temperton and Williamson (1981) show one way to close the system.

In general, the method of Temperton and Williamson (1981) is to first compute

$$\Delta p = \bar{p} \int_0^1 r^{-1} \Delta h \sigma.$$  (2.5.1)

Then, the temperature changes are computed as

$$\Delta T = R^{-1} B^{-1} (\Delta h - RT \frac{\sigma}{\sigma + p_T / \bar{p}} \Delta p)$$  (2.5.2)

For further discussion of this closure, the interested reader should consult Temperton and Williamson (1981) and the references they cite.
The specific sequence of calculations needed to balance the dynamic fields is

1. Compute $\zeta, \delta, h$ from $u, v, T, p^*$ at $t = 0$ using (2.1.15), (2.1.16), and (2.1.26).

2. Compute $\hat{\zeta}, \hat{\delta}, \hat{h}$ from $\zeta, \delta, h$ using (2.3.10).

3. Run the model one time step to determine fields at $t = \Delta t$.

4. Repeat steps 1 and 2 with the data at $t = \Delta t$.

5. Compute time tendencies of $\dot{\zeta}, \dot{\delta}, \dot{h}$ using (2.4.8).

6. Compute the right hand sides of (2.4.5) and (2.4.6).

7. Solve the Helmholtz equations (2.4.5) and (2.4.6) for $\Delta \hat{\zeta}_i$ and $\Delta \hat{\delta}_i$ for some set of $i$ and suitable lateral boundary conditions.

8. Compute $\Delta \hat{h}_i$ using (2.4.7).

9. Compute velocity modifications in terms of fields of $\Delta \hat{\psi}_i$ and $\Delta \hat{\chi}_i$ corresponding to $\Delta \hat{\zeta}_i$ and $\Delta \hat{\delta}_i$, respectively by inverting (2.1.17) and (2.1.18) with suitable lateral boundary conditions.

10. Compute velocity modifications in terms of $\Delta \hat{u}_i$ and $\Delta \hat{v}_i$ using (2.1.20) and (2.1.21).

11. Compute fields of $\Delta u$, $\Delta v$, and $\Delta h$ using (2.3.11).
12. Compute changes to the surface pressure field ($p^*$) using (2.5.1).

13. Compute changes to $T$ using (2.5.2).

14. Compute new values to the initial fields using expressions analogous to (2.2.10)–(2.2.12).

15. Repeat the entire procedure using the new values from step 14.

Actually, the above steps may be re-ordered in some ways. For example, step 11 may precede step 9, transforming different fields. The above order is that used in the INIT, with minor modifications. These modifications and the above order can be used to reduce the number of computations.
Section 4. Characteristics of the PSU/NCAR Mesoscale Model

The PSU/NCAR mesoscale model, designated MM4, is partly described in Anthes and Warner (1978). The INIT software is particularly designed to be used with that model, although the sequence of operations would be similar for other models as well.

The MM4 uses a flux form of the primitive equations. That is, the prognostic variables are actually $p^*u$, $p^*v$, and $p^*T$, rather than $u$, $v$, and $T$. Fortunately, this does not change the balance conditions or the method of solution, except for the way the data is handled. This latter exception occurs only because the MM4 software is designed to only input and output data in flux form. However, this is simply corrected for in the INIT software by quickly removing the $p^*$ factors by dividing the flux forms by the those factors.

4.1 Vertical discretization

In the MM4, $u$, $v$, $T$, $\omega$, and $\phi$ are defined on $K$ $\sigma$-surfaces. The values of $\sigma$ which specify those surfaces are indexed with values $1 + \frac{1}{2}, 2 + \frac{1}{2}, \ldots, K + \frac{1}{2}$. The field of $\dot{\sigma}$ is defined on $K + 1$ surfaces, indexed with integral values $1, 2, \ldots, K + 1$. The former surfaces will be called half-levels and the latter called full-levels. Thus only $\dot{\sigma}$ is defined on full-levels. The vertically discrete structure of the MM4, as it pertains to the initialization software, is presented in Fig. 4.1.
Fig. 4.1: The vertically discrete structure of the MM4 as it pertains to the initialization procedures.
Values of $\sigma$ at full-levels are specified by the user with the conditions that $\sigma_1 = 0$, $\sigma_{K+1} = 1$, and $\sigma_k < \sigma_{k+1}$ for all $k$. Values of $\sigma$ at half-levels are determined as arithmetic averages of $\sigma$-values at adjacent full-levels:

$$\sigma_{k+\frac{1}{2}} = \frac{1}{2}(\sigma_k + \sigma_{k+1}).$$  \hspace{1cm} (4.1.1)

The thickness of each layer of data is defined by

$$\Delta \sigma_{k+\frac{1}{2}} = \sigma_{k+1} - \sigma_k.$$  \hspace{1cm} (4.1.2)

In the MM4 the integral operator in (2.1.11) is discretized as

$$\int_0^1 \delta d\sigma = \sum_{k=1}^{K} \delta_{k+\frac{1}{2}} \Delta \sigma_{k+\frac{1}{2}}.$$  \hspace{1cm} (4.1.3)

Similarly, corresponding to (2.1.13),

$$\dot{\sigma}_k = \sigma_k \sum_{k'=1}^{K} \delta_{k'+\frac{1}{2}} \Delta \sigma_{k'+\frac{1}{2}} - \sum_{k'=1}^{K} \delta_{k'+\frac{1}{2}} \Delta \sigma_{k'+\frac{1}{2}}.$$  \hspace{1cm} (4.1.4)

The discretization of (2.1.14) requires a value of $\dot{\sigma}$ at half-levels. These later values are determined by linear interpolation of values at full-levels. The complete expression is

$$\left(\frac{\omega}{p}\right)_{k+\frac{1}{2}} = \frac{1}{2}(\dot{\sigma}_k + \dot{\sigma}_{k+1}) - \sigma_{k+\frac{1}{2}} \sum_{k'=1}^{K} \delta_{k'+\frac{1}{2}} \Delta \sigma_{k'+\frac{1}{2}}.$$  \hspace{1cm} (4.1.5)

The hydrostatic formulation used in the MM4 is

$$\phi_{k+\frac{1}{2}} = R \sum_{k<j\leq K} \left( \frac{T_{j+\frac{1}{2}} \Delta \sigma_{j+\frac{1}{2}} + T_{j-\frac{1}{2}} \Delta \sigma_{j-\frac{1}{2}}}{\Delta \sigma_{j+\frac{1}{2}} + \Delta \sigma_{j-\frac{1}{2}}} \right) d_j + RT_{K+\frac{1}{2}} d_{K+1} + \phi_x.$$  \hspace{1cm} (4.1.6)
where

\[ d_j = \ln\left[\frac{(\sigma_{j+\frac{1}{2}} + p_T/p^*)}{(\sigma_{j-\frac{1}{2}} + p_T/p^*)}\right], \quad (4.1.7) \]

for \( j = 1, \ldots, K \), and

\[ d_{K+1} = \ln\left[\frac{(1 + p_T/p^*)}{(\sigma_{K+\frac{1}{2}} + p_T/p^*)}\right]. \quad (4.1.8) \]

The first term on the right hand side of (2.1.10) is actually determined as the sum of two terms:

\[ -\frac{\partial \dot{\sigma} T}{\partial \sigma} = T \frac{\partial \dot{\sigma}}{\partial \sigma} - \frac{\partial \dot{\sigma} \dot{T}}{\partial \sigma}. \quad (4.1.9) \]

In the MM4, these two terms are actually discretized in very different manners. The first term is appropriately discretized as

\[ \left(T \frac{\partial \dot{\sigma}}{\partial \sigma}\right)_{k+\frac{1}{2}} = -\dot{T}_{k+\frac{1}{2}} \left(\delta_{k+\frac{1}{2}} - \sum_{k'=1}^{K} \delta_{k'+\frac{1}{2}} \Delta \sigma_{k'+\frac{1}{2}}\right), \quad (4.1.10) \]

where (4.1.4) has been used. The second term uses an interpolated value of temperature at full levels. The interpolation is linear in potential temperature \( \theta \), which is related to \( T \) by

\[ T = \left(\frac{P}{p^*}\right)^{\kappa} \theta. \quad (4.1.11) \]

The discrete expression for the second term in (4.1.9) is therefore

\[ \left(\frac{\partial \dot{\sigma} \dot{T}}{\partial \sigma}\right)_{k+\frac{1}{2}} = (\sigma_{k+1} \ddot{T}_{k+1} - \dot{\sigma}_k \ddot{\theta}_k)(\Delta \sigma_{k+\frac{1}{2}})^{-1}, \quad (4.1.12) \]

where

\[ \ddot{T}_k = \ddot{\theta}_k (\sigma_k + p_c/p^*)^{-\kappa}, \quad (4.1.13) \]
\[
\bar{\theta}_k = \bar{\theta}_{k-\frac{1}{2}} \left( \frac{\sigma_{k+\frac{1}{2}} - \sigma_k}{\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}} \right) + \bar{\theta}_{k+\frac{1}{2}} \left( \frac{\sigma_k - \sigma_{k-\frac{1}{2}}}{\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}} \right),
\]
and
\[
\bar{\theta}_{k+\frac{1}{2}} = \bar{T}_{k+\frac{1}{2}} (\sigma_k + p_t/p^*)^{-\kappa}.
\]

For the linearized model, \( \bar{p} \) replaces \( p^* \) everywhere the latter occurs in (4.1.6)–(4.1.15).

### 4.2 Horizontal discretization

The MM4 uses a staggered grid defined on a Lambert conformal map. The domain and (portion) of the grid used in the standard model is presented in Fig. 4.2. Both \( p^* u \) and \( p^* v \) are defined at “dot” points, whereas \( p^* T \) and \( p^* \) itself are defined at “cross” points. Values of \( p^* \) at dot points are determined by interpolation. In the following, the nominal grid distance is denoted by \( \Delta x \) and the map scale factors by \( m_{i,j} \) with a superscript to distinguish values for the dot grid (superscript *) from those for the cross grid (superscript x).

Grid points on the MM4 grids are defined using the indices \( i \) and \( j \). For fixed \( j \), larger values of \( i \) indicate a more northerly location, whereas for fixed \( i \), values of \( j \) increase eastward. Both \( i \) and \( j \) are integers with minimum values of 1, and respective maximum values of \( N_i \) and \( N_j \). The cross grid lies within the dot grid, so the cross grid
Fig. 4.2: The domain of the $46 \times 61$, 80 km grid of the MM4 centered over the United States. A portion of the grid structure, denoted by cross ($\times$) and dot ($\cdot$) points is also presented.
Values of $p^*$ at dot points are determined from those at cross points by:

$$p_{i,j}^{**} = \frac{1}{4} \left( p_{i,j}^x + p_{i-1,j}^x + p_{i,j-1}^x + p_{i-1,j-1}^x \right)$$  \hspace{2cm} (4.2.1a)

for the interior $i = 2, \ldots, N_I - 1$, and $j = 2, \ldots, N_J - 1$;

$$p_{i,1}^{**} = \frac{1}{2} (p_{i,1}^x + p_{i-1,1}^x),$$  \hspace{2cm} (4.2.1b)

$$p_{i,N_J}^{**} = \frac{1}{2} (p_{i,N_J-1}^x + p_{i-1,N_J-1}^x),$$  \hspace{2cm} (4.2.1c)

and for $j = 2, \ldots, N_J - 1$;

$$p_{1,j}^{**} = \frac{1}{2} (p_{1,j}^x + p_{1,j-1}^x),$$  \hspace{2cm} (4.2.1d)

$$p_{N_I,j}^{**} = \frac{1}{2} (p_{N_I-1,j}^x + p_{N_I-1,j-1}^x),$$  \hspace{2cm} (4.2.1e)

for the corners

$$p_{1,1}^{**} = p_{1,1}^x,$$  \hspace{2cm} (4.2.1f)

$$p_{1,N_J}^{**} = p_{1,N_J-1}^x,$$  \hspace{2cm} (4.2.1g)

$$p_{N_I,1}^{**} = p_{N_I-1,1}^x,$$  \hspace{2cm} (4.2.1h)

$$p_{N_I,N_J}^{**} = p_{N_I-1,N_J-1}^x.$$  \hspace{2cm} (4.2.1i)

Values along the grid border have been determined consistent with a condition of zero gradients normal to the boundaries (see the discussion surrounding 4.2.4 below).
Consistent with the finite differences in the MM4, \( \zeta \) and \( \delta \) may be determined on the cross grid by

\[
\zeta_{i,j} = \frac{(m_{ij}^*)^2}{2\Delta x} \left[ -u'_{i+1,j} - u'_{i+1,j+1} + u'_{i,j} + u'_{i,j+1} - v'_{i,j+1} + v'_{i+1,j+1} - v'_{i+1,j} - v'_{i,j} \right],
\]

(4.2.2)

\[
\delta_{i,j} = \frac{(m_{ij}^*)^2}{2\Delta x} \left[ -u'_{i+1,j+1} + u'_{i+1,j} + u'_{i,j+1} + v'_{i+1,j} + v'_{i,j+1} - v'_{i+1,j+1} - v'_{i,j} \right],
\]

(4.2.3)

for all cross points \((i = 1, \ldots, N_I - 1, j = 1, \ldots, N_J - 1)\). The primes here indicate values divided by corresponding grid values of \( m^* \). These expressions are equivalent to averaging values in one direction when differencing in the other direction.

The INIT computes the Laplacian of \( h \) on cross points as:

\[
(\nabla^2 h)_{i,j} = \frac{(m_{ij}^*)^2}{(\Delta x)^2} (h_{i+1,j} + h_{i-1,j} + h_{i,j+1} + h_{i,j-1} - 4h_{i,j})
\]

(4.2.4a)

for the interior \( i = 2, \ldots, N_I - 2, j = 2, \ldots, N_J - 2 \). For the border points where \( i = 1 \) or \( N_I - 1 \), or \( j = 1 \) or \( N_J - 1 \), the same equation is used, with the undefined values on the right hand side determined by assuming zero gradient normal to the boundary; i.e., for an augmented cross grid with indices 0, 1, \ldots, \( N_I, N_J \),

\[
h_{0,j} = h_{1,j},
\]

(4.2.4b)

\[
h_{N_I,j} = h_{N_I-1,j},
\]

(4.2.4c)

\[
h_{i,0} = h_{i,1},
\]

(4.2.4d)

\[
h_{i,N_J} = h_{i,N_J-1},
\]

(4.2.4e)
for $i = 1, N_I - 1$, and for the corners

$$h_{0,0} = h_{1,1}, \quad (4.2.4f)$$

$$h_{N_I,0} = h_{N_I-1,1}, \quad (4.2.4g)$$

$$h_{0,N_J} = h_{1,N_J-1}, \quad (4.2.4h)$$

$$h_{N_I,N_J} = h_{N_I-1,N_J-1}. \quad (4.2.4i)$$

The finite difference Laplacian which is consistent with the MM4 is not as defined in (4.2.4a). The consistent expression is also a 5-point Laplacian, but involves different points. The differences are computed as though they were along diagonals on the $i,j$ grid, rather than along lines of constant $i$ or $j$. Specifically, the consistent expression in place of (4.2.4a) should be:

$$\left(\nabla^2 h\right)_{i,j} = \frac{(m \times_{i,j})}{2(\Delta x)^2}(h_{i+1,j+1} + h_{i-1,j+1} + h_{i-1,j-1} + h_{i+1,j-1} - 4h_{i,j}). \quad (4.2.5)$$

The INIT software uses the form (4.2.4) because then available software can be used for inverting the expression; i.e. for determining a field of $h$ given a field of $\nabla^2 h$. This is not the case for the expression (4.2.5).

The discrete Helmholtz equation is

$$m^2_{i,j}(\nabla^2 f)_{i,j} + \lambda f_{i,j} = g_{i,j}, \quad (4.2.6)$$

where $m^2_{i,j}(\nabla^2 f)_{i,j}$ is represented by the right hand side of (4.2.4a), and the field $f$, with suitable boundary conditions, is to be determined from the field $g$. The form of
(4.2.6) presents a problem if available software is to be used for its solution. Generally, such software does not explicitly consider map scale factors that vary in two directions. If \( \lambda = 0 \), the \( m_{i,j} \) can be made implicit by replacing \( g_{i,j} \) by

\[
g'_{i,j} = g_{i,j} m_{i,j}^2,
\]

and solving

\[
\frac{1}{(\Delta x)^2} (f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) - 4f_{i,j} = g'_{i,j}
\]

for \( f \). This procedure is used in INIT to determine \( \Delta \hat{\psi} \) and \( \Delta \hat{\chi} \) from \( \Delta \hat{\xi} \) and \( \Delta \hat{\delta} \), respectively.

When \( \lambda \neq 0 \), the \( m_{i,j} \) cannot be made implicit as in (4.2.8). However, if values of \( m_{i,j} \) do not differ greatly from 1, (4.2.6) can be solved iteratively by first solving (4.2.6) with \( m_{i,j} = 1 \) for all \( i, j \) and then recomputing \( f \) in the same manner, but with \( g_{i,j} \) replaced by

\[
g''_{i,j} = g_{i,j} + (m_{i,j}^2 - 1)m_{i,j}^2 (\nabla^2 f)_{i,j},
\]

with \( m_{i,j}^2 (\nabla^2 f)_{i,j} \) evaluated from the previous iterate of \( f \) using (4.2.4a). The second part of this procedure is then repeated, iteratively. If \( m \) is nowhere very different from 1, the procedure should converge. If \( m > \sqrt{2} \) anywhere, the procedure will likely diverge. This procedure is used in INIT to determine \( \Delta \hat{\xi} \) and \( \Delta \hat{\delta} \) using the discrete forms of (2.4.5) and (2.4.6), respectively.

Values of \( u \) and \( v \) on dot points are determined from \( \psi \) and \( \chi \) on cross points by:

\[
\frac{u_{i,j}}{m_{i,j}} = \frac{1}{2\Delta x} [X_{i,j} + X_{i-1,j} - X_{i,j-1} - X_{i-1,j-1} - \psi_{i,j} + \psi_{i-1,j} - \psi_{i,j-1} + \psi_{i+1,j-1}],
\]

(4.2.10)
\[ \frac{\psi_{i,j}}{m_{i,j}} = \frac{1}{2\Delta x} [x_{i,j} - x_{i-1,j} + x_{i,j-1} - x_{i-1,j-1} + \psi_{i,j} + \psi_{i-1,j} - \psi_{i,j-1} - \psi_{i+1,j-1}], \]

These relations are specifically used to determine interior values of \( \Delta \hat{u} \) and \( \Delta \hat{\vartheta} \) from values of \( \Delta \hat{\psi} \) and \( \Delta \hat{\chi} \).

As a diagnostic, the square root of various horizontal (area) mean squared (denoted rms) values are computed. As an example, the rms value of the \( \hat{h} \) field for the \( k \)th vertical mode is computed on the cross grid as:

\[ |\hat{h}_k| = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left( m_{i,j}^x \right)^{-2} \hat{h}_{i,j,k}^2}{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left( m_{i,j}^x \right)^{-2}}. \]

### 4.3 Horizontal boundary conditions

As discussed in Section 2.2, it is necessary to specify boundary conditions for the changes to the horizontal fields. The boundary conditions used in INIT are not self-consistent. The net result is that \( \Delta u \) and \( \Delta v \) on the dot-grid and \( \Delta p \) and \( \Delta T' \) on the cross-grid all are zero along the boundary. However, the linearized potential vorticity is not constrained along the cross-grid boundary, and the balance conditions are not satisfied there.

Specifically, the boundary conditions which are applied are as follows:

**Condition 1.** The finite-difference forms of (2.4.5) and (2.4.6) are solved with the condition that \( \Delta \hat{z} \) and \( \Delta \hat{\delta} \) vanish along the cross-grid boundary. However, these values are later altered, as described below.
Condition 2. The finite-difference forms of (2.1.17) and (2.1.18) are solved with the condition that $\Delta \hat{\psi}$ and $\Delta \hat{\chi}$ vanish along the cross-grid boundary.

Condition 3. Values of $\Delta \hat{u}$ and $\Delta \hat{v}$ on the dot-grid boundary are set to zero.

Once $\Delta \hat{u}$ and $\Delta \hat{v}$ are determined everywhere, the values of $\Delta \hat{z}$ and $\Delta \hat{\theta}$ should be considered as defined by (4.2.2) and (4.2.3) everywhere. Then they will no longer satisfy Condition 1. Nor will they satisfy the discrete forms of the balance equations (2.4.5) and (2.4.6).

Values of $\Delta \hat{h}$ are computed from those of $\Delta \hat{z}$ consistent with (2.4.7) everywhere except along the boundaries. At the boundaries, values of $\Delta \hat{h}$ are computed consistent with Condition 1 with respect to (2.4.7). Therefore, those values vanish there, but since values of $\Delta \hat{z}$ do not, the linearized potential vorticity is changed along the boundary.
Part 2 consists of three sections. The first, Section 5, is an introduction to the vertical mode software. Section 6 is a mathematical description of the vertical mode problem in terms of vectors of data fields and matrices in place of integral-differential operators. Section 7 is a description of the actual software used to perform the calculations described in Sections 6 and 2.3.

Section 5. Introduction to Vertical Mode Software

The vertical mode software will normally be used to generate various transformation matrices. These include matrices for determining vertical mode amplitudes from $\sigma$-level data, and for determining geopotential from temperature. The software will therefore normally be used in a multi-step job, where this software will be executed in one step, followed by the execution of a program requiring output from the former. The necessary matrices are passed from one program to another via out-of-core, local datasets.

The vertical mode software is stored using CRAY UPDATE with the deck name VMODES in the program library READMP (for Ron Errico, Acid Deposition Modeling Project). It is written in standard FORTRAN-77 with some CRAY extensions. It also uses two NCAR library subroutines: EIGRG1 is used to compute eigenvectors and eigenvalues of a real general matrix; INVMTX is used to invert a real matrix.
This software is designed for use with the ADMP PSU/NCAR Mesoscale Model. For other models, it would have to be modified.
Section 6. Matrix Form of Vertical Discretization of the MM4

The operators needed to describe the time tendency of \( h \) at \( K \)-discrete \( \sigma \)-levels in terms of \( \delta \) at those same levels in the MM4 are most simply determined as the sums and products of several elementary matrices. Those matrices, all of dimension \( K \times K \), are:

- **\( D_1 \)** a diagonal matrix with elements \( d_k = \Delta \sigma_{k + \frac{1}{2}} \)

- **\( D_2 \)** a diagonal matrix with elements \( d_k = \kappa \tilde{T}_{k + \frac{1}{2}} / (\sigma_{k + \frac{1}{2}} + p_T / \bar{p}) \)

- **\( E_1 \)** a matrix whose elements all equal 1

- **\( E_2 \)** a lower-triangular matrix, with elements defined by \( e_{k,l} = 0 \) if \( k < l \), and equal to 1 otherwise.

- **\( E_3 \)** a bi-diagonal matrix with non-zero elements \( e_{k,l} = 1 \) if \( k = l \) or \( k = l - 1 < K \).

- **\( G_1 \)** a bi-diagonal matrix with non-zero elements \( g_{k,l} = \tilde{T}_k \) if \( k = l \) or \( k = l - 1 < K \).

- **\( S_1 \)** a diagonal matrix with elements \( s_k = \sigma_k \)

- **\( S_2 \)** a diagonal matrix with elements \( s_k = \sigma_{k + \frac{1}{2}} \).

In order to describe the vertical mode problem in terms of matrices, it is necessary to describe the data in terms of vectors, as discussed generally in Section 2.3. Here, data for \( \delta \), \( T \), \( \phi \), \( h \), and \( \omega / \bar{p} \) are considered as vectors with ordered components at half-levels \( 1 + \frac{1}{2}, \ldots, K + \frac{1}{2} \). A vector of \( \dot{\sigma} \) data is ordered with respect to full-levels \( 1, \ldots, K \). The additional value of \( \dot{\sigma}_{K + 1} \) is considered only implicitly, since it is always zero-valued (as is \( \dot{\sigma}_1 \)). In this way, all vectors have length \( K \) and all matrices are square.
The hydrostatic matrix \( B \) is used to transform from a vector of \( T \) to one of \( \phi \), as shown in (2.1.25). The appropriate matrix for the MM4 is determined from (4.1.6) as

\[
b_{i,j} = \begin{cases} 
\frac{\Delta \sigma_{j+\frac{1}{2}} d_{j+1}}{\Delta \sigma_{j+\frac{1}{2}} + \Delta \sigma_{j+\frac{1}{2}}} & \text{if } i = j \\
\frac{\Delta \sigma_{j+\frac{1}{2}} d_{j+1}}{\Delta \sigma_{j+\frac{1}{2}} + \Delta \sigma_{j-\frac{1}{2}}} + \frac{\Delta \sigma_{j+\frac{1}{2}} d_j}{\Delta \sigma_{j+\frac{1}{2}} + \Delta \sigma_{j-\frac{1}{2}}} & \text{if } i < j < K \\
d_{K+1} + \frac{\Delta \sigma_{K+\frac{1}{2}} d_K}{\Delta \sigma_{K+\frac{1}{2}} + \Delta \sigma_{K-\frac{1}{2}}} & \text{if } i < K; j = K \\
d_{K+1} & \text{if } i = j = K.
\end{cases}
\] (6.1)

The matrix \( G_2 \) is used to determine the vector of \( \delta \) from the vector of \( \delta \). It is determined from (4.1.4) as

\[
G_2 = S_1 E_1 D_1 - (E_2 - I) D_1.
\] (6.3)

The thermodynamic matrix \( A \) is used to compute the vector of \( \partial T / \partial t \) from the vector of \( \delta \), as in (2.1.24). This \( A \) is most simply determined as the sum

\[
A = A_1 + A_2 + A_3 + A_4,
\] (6.4)

where

\[
A_1 = D_1^{-1} G_1 G_2,
\] (6.5)

\[
A_2 = D_2 (\frac{1}{2} E_3 G_2 - S_2 E_1 D_1),
\] (6.6)

\( A_3 \) is a diagonal matrix with elements \( a_k = -\bar{T}_{k+\frac{1}{2}} \), and

\[
A_4 = -A_3 E_1 D_1.
\] (6.7)

These relations are derived from (4.1.10) and (4.1.12).

The matrix \( r \) is used to compute the vector of \( \partial h / \partial t \) from the vector of \( \delta \). It may be computed as

\[
\tau = R(BA - S_2 D_2 E_1 D_1).
\] (6.8)
Section 7. The Program VMODES

The program VMODES consists of a main program, of the name VMODES, and 6 subroutines. VMODES also calls the NCAR library routines INVMTX (which inverts a matrix) and EIGRG1 (which determines eigenvalues and eigenvectors of a real general matrix).

The program requires specification of only three parameters for most applications. They are specified in a PARAMETER statement. These parameters are:

KX. As in the MM4, this is the number of levels on which the temperature is independently defined (excluding the surface).

PTOP. As in the MM4, this is the pressure at the model top, in centibars.

PSMEAN. A horizontal-mean value of the surface pressure, in centibars. Results should be independent of this value, except in the range of round-off errors. Therefore PSMEAN may be kept equal to 100.

Other parameters are also specified within the program. These include values for the gas constant $R$ and $\kappa$, whose values are given as 287. and 0.287, respectively. A unit number for output of binary data is also specified as IUNIT, with a given value of 69. Values of the three user-specified parameters are copied to another three variables, since the program was originally written using parameters NK, PT, and PS in place of KX, PTOP, and PSMEAN, respectively.
Most of the variable names agree with the matrix names used in Section 6. For example, the matrix $A$, is simply called $A$, and the matrix $E_1$ is called $E1$. Other arrays are:

- HYDROS the hydrostatic matrix $B$.
- HYDROR the matrix inverse of $B$.
- ZMATRX the matrix $Z$ whose columns are the vertical structure functions (normal modes).
- ZRMTRX the matrix inverse of $Z$.
- TAU the matrix $\tau$.
- HBAR a vector containing the set of equivalent depths (times $g$, so the units are in $m^2/s^2$).
- TBARH the specified horizontal-mean temperature at data levels (units of K).
- TBARF the mean temperature at full-levels, interpolated from data levels as described in Section 6.
- SIGMAH the interpolated values of $\sigma$ at half-levels.
- SIGMAF the user-specified values of $\sigma$ at full levels.
- CPFAC is any row of $pE_1\tau^{-1}D_1$, corresponding to the operator in 2.5.1 (all rows are identical).
RTPFAC is the vector whose elements are $RTPFAC = \frac{RT}{k+\frac{1}{2}} \left( \bar{P} + P_T / \sigma_{k+\frac{1}{2}} \right)^{-1}$ which are factors used in the discrete evaluation of $h$ (2.1.26).

Values of TBARH are specified as those of a U.S. Standard atmosphere, as described by Hess (1959). The precise specification is

\begin{align*}
\tilde{T}_{k+\frac{1}{2}} &= T_o \left( \frac{\sigma_{k+\frac{1}{2}} \bar{P} + P_T}{p_o} \right)^{\frac{B_o}{\gamma}} \\
\tilde{Z}_{k+\frac{1}{2}} &= \frac{1}{\gamma} (T_o - \tilde{T}_{k+\frac{1}{2}}),
\end{align*}

\begin{equation}
T_{k+\frac{1}{2}} = \begin{cases} 
\tilde{T}_{k+\frac{1}{2}}, & \text{if } \tilde{Z}_{k+\frac{1}{2}} \leq \tilde{Z}_X \\
T_X, & \text{if } \tilde{Z}_{k+\frac{1}{2}} > \tilde{Z}_X 
\end{cases}
\end{equation}

where

- $\gamma$ is a standard lapse rate parameter $= 0.0065 \text{ K/m}$,
- $T_o$ is a standard surface temperature $= 288.15 \text{ K}$,
- $T_X$ is a standard stratospheric temperature $= 218.15 \text{ K}$,
- $\tilde{Z}_X$ is a standard tropopause height $= 10769 \text{ m}$, and
- $p_o$ is a standard surface pressure $= 101.325 \text{ cb}$.

These values are set in the subroutine TLAPSE.

The normal modes of $\tau$ are normalized such that:

\begin{equation}
1 = \sum_{k=1}^{K} (z_{k,i})^2 \Delta \sigma_{k+\frac{1}{2}},
\end{equation}

44
where this $Z_{k,l}$ is the $K$th component of the $l$th eigenvector, and the component with the largest absolute value is positive valued, for all modes $l$. Note that (7.4) determines a magnitude for the normalization, and the second condition determines an arbitrary sign. The modes are normalized in the subroutine NORMLZ.

Standard printed output from VMODES consists of the single line

\[
\text{VERTICAL MODE PROBLEM SUCCESSFULLY COMPLETED FOR NK=xxx}
\]

where $xxx$ is the number of half-levels. Additional output is obtained by specifying LPRINT as .TRUE. in a data statement. In that case, the subroutines PRINTM and PRINTV are called for the printing of two-dimensional and one-dimensional real variables, respectively. These include all those arrays discussed in Part 2 of this report. All the printed arrays are labeled. The printing is such that the first variable printed is the first in the array. For two dimensional arrays, the second index varies across the page.

Standard binary data are also output in the main program. This consists of a single record containing the variables: $\text{TBARH, SIGMAH, HBAR, CPFAC, RTPFAC, HWEIGH, ZMATRIX, ZRMTRX, TAU, HYDROS, HYDROR, DSIGMA, RGAS, PT, PD, ALPHA1, ALPHA2}$. The first six variables, and $\text{DSIGMA}$, are vectors of length $KX$. Variables $\text{ZMATRIX}$ through $\text{HYDROR}$ are matrices of size $KX \times KX$. The remaining variables are single valued. The variables $\text{HWEIGH}$, $\text{ALPHA1}$ and $\text{ALPHA2}$ are not used in the current version of the software.
Part 3. INITIALIZATION SOFTWARE

Part 3 is divided into 4 sections. The first, Section 8 is an introduction to the initialization software (excluding the program VMODES). Section 9 describes the sequence of job steps required to produce an initialized dataset. The program library decks INITAL and FSUBS are described in Sections 10 and 11 respectively.

Section 8. Introduction to the Initialization Software

The software is particularly designed to produce a balanced set of data from an analyzed set of data, consistent with the MM4. To use it for this purpose is especially simple. However, with some modification, it can be used for other purposes; e.g., it may be used as a diagnostic tool to dynamically analyze model output. Only the first use is specifically described in this report.

The initialization software consists of two CRAY UPDATE program library decks named INITAL and FSUBS together with a common deck named PARAMS. These decks constitute a single program which is used together with the program VMODES and the model program to produce an initialized dataset. The deck INITAL contains the main program and basic transform subroutines (for determining one field from another; e.g. vorticity from velocity). The deck FSUBS contains the subroutines which sequence the operations necessary for determining the modifications to the fields in order to satisfy the balance condition described in Section 2.2. This program reads the file generated by
VMODES, and uses output from the MM4 to determine time tendencies of the dynamic fields. The program library (excluding the model) resides on a volume named READMP.

Without modification to the program, this software produces a data input file for the MM4. This file consists of a single set of data at a single time. Only the fields of $p^* u$, $p^* v$, $p^* T$, $p^* Q$ (although not $Q$ itself) and $p^*$ are altered. Other fields on the input file are copied unaltered from the file which contained the data to be initialized. That original file is left intact, so that either that file or the new initialized file may be used to start the model.
Section 9. Instructions in Job Control

As described in the Introduction, several different programs are used to compute various quantities needed for the initialization procedure. Since the vertical modes are iteration independent, they can be calculated once at the beginning of an initialization procedure, and stored on a file for subsequent use (alternatively, such stored values can be read from an acquired dataset, but the acquiring of an additional dataset would result in significantly slower job execution). Also, the initialization procedure is treated somewhat independently of the model by keeping the software distinct. In particular, the model software is used to compute all nonlinear and any diabatic terms that are relevant to the initialization. If the model is changed, the initialization software need not be changed unless the model data file formats are changed. It is through the exchange of such files that the initialization and model software communicate data between each other. For these reasons, there are three distinct programs that must be executed, in proper order, during a nonlinear initialization procedure.

The order of execution of the three programs is specified by instructions in CRAY Job Control Language (JCL). An outline of the order of these instructions is:

1. Acquire needed datasets, compile the three programs after modifying them using CRAY UPDATE, and create absolute binary load modules of each, so that each program is in a form ready for execution.

2. Execute VMODES, and save necessary output on a local file (i.e., FORTRAN unit).
3. Execute INITAL in order to compute fields which describe a starting iterate for non-linear initialization.

4. Execute the model program, whose output will be used to compute time tendencies (and thereby, nonlinear terms). Input, as initial data, is the output from the last execution of INITAL (i.e., the latest iterate of the data to be initialized).

5. Execute INITAL to produce the next iterate of fields to be initialized, and to output diagnostics. Input as data are the output from the model and the previous iterate of the fields to be initialized.

6. Either return to Step 4 or permanently save and dispose the initialized dataset, depending on the number of iterations completed.

Steps 4 and 5 are in a CRAY JCL loop. It allows repeated execution of a set of instructions. In the case of this initialization software, these instructions are repeated \( J_1 + 1 \) times, where \( J_1 \) is a JCL integer variable set at the beginning of the JCL instructions. The last time through the loop, a new iterate of the data is created, but not permanently saved. Instead, the purpose of this last time through the loop is only to compute diagnostics used to verify that the initialization procedure is functioning as intended.

The SAVE instruction creates a permanent dataset on the CRAY discs having a name which the user specifies. A copy of the same dataset, having an identical name, is also disposed to a more permanent storage device. This latter step is necessary, because space on the CRAY discs is necessarily limited, and no datasets of this size will remain on the discs very long.
Execution of the instructions ABINIT and ABMODL result in running of the program INITAL and the model program, respectively. These are absolute binary load modules, produced as output from execution of a CRAY LDR command. The LDR commands in this case do not result in execution of the programs, since such execution is turned off by the parameter NX.

Before execution of each program, datasets, referenced as FORTRAN units, are appropriately positioned so that data may be read properly by the program. Within the programs VMODES and INITAL, all datasets are rewound after reading or writing.
Section 10. The Program INITAL

Successive steps in the program INITAL are labeled in comments with steps indexed with letters A through J. In some cases, the letters are followed by numbers, for further break down. Step A includes all non-executable statements (parameter, dimension, and data statements, and common block definitions). Steps B-E include all necessary steps to compute $\hat{\xi}$, $\hat{\delta}$, and $\hat{h}$ from an MM4 input or output dataset. One of various routines used to modify fields of $\hat{\xi}$, $\hat{\delta}$, and $\hat{h}$ is called in Step F. In Steps G-K, data suitable for input to the MM4 are modified, consistent with the modifications in Step F. From this, a new input file is created. The order of operations is similar to that presented in Section 3. The precise order of operations appears below.

10.1 Variable names

The variables used in INITAL are discussed in this section. For this purpose, the variables are divided into three sets. The first includes all those defined as FORTRAN parameters. The second includes all variables which may be specified using NAMELIST. The last includes all data arrays. In this software, all integer variables begin with the letters I through N, and real variables begin with A-H or P-Z. An exception is the real variable LAMBDA in subroutine QSUB. All logical variables begin with L.
There are seven user-defined parameters. Three are the grid dimensions IX, JX, KX. These are the same parameters as specified in the MM4. The parameter NIJMAX must be set to the largest of IX and JX. DX is the grid spacing \( \Delta x \) (in meters). These values must be identical to those in the datasets to be initialized. Within the initialization software, IX, JX, KX, and DX are denoted as the respective parameters NI, NJ, NK, and DELX. Values of the former parameters are copied to the latter in a parameter statement. The software will not function properly if KX<4, because the equivalencing used will then result in the over-writing of necessary data.

The user-defined parameter CLAT is the central latitude of the grid (in degrees north), used to compute a coriolis parameter \( f_0 \). CLAT must be nonzero. Although a constant value of \( f_0 \) is used on the grid, if the initialization is nonlinear, any variations in \( f \) are implicitly considered in nonlinear terms.

The user-defined parameter DT is the time step \( \Delta t \) (in seconds) used as in the denominator of (2.4.8) to compute the time tendencies of the dynamic fields. It must be specified identically in the particular model version used with the initialization scheme (which is the \( \Delta t \) which appears in the numerator of (2.4.8)). This DT need not be the same as that used to produce any subsequent forecasts from the initialized data.

The parameters NI1 and NJ1 are determined as one less than the respective values of NI and NJ. They are the dimensions of the cross grid, which is one row and column

52
smaller than the dot grid. Remaining parameters are data pointers, described in Section 10.1.3

The parameter ITERT1 is the number of iterations used to solve (4.2.6) using (4.2.9). For most cases, a value of 2 is sufficient.

10.1.2 Namelist variables

Those variables which need to vary with iteration number in an initialization procedure are specified in the namelist LINIT (List Init). These variables include unit numbers for each of the types of datasets which are read or written to by the initialization software, as well as the variables NMODES and IFUNCT.

The number of vertical modes to be initialized is specified by the variable NMODES. An acceptable value is one of the integers $1, \ldots, KX$. The first NMODES vertical modes are initialized, chosen from the set of modes ordered from largest equivalent depth to smallest. For example, NMODES=3 indicates that only the external and first two internal modes are initialized (modes with 0, 1, and 2 zero crossings, respectively).

The program INITAL is designed to perform several different functions. The function performed depends on the value of IFUNCT:

If IFUNCT=0, then no fields are modified. Instead, the software is used to compute $\hat{\zeta}$, $\hat{\delta}$, and $\hat{h}$, write those fields to a file, and compute diagnostics. This function is
appropriate if the starting iterate for a nonlinear initialization is to be unmodified data.

If IFUNCT=1, then a linearly initialized dataset is created. Only those vertical modes specified by the variable NMODES are initialized. The divergence corresponding to those modes is removed (actually reduced an order of magnitude, because the executed finite difference operations are inexact). The rotational and pseudo-geopotential fields are geostrophically balanced, with the changes in both fields such as to preserve the linearized potential vorticity of the original data. A file of the linearly initialized \( \hat{\zeta}, \hat{\delta}, \) and \( \hat{h} \) is written as for the case IFUNCT=1.

If IFUNCT=2, one non-starting iterate of the nonlinear initialization procedure is executed. In this case, input data is read from an MM4 output file used to compute time tendencies of the fields, and a file of \( \zeta, \delta, \) and \( h \) is read.

There are 6 unit numbers that must be specified. These are:

IUNIT1 is the unit containing a file of map scale factors and terrain for the grid corresponding to the data to be initialized. This file must be in an MM4 input file format. Default value is 61.

IUNIT2 is the unit containing a file of data from which \( \zeta, \delta, \) and \( h \) are to be computed. This file must be in the format of either an input or output file for the MM4. Default value is 20.

IUNIT3 is the unit containing values of \( \hat{\zeta}, \hat{\delta}, \) and \( \hat{h} \) determined from a previous execution of INITAL. The file format is described in Section 10.3. This file is read or written in the
subroutine F1SUB. Default value is 68.

IUNIT4 is the unit containing the data to which modifications are to be added. This file must be in an MM4 input file format. Default value is 10.

IUNIT5 is the unit to which the data for the new iterate is written. This file must be in an MM4 input file format. Default value is 10.

IUNIT6 is the unit containing the output from the program VMODES. It contains transform arrays for the hydrostatic equation and vertical modes. Its format was described in Section 7. Default value is 69.

10.1.3 Variable arrays

All the data are stored in one of 6 arrays. These are:

- VERTS, with dimension NVERTS. This includes all the vertical mode information as output by VMODES. This includes all the Common Block CVERT, because it is equivalenced to the variables in that block, beginning with the first word.

- RMDOT, with dimensions NI,NJ. These are the map factors for the dot grid.

- RMCRS, with dimensions NI,NJ. These are the map factors for the cross grid. The last row and column of this array do not contain any information.
PHIS, with dimensions NI,NJ. These are the terrain heights (in meters) for the cross grid.

PS1, with dimensions NI,NJ,4. This array is for storage of 4 pressure fields, each corresponding to a different value of the last index. The first two dimensions are the grid indices, as described in Section 4.2. This array is equivalenced to the array PS2 with dimensions NIJ,4, where NIJ=NI*NJ. By means of this equivalencing, the pressure fields can be referenced using 2 or 3 dimensioned arrays, with the last index denoting a specific field. This array is described further below.

DAT1, with dimensions NI,NJ,NK,9. This array is for storage of all the three dimensional data fields used in the initialization procedure. Values of the last index denote a particular field, as described below. The first three dimensions are the grid indices, as described in Section 4.2. This array is equivalenced to the array DAT2 with dimensions NIJK,9, where NIJK=NI*NJ*NK. By means of this equivalencing, the various three-dimensional fields can be referenced using 2 or 4 dimensional arrays, with the last index denoting a specific field.

Specific fields stored in the arrays PS1 (or PS2) or DAT1 (or DAT2) are referenced by means of pointers. These pointers are parameters which indicate a value for the last index of the appropriate array. By means of these pointers, it is easy to store different fields in the same location by assigning the same value to the pointer. The same could have been achieved by suitably using common blocks and equivalence statements. This effective equivalencing has been used to reduce the total core storage requirements for the initialization software.
All pointer names begin with the letter M. The naming convention is described in comments within the program. Here, the pointers are described as they correspond to the mathematical symbols used in Part 1 of this report. The pointers and their corresponding values and symbols are presented in Table 10.1.

An example of the use of a pointer is as follows: Suppose the field of $\zeta^N$ is to be referenced. The first component of that field (i.e., the $i = 1, j = 1, k = 1$ component) may be referenced either as DAT1(1,1,1,MZHN) or DAT2(1,MZHN). The letters M,Z,H,N respectively indicate: 1. this parameter is a pointer; (2) for the zeta (i.e., $\zeta$) field; (3) in terms of vertical mode amplitudes, denoted by a hat ($\hat{\cdot}$); 4. for a new iterate, as opposed to a former iterate. Actually, at some times during program execution, this will instead point to the first value of the $\Delta u$ field or some other field, since this and other fields are stored in the same location, although at different times. It is necessary for the user to examine the program itself to determine which field has last been referenced for a particular location at a particular time.

The equivalencing is summarized in Table 10.2. Note that in most instances fields are ordered as $u,v,T$ or $\zeta,\delta,h$ or a corresponding set, such as time tendencies or changes of those fields.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pointer</th>
<th>Value</th>
<th>Calculated from</th>
<th>Using equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>((p^*u)^O)</td>
<td>MPUO</td>
<td>1</td>
<td>read last iterate</td>
<td>IUNIT4</td>
</tr>
<tr>
<td>((p^*v)^O)</td>
<td>MPVO</td>
<td>2</td>
<td>read last iterate</td>
<td>IUNIT4</td>
</tr>
<tr>
<td>((p^*T)^O)</td>
<td>MPTO</td>
<td>3</td>
<td>read last iterate</td>
<td>IUNIT4</td>
</tr>
<tr>
<td>((p^*u)^N)</td>
<td>MPUN</td>
<td>1</td>
<td>read model forecast</td>
<td>IUNIT2</td>
</tr>
<tr>
<td>((p^*v)^N)</td>
<td>MPVN</td>
<td>2</td>
<td>read model forecast</td>
<td>IUNIT2</td>
</tr>
<tr>
<td>((p^*T)^N)</td>
<td>MPTN</td>
<td>3</td>
<td>read model forecast</td>
<td>IUNIT2</td>
</tr>
<tr>
<td>(u^O)</td>
<td>MUN</td>
<td>1</td>
<td>((p^*u)^O, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(v^O)</td>
<td>MVN</td>
<td>2</td>
<td>((p^*v)^O, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(T^O)</td>
<td>MTO</td>
<td>3</td>
<td>((p^*T)^O, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(u^N)</td>
<td>MUN</td>
<td>1</td>
<td>((p^*u)^N, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(v^N)</td>
<td>MVN</td>
<td>2</td>
<td>((p^*v)^N, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(T^N)</td>
<td>MTN</td>
<td>3</td>
<td>((p^*T)^N, p^{**})</td>
<td>division</td>
</tr>
<tr>
<td>(\dot{\zeta}^O)</td>
<td>MZHO</td>
<td>1</td>
<td>read or (\zeta^O, \Delta \zeta^N)</td>
<td>IUNIT3</td>
</tr>
<tr>
<td>(\dot{\delta}^O)</td>
<td>MDHO</td>
<td>2</td>
<td>read or (\delta^O, \Delta \delta^N)</td>
<td>IUNIT3</td>
</tr>
<tr>
<td>(\dot{h}^O)</td>
<td>MGHO</td>
<td>3</td>
<td>read or (\dot{h}^O, \Delta \dot{h})</td>
<td>IUNIT3</td>
</tr>
<tr>
<td>(\dot{\zeta}^N)</td>
<td>MZN</td>
<td>1</td>
<td>(u^N/m, v^N/m)</td>
<td>4.2.2</td>
</tr>
<tr>
<td>(\dot{\delta}^N)</td>
<td>MDN</td>
<td>2</td>
<td>(u^N/m, v^N/m)</td>
<td>4.2.3</td>
</tr>
<tr>
<td>(\dot{h}^N)</td>
<td>MGNP</td>
<td>3</td>
<td>(h^N)</td>
<td>equals</td>
</tr>
<tr>
<td>(\zeta^N)</td>
<td>MZHN</td>
<td>4</td>
<td>(\zeta^N)</td>
<td>2.3.10</td>
</tr>
<tr>
<td>(\delta^N)</td>
<td>MDHN</td>
<td>5</td>
<td>(\delta^N)</td>
<td>2.3.10</td>
</tr>
<tr>
<td>(\dot{h})</td>
<td>MGHN</td>
<td>6</td>
<td>(h^N, \dot{h}^O)</td>
<td>2.4.8</td>
</tr>
<tr>
<td>(\Delta u)</td>
<td>MDU</td>
<td>4</td>
<td>(\Delta u/m, m)</td>
<td>multiplication</td>
</tr>
<tr>
<td>(\Delta v)</td>
<td>MDV</td>
<td>5</td>
<td>(\Delta v/m, m)</td>
<td>multiplication</td>
</tr>
</tbody>
</table>

58
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pointer</th>
<th>Value</th>
<th>Calculated from</th>
<th>Using equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta \hat{\psi}$</td>
<td>MDSH</td>
<td>4</td>
<td>$\Delta \xi^O$</td>
<td>4.2.6</td>
</tr>
<tr>
<td>$\Delta \hat{\chi}$</td>
<td>MDCH</td>
<td>5</td>
<td>$\Delta \delta^O$</td>
<td>4.2.6</td>
</tr>
<tr>
<td>$\Delta \xi^N$</td>
<td>MDZHN</td>
<td>4</td>
<td>$\Delta \hat{u}/m$, $\Delta \hat{v}/m$</td>
<td>4.2.2</td>
</tr>
<tr>
<td>$\Delta \delta^N$</td>
<td>MDDHN</td>
<td>5</td>
<td>$\Delta \hat{u}/m$, $\Delta \hat{v}/m$</td>
<td>4.2.3</td>
</tr>
<tr>
<td>$\Delta u/m$</td>
<td>MDUM</td>
<td>4</td>
<td>$\Delta \hat{u}/m$</td>
<td>2.3.11</td>
</tr>
<tr>
<td>$\Delta v/m$</td>
<td>MDVM</td>
<td>5</td>
<td>$\Delta \hat{v}/m$</td>
<td>2.3.11</td>
</tr>
<tr>
<td>$u^N/m$</td>
<td>MUMN</td>
<td>4</td>
<td>$u^N$, $m$</td>
<td>division</td>
</tr>
<tr>
<td>$v^N/m$</td>
<td>MVMN</td>
<td>5</td>
<td>$v^N$, $m$</td>
<td>division</td>
</tr>
<tr>
<td>$\xi^N$</td>
<td>MGN</td>
<td>6</td>
<td>$T^N$, $p^*$</td>
<td>2.1.26</td>
</tr>
<tr>
<td>$\Delta h$</td>
<td>MDG</td>
<td>6</td>
<td>$\Delta \hat{h}$</td>
<td>2.3.11</td>
</tr>
<tr>
<td>$\Delta \xi^O$</td>
<td>MDZHO</td>
<td>7</td>
<td>$\partial \xi/\partial t$, $\partial \nabla^2 \hat{h}/\partial t$</td>
<td>2.4.6</td>
</tr>
<tr>
<td>$\Delta \delta^O$</td>
<td>MDDHO</td>
<td>8</td>
<td>$\Delta \hat{\psi}$, $\Delta \hat{\chi}$</td>
<td>4.2.10</td>
</tr>
<tr>
<td>$\Delta \hat{u}/m$</td>
<td>MDUHM</td>
<td>7</td>
<td>$\Delta \hat{\psi}$, $\Delta \hat{\chi}$</td>
<td>4.2.11</td>
</tr>
<tr>
<td>$\Delta \hat{v}/m$</td>
<td>MDVHM</td>
<td>8</td>
<td>$\Delta \hat{\psi}$, $\Delta \hat{\chi}$</td>
<td>4.2.11</td>
</tr>
<tr>
<td>$\Delta \hat{h}$</td>
<td>MDGH</td>
<td>9</td>
<td>$\Delta \xi^N$</td>
<td>2.4.7</td>
</tr>
<tr>
<td>$\partial \nabla^2 \hat{h}/\partial t$</td>
<td>METH</td>
<td>9</td>
<td>$\partial \hat{h}/\partial t$</td>
<td>4.2.4</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>MDT</td>
<td>9</td>
<td>$\Delta h$, $\Delta p$</td>
<td>2.5.2</td>
</tr>
<tr>
<td>$p^*q$</td>
<td>MQ</td>
<td>7</td>
<td>read</td>
<td>IUNIT4</td>
</tr>
<tr>
<td>$f$</td>
<td>MF</td>
<td>8</td>
<td>read</td>
<td>IUNIT2</td>
</tr>
</tbody>
</table>

Table 10.1. List of variable symbols and pointer names, values of pointers, and references for how the variables are calculated. The description of how variables are calculated is incomplete. It applies primarily to a non-starting iteration for nonlinear initialization. Also see Sections 3, 10.2, and 11.3.
<table>
<thead>
<tr>
<th>Pointer</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(p^*u)^O$ $(p^*u)^N$ $u^O$ $u^N$ $\dot{\xi}^O$ $\xi^N$</td>
</tr>
<tr>
<td>2</td>
<td>$(p^*v)^O$ $(p^*v)^N$ $v^O$ $v^N$ $\dot{\delta}^O$ $\delta^N$</td>
</tr>
<tr>
<td>3</td>
<td>$(p^*T)^O$ $(p^*T)^N$ $T^O$ $T^N$ $\dot{h}^O$ $h^{N'}$</td>
</tr>
<tr>
<td>4</td>
<td>$\dot{\xi}^N$ $\partial \xi / \partial t$ $\Delta u$ $\Delta \psi$ $\Delta \xi^N$ $\Delta u/m$ $u^N/m$</td>
</tr>
<tr>
<td>5</td>
<td>$\dot{\delta}^N$ $\partial \delta / \partial t$ $\Delta v$ $\Delta \chi$ $\Delta \delta^N$ $\Delta v/m$ $v^N/m$</td>
</tr>
<tr>
<td>6</td>
<td>$\dot{h}^N$ $\partial \dot{h} / \partial t$ $\Delta h$ $h^N$</td>
</tr>
<tr>
<td>7</td>
<td>$\Delta \xi$ $\Delta u/m$ $p^*q$</td>
</tr>
<tr>
<td>8</td>
<td>$\Delta \delta$ $\Delta v/m$ $f$</td>
</tr>
<tr>
<td>9</td>
<td>$\Delta \dot{h}$ $\Delta T$ $\partial \nabla^2 h / \partial t$</td>
</tr>
</tbody>
</table>

Table 10.2. Equivalencing of fields. Fields indicated in each row share identical memory locations in the initialization software.
10.2 Program list

In the following, Steps B-K, excluding Step F, are described as they appear in the program INITAL. Step A has been described in Section 10.1 of this report. Discussion of Step F is deferred until Section 11.

B. Set some constants.

B0. Set coriolis parameter $f_o = 2\pi \sin(\text{CLAT})/86400$ sec.

B1. Read namelist parameters from unit 5 (input) and write list to unit 6 (output).

B2. Read binary data output from the Program VMODES.

B3. Read $m^x$, $m^*$, and $\phi_*$ from an MM4 input file (file must be for the same domain and grid as the data file to be initialized, so that same file may be used).

C. Read data. This data may be either an MM4 input or ouput file. If this is the first (starting) iteration for a standard initialization procedure (signified by a value of IFUNCT $\leq 1$), this will be an input file, read in the subroutine RFIN (Read File In). If it is a nonlinear iteration (signified by a value of IFUNCT $> 1$), this will be an output file, which will be used to compute time-tendencies, in accordance with (2.4.8). In this latter case, the data is read in the subroutine RFOUT (Read File Out).

D. Determine $\zeta$, $\delta$, and $h$.

D0. Determine $p^*$ on dot grid from values on cross grid using the 4-point interpolation defined in (4.2.1), in subroutine P1P2.
D1. Remove the factor of $p^*$ from the data read in flux form as $p^*u$, $p^*v$, $p^*T$, and $p^*Q$ by simple division by $p^*$ in the subroutine DIVIDF.

D2. Divide values of $u$ and $v$ at each grid point by values of the map factors at those same points, in preparation for the computation of $\zeta$ and $\delta$. This step reduces the amount of computation in Step D4, and is performed in the routine DIVIDF.

D3. Determine $h$ from $T$ and $p^*$. If LVIRT=.TRUE., $T$ is first replaced by the virtual temperature in the subroutine VIRTT. Values of $h$ are calculated in the subroutine GEOPT using the matrix forms of (2.1.25) and (2.1.26).

D4. Values of $\zeta$ and $\delta$ are computed from those of $u/m$ and $v/m$ using (4.2.2) and (4.2.3) in the subroutines VORT and DIVG, respectively.

D5. Values of $h$ are copied to a new storage location in order to maintain some symmetry in the way variables are stored (i.e., so that later-computed, similar-type variables are stored in adjacent locations).

E. Compute $\hat{\zeta}$, $\hat{\delta}$, $\hat{h}$ from $\zeta$, $\delta$ and $h$ using (2.3.10) in the subroutine TRANSV (Transform Vertical).

F. Modify $\hat{\zeta}$, $\hat{\delta}$, $\hat{h}$.

F1. If IFUNCT=0, no modification is done. Fields of $\hat{\zeta}$, $\hat{\delta}$, $\hat{h}$ are output on IUNIT3, in subroutine F1SUB. Some diagnostics are printed. Some files are repositioned (rewound). Execution terminates with a STOP.
F2. If IFUNCT=1, field modification consistent with a linear initialization is determined in subroutine F2SUB.

F3. If IFUNCT=2, field modification consistent with a non-starting iterate of a non-linear balance condition is determined in subroutine F3SUB.

G. Transform from $\Delta \hat{u}/m$, $\Delta \hat{v}/m$, $\Delta \hat{h}$ to $\Delta u/m$, $\Delta v/m$, $\Delta h$ using the inverse transform (2.3.11) applied in Subroutine TRANSV.

H. Determine $\Delta u$, $\Delta v$, $\Delta T$, and $\Delta p$.

H1. Remove factor $m$ from values of $\Delta u/m$ and $\Delta v/m$ by simple multiplication in subroutine MULTPF.

H2. Compute $\Delta p$ from $\Delta h$ using (2.5.1) in subroutine PFROMH ($p^* \text{ From } h$).

H3. Compute $\Delta p$ on dot grid from values on cross grid using 4-point interpolation (4.2.1) in subroutine P1P2.

H4. Compute $\Delta T$ from $\Delta h$ and $\Delta p$ using (2.5.2)

H5. Zero elements of $\Delta T$ and $\Delta p$ denoted by indices $i=NI$ or $j=NJ$. These values are extraneous to the cross grid, but are present because both cross and dot grids have been dimensioned alike, as discussed in Section 10.1.

I. Read data $p^*u$, $p^*v$, $p^*T$, $p^*Q$, and $p^*$ produced by previous iteration of the initialization procedure. This data is on a dataset in the format of an MM4 input file, read in subroutine RFIN.
J. Modify data to compute new iterate $p^* u$, $p^* v$, $p^* T$, $p^* Q$, $p^*$ from old iterate and the modifications $\Delta u$, $\Delta v$, $\Delta T$, and $\Delta p$.

- **J1.** Remove the factor of $p^*$ from the previous iterate of data (Step I), as in Step D1.

- **J2.** Add $\Delta p$ to old iterate of $p^*$ to determine new iterate $p^*$, computed in subroutine ADDV (Add Vector).

- **J3.** If LVIRT=.TRUE., replace values of $\Delta T$ (in terms of virtual temperature) with those of $\Delta T$ (in terms of true temperature) in the subroutine (entry) VIRTTR.

- **J4.** Add modifications of $u$, $v$, $T$ fields to old iterates to form new iterates, similarly to step J2.

- **J5.** Compute new $\phi$ field, consistent with new $T$ field in subroutine GEOPT2.

- **J6.** Multiply values of new fields of $u$, $v$, $T$, $Q$, $\phi$ by corresponding values of new field of $p^*$, in subroutine MULTPF.

K. Output new iterate of initialized fields in the format of an MM4 data input file in subroutine entry WFIN. Then print that the initialization program has terminated successfully, in the specific sense that the final STOP instruction has been reached.
There are 4 file formats referenced in the initialization software. One is the file of vertical mode information, described in Section 7. Another is a file of \( \zeta, \delta, \hat{h} \). The remaining two are data input and output files of the MM4.

The file of \( \zeta, \delta, \hat{h} \) has NK sets of three records. Each of the three records contains respective values of \( \zeta, \delta, \hat{h} \) for a single equivalent depth. The NK sets are ordered from the largest to smallest equivalent depths. The first word on each record is the integer IC, which is an iteration counter. For any file of this format, all records will have the same value of IC. Following the word IC, are the fields, dumped in the form of an array A(NI,NJ).

The MM4 input files begin with a 1 record header, which contains a single integer word describing the date and time of the data. This is followed by \( 17 \times NJ \) records of data and terminated by an end-of-file. Each data record contains data for a single field for fixed J but all values of I=1,..,NI and, for multilevel fields, all values of K=1,..,NK. For each J=1,..,NJ, the order of the fields is: \( p^*u, p^*v, p^*\phi, p^*Q, p^*T, p^*x, f, m^x, m^* \), latitudes of cross grid, longitudes of cross grid, latitudes of dot grid, longitudes of dot grid, surface T field, \( \phi_s \), land use, snow cover. Each of the fields for a single J appear in the order presented, followed by the ordered fields for the next value of J.

The only records to be read from MM4 output files are records of \( p^*u, p^*v, p^*T, p^*Q, \) and \( p^* \). These follow each other in that order, except in some instances, other records lie in between. These others must be skipped so that the data is read properly. The first record is a header which sufficiently explains the file format so that the software can properly and
automatically skip appropriate records. Data on records are arranged as for input files: each record for a single value of \( J \), with all fields for a single value of \( J \) appearing before those for the next value.

After records of \( p^* T, p^* Q, \) and \( p^* \) are read, extraneous elements of the arrays are set to zero (see the discussion of step H5 in Section 10.2). If this is not done, those elements have indefinite values and should not be used in any calculations. Although such calculated values are necessarily meaningless, portions of the software are simplified by not distinguishing between meaningful and extraneous elements of the arrays. In particular, the arrays can then be treated as single-indexed vectors for some calculations without affecting the meaningful results.
Section 11. The FSUB and QSUB Routines

The deck FSUBS contains all the subroutines for determining modifications to the $\zeta$, $\delta$, and $\hat{h}$ fields given the corresponding fields of the previous iteration. There are 4 subroutines in this deck. These are F1SUB, F2SUB, F3SUB, and QSUB. Each is discussed individually below.

11.1 Subroutine F1SUB

Subroutine F1SUB itself results in no modifications to the fields. It simply writes three 3-dimensional fields to a FORTRAN unit (a disk) for temporary storage. The structure of this file is described in Section 10.3. After writing, the file is repositioned at its beginning. The choice of fields is designated by subroutine arguments upon CALLing. Normally, these are the fields $\zeta^N$, $\delta^N$, and $\hat{h}^N$. If the variable IC (normally the iteration counter ICOUNT) equals 0, diagnostics are also output, as described in Sections 12.7 and 15. This output is labeled as that describing a starting iterate with no data modification.

Subroutine F1SUB also contains the entry F1SUBR. This routine reads a file previously written in the routine entry F1SUB (the character R denotes a reverse (or inverse) function in this notation). After reading, the file is repositioned at its beginning. The fields read will normally be $\zeta^O$, $\delta^O$, and $\hat{h}^O$. 
11.2 Subroutine F2SUB

Subroutine F2SUB sequences the operations required to compute $\Delta \xi$, $\Delta \delta$, $\Delta \hat{h}$, $\Delta \hat{u}$, and $\Delta \hat{v}$ from $\zeta^N$, $\delta^N$, and $\hat{h}^N$ corresponding to a linear initialization (described in Section 10.1.2). The sequence of operations is described below using labels appearing identically as FORTRAN comments within the subroutine itself.

F2.0 Compute $f_o$ as in the main program.

F2.1 Copy $\zeta^N$, $\delta^N$, $\hat{h}^N$ to $\zeta^O$, $\delta^O$, $\hat{h}^O$, respectively, using the routine COPYAB. Also set ICOUNT=0.

F2.2 Compute $\nabla^2 \hat{h}$ using (4.2.4) in the subroutine LAPLAC.

F2.3 Compute the linearized tendencies $\partial \zeta / \partial t = -f_o \delta$ and $\partial \delta / \partial t = f_o \zeta - \nabla^2 \hat{h}$

F2.4 In the subroutine DIAGNT, compute rms values of the fields of $\zeta^N$, $\delta^N$, $\hat{h}^N$ (see step F2.1) and the fields computed in step F2.3. Ignore the fields designated DG/DT and D(Z-(F/H)G)/DT. The rms values are calculated as in (4.2.12).

F2.5 Determine $\Delta \hat{\psi}$ using the discrete form of $xx$ and $\Delta \hat{v} = -\delta^N$. These values will actually be recomputed, so that the final values are compatible with the $\Delta \hat{u}$ and $\Delta \hat{v}$ fields (in general the finite difference operations used for the various computations are not strictly compatible).

F2.6 Compute $\Delta \hat{\psi}$ from $\Delta \zeta$ and compute $\Delta \hat{x}$ from $\Delta \hat{v}$ using (4.2.8) in subroutine QSUB. Prior to the call to QSUB, overwrite the cross-grid boundary values of $\Delta \zeta$ and $\Delta \hat{v}$ by
the desired boundary values of $\Delta \psi$ and $\Delta \chi$, respectively. Those boundary values are specified as zero in the routine BOUNDF.

F2.7 Compute $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$ from $\Delta \hat{\psi}$ and $\Delta \hat{\chi}$ using (4.2.10) and (4.2.11), respectively, in subroutine UVSUB.

F2.8 Re-compute $\Delta \hat{\zeta}$ and $\Delta \hat{\delta}$ from $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$. The new fields of $\Delta \hat{\zeta}$ and $\Delta \hat{\delta}$ will generally be smoother than the earlier values computed in Step F2.5. This is a property of the finite difference approximations used (i.e., they use average values in directions perpendicular to the direction differences are taken, and therefore act to smooth the fields). Set boundary values of $\Delta \hat{\zeta}$ and $\Delta \hat{\delta}$ equal to zero in BOUNDF. These boundary values are no longer correctly related to $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$.

F2.9 Compute $\Delta \hat{h}$ corresponding to $\Delta \hat{\zeta}$ such that the linearized potential vorticity is unchanged, using the discrete analog of (2.4.7).

2.10 Add $\zeta^O$, $\delta^O$, $\hat{h}^O$ to the corresponding values of and $\Delta \hat{\zeta}$, $\Delta \hat{\delta}$, $\Delta \hat{h}$ to obtain $\zeta^N$, $\delta^N$, $\hat{h}^N$, as in (2.2.10) using subroutine ADDV.

F2.11 Write $\zeta^N$, $\delta^N$, $\hat{h}^N$ to a file using subroutine F1SUB.

11.3 Subroutine F3SUB

Subroutine F3SUB sequences the operations required to compute $\Delta \hat{\zeta}$, $\Delta \hat{\delta}$, $\Delta \hat{h}$, $\Delta \hat{u}$, and $\Delta \hat{v}$ from $\zeta^N$, $\delta^N$, $\hat{h}^N$, $\zeta^O$, $\delta^O$, and $\hat{h}^O$ corresponding to one iteration of a nonlin-
ear initialization. The sequence of operations is described below using labels appearing identically as FORTRAN comments within the subroutine itself.

F3.0 Compute $f_0$ as in the main program.

F3.1 Read old values of $\xi^O$, $\delta^O$, $h^O$ from a file using entry F1SUBR in subroutine F1SUB. Also increment the iteration counter ICOUNT by 1.

F3.2 Compute $\partial\xi/\partial t$, $\partial\delta/\partial t$, $\partial h/\partial t$, using (2.4.8), computed in subroutine DIFFV and DIVIDV.

F3.3 Compute $\partial^2 v^2 \hat{h}/\partial t$ using (4.2.4) in the subroutine LAPLAC.

F3.4 Compute rms values of the fields of $\xi^O$, $\delta^O$, $h^O$ and $\partial\xi/\partial t$, $\partial\delta/\partial t$, $\partial h/\partial t$ using (4.2.12) in the subroutine DIAGNT.

F3.5 Determine $\Delta \xi$ and $\Delta \delta$ using (4.2.2) and (4.2.3), respectively, in routine QSUB. These values will actually be recomputed, so that the final values are compatible with the $\Delta \hat{u}$ and $\Delta \hat{v}$ fields (in general the finite difference operations used for the various computations are not strictly compatible). Boundary values for $\Delta \xi$ and $\Delta \delta$ are specified as zero in routine BOUNDF, prior to the calls to QSUB.

F3.6 Compute $\Delta \hat{\psi}$ from $\Delta \xi$ and compute $\Delta \hat{\chi}$ from $\Delta \delta$ using (4.2.8) in subroutine QSUB. Prior to the call to QSUB, overwrite the cross-grid boundary values of $\Delta \xi$ and $\Delta \delta$ by the desired boundary values of $\Delta \hat{\psi}$ and $\Delta \hat{\chi}$, respectively. Those boundary values are specified as zero in the routine BOUNDF.
F3.7 Compute $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$ from $\Delta \hat{\psi}$ and $\Delta \hat{\chi}$ using (4.2.10) and (4.2.11), respectively, in subroutine UVSUB.

F3.8 Re-compute $\Delta \zeta$ and $\Delta \hat{\delta}$ from $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$ in routines VORT and DIVG. The new fields of $\Delta \zeta$ and $\Delta \hat{\delta}$ will generally be smoother than the earlier values computed in Step F3.5. This is a property of the finite difference approximations used (i.e., they use average values in directions perpendicular to the direction differences are taken, and therefore act to smooth the fields). Set boundary values of $\Delta \zeta$ and $\Delta \hat{\delta}$ equal to zero in BOUNDF. These boundary values are no longer correctly related to $\Delta \hat{u}/m$ and $\Delta \hat{v}/m$.

F3.9 Compute $\Delta \hat{h}$ corresponding to $\Delta \hat{\zeta}$ such that the linearized potential vorticity is unchanged, using the discrete analog of (2.4.7).

F3.10 Add $\xi^O$, $\delta^O$, $h^O$ to the corresponding values of and $\Delta \zeta$, $\Delta \hat{\delta}$, $\Delta \hat{h}$ to obtain $\xi^N$, $\delta^N$, $h^N$, as in (2.2.10) using subroutine ADDV.

F3.11 Write $\xi^N$, $\delta^N$, $h^N$ to a file using subroutine FISUB.

11.4 Subroutine QSUB

Subroutine QSUB sequences the operations specifically necessary to solve a discrete Helmholtz equation with variable map factors. The procedure is iterative, as described by (4.2.4)-(4.2.9). In particular, it is used to determine fields of $\Delta \zeta$ and $\Delta \hat{\delta}$. The routine required to solve the Poisson equations for determining $\psi$ and $\chi$ from $\zeta$ and $\delta$ is called from
QSUB also. The solution of a Poisson equation is indicated by the value of the argument IFUNCT equal to -1. In either case, the primary computation is performed in the NCAR library subroutine HWSCRT.

If difficulties are detected in execution of HWSCRT, error messages are output to the printer by QSUB. A value of the argument PERTRB is returned upon completion of HWSCRT. When the solution of a Poisson equation is called for, the value of PERTRB should be small compared to the magnitude of the field values input to HWSCRT (e.g., values of $\psi$ from which $\psi$ is to be determined). The value of PERTRB is printed only if it is nonzero. A value of the argument IERROR is also returned upon completion of HWSCRT. Only nonzero values are printed. They indicate invalid parameters have been input to HWSCRT from the calling program QSUB. In this case, the solutions has not been attempted by HWSCRT, except when IERROR=6, which indicates that $\lambda > 0$, for which a solution may not exist.

The library subroutine HWSCRT permits specification of several different kinds of boundary conditions. In QSUB, that boundary condition is chosen to be a Dirichlet condition at all boundaries (indicated by the two arguments equal to 1 in the calls to HWSCRT). For this case, the boundary values of the input field (e.g., $\zeta$) are over-written by the prescribed boundary values (e.g., $\psi$), as prescribed in the routine which called QSUB. The array BD contains no information in this case, since it is relevant to Neumann boundary conditions (i.e., normal derivative conditions) only.

The subroutine HWSCRT uses a work array named WORK. In most cases, WORK
is over dimensioned, although it is possible but rare that it is under dimensioned. See a description of the library subroutine HWSCRT for details.
The format of a CRAY job deck which may be used for executing a nonlinear initialization procedure is presented in 5 parts in Figs. 12.1-12.5. These respective parts are: a CRAY JCL file; an UPDATE file for VMODES; an UPDATE file for INITAL and FSUBS; an UPDATE file for the MM4-V3; and 3 files of namelist input to the various programs. Each part is described separately below, although each part may need modification for a particular use. Places where a user must supply values are indicated by lower case letters. This section is written so that a user can read it apart from other sections and understand the software sufficiently to execute a nonlinear initialization of an MM4 input dataset. Execution of other tasks may require reading of other sections of this report.

12.1 The JCL file

The CRAY JCL file for executing a nonlinear initialization is presented in Fig. 12.1a. The user must supply a CRAY job instruction, with user number “uuuu” and project number “pppppppp”. This may be followed by a dispose instruction for printed output on the dataset $OUT (aliased to fortran unit 6). In the case shown, that dataset is disposed to the user’s NCAR IBM personal (virtual) machine (determined by the user and project numbers specified). The output will appear in the virtual reader with file name “INITAL” and filetype “OUTPUT”. Following this, there are three file names, one password, and one variable that must be user specified in the JCL file.
* THIS JOB IS FOR INITIALIZING DATA FOR MM4
* THE USER MUST PROVIDE:
* USER/PROJECT NUMBER IN JOB STATEMENT (uuuuppppppp)
* ITERATION NUMBER (jj=3 recommended)
* DATA FILE NAME (THIS MUST BE AN MM4 DATA INPUT FILE; iiii)
* BOUNDARY CONDITION FILE NAME (bbbbbb)
* OUTPUT (initialized) FILE NAME (2 PLACES; oooooo)
* WRITE PROTECT PASSWORD FOR TBM VOLUME (W=wwwwww)
*
* ADDITIONALLY, IF GRID DIMENSIONS, SIZES, OR SIGMA LEVELS ARE CHANGED, THOSE CHANGES MUST BE MADE CONSISTENTLY IN ALL PROGRAMS AS DESCRIBED BELOW.
*
*
* SET NUMBER OF ITERATIONS
SET(J1=jj)
*
* ACQUIRE INITIAL DATA FILES
ACQUIRE,DN=FT61,PDN=iiiiii,MF=MS,TEXT='FLNM=''iiiiii'''. INITIAL FILE
ACQUIRE,DN=FT11,PDN=bbbbbb,MF=MS,TEXT='FLNM=''bbbbbb'''. BOUNDARY FILE
*
* ACQUIRE INITIALIZATION PROGRAM.
ACQUIRE,DN=INITAL,PDN=READMP,MF=MS,ID=ADMP,TEXT='USER=RMERRICO'.
UPDATE,P=INITAL,C=CVMODE,Q=VMODES,ED,ID.
UPDATE,P=INITAL,C=CINITL,Q=INITAL:FSUBS,ED.
CFT,I=CVMODE,B=BVMODE,L=O.
CFT,I=CINITL,B=BINITL,L=O.
LDR,DN=BINITL,AB=ABINIT,NX.
*
* ACQUIRE MODEL LIBRARY, UPDATE IT, AND CREATE LOAD MODULE.
ACQUIRE,DN=FT12,PDN=EHTRAN,MF=MS,ID=MM4MISC,TEXT='USER=MSKU'.
ACQUIRE,DN=MM4,PDN=MM4V3PL,MF=MS,ID=MM4MISC,TEXT='USER=MSKU'.
UPDATE,F,P=MM4,N=O,S=O,ED.
CFT,I=$CPL,OFF=CST,B=BMODEL.
LDR,DN=BMODEL,AB=ABMODL,NX,LIB=IMSLIB.
*

Fig. 12.1a: The CRAY JCL file for initializing a dataset for the MM4 (continued in Fig. 12.1b).
RUN STARTING PROGRAMS.
LDR,DN=BVMODE.
ABINIT.

RELEASE FILES NO LONGER NEEDED
RELEASE,DN=CVMODE.
RELEASE,DN=CINITL.
RELEASE,DN=BVMODE.
RELEASE,DN=BINITL.
RELEASE,DN=INITAL.
RELEASE,DN=$CPL.
RELEASE,DN=BMODEL.
RELEASE,DN=MM4.

BEGIN LOOP
SET(J1=J1+1)
ENDLOOP.

Fig. 12.1b: The CRAY JCL file for initializing a dataset for the MM4 (continued from Fig. 12.1a).
jj is a JCL variable that prescribes the number of nonlinear iterations, (excluding the starting iteration). For appropriate values, see Section 12.6.

iiiiii is the name of the volume of data to be initialized.

bbbbbb is the name of the volume of containing corresponding lateral boundary conditions.

oooooo is the name of the volume to be created which will contain the initialized data, in the same format as for the initial data.

wwwwww is a password name for dataset write protection on the disposed dataset.

12.2 VMODES UPDATE file

Immediately following the JCL file is a file containing UPDATE instructions for editing the program VMODES. For most purposes, only two sets of modifications need be made. The first is to set parameters, and the second is to specify a set of $\sigma$-levels.

The three parameters to set are:

KX, the number of data levels in the model, defined identically in the MM4;

PTOP, the pressure at the model top (in centibars), defined identically in the MM4;

PSMEAN, a reference value for surface pressure (in centibars; use the value 100).
The values of σ which must be specified are those indexed by integral values in the notation of Anthes and Warner (1978). They are specified by assigning KX+1 values to an array SIGMAF in KX+1 arithmetic assignment statements. They must agree with the sigma levels defined in the model and with those used to determine the data. The first and last values of SIGMAF should be 0 and 1, respectively.

```fortran
BEGIN MODS TO VMODES
*ID SETRUN
*D VMODES.4
*
/* SET KX, PTOP, AND PSMEAN, AS DESCRIBED BELOW.
*/
PARAMETER (KX=10, PTOP=10., PSMEAN=100. )
/
/* KX = NUMBER OF MODEL DATA LEVELS (SAME AS IN MM4).
*/ PTOP = PRESSURE AT TOP (CBARS; SAME AS IN THE MM4)
/* PSMEAN = 100.
*/
*D VMODES.38,53
*/ SET SIGMAF (SIGMA AT FULL (INTEGRAL) INDEX LEVELS; KX+1 VALUES
*/ IN THE MM4, THE NAMELIST VARIABLE SIGMA MUST BE SET WITH SAME VALUES.
SIGMAF(1)=0.
SIGMAF(2)=.100
SIGMAF(3)=.200
SIGMAF(4)=.300
SIGMAF(5)=.400
SIGMAF(6)=.500
SIGMAF(7)=.600
SIGMAF(8)=.700
SIGMAF(9)=.800
SIGMAF(10)=.900
SIGMAF(11)=1.000
*/
END MODS TO VMODES
/*/ -----------------------------------------------
\EOF
```

Fig. 12.2: The file of CRAY UPDATE mods for the program that determines vertical modes of the 10-level MM4.
12.3 INITIAL UPDATE file

Immediately following the VMODES UPDATE file is a file containing UPDATE instructions for editing the program INITAL. For most purposes, only two sets of parameters need modification. The first is the set of grid dimensions, and the second describes other grid properties. These parameters are:

IX, JX, KX, defined identically in the MM4;

NIJMAX, the larger of the values of IX or JX;

DX, the grid separation in meters, defined identically in the MM4;

DT, a time step used to compute field time-tendencies, which should be defined identically in the MM4 (See Section 12.4). This value need not be the same in subsequent forecasts made with the initialized data. Therefore, DT=60. (seconds) is suitable in most cases;

CLAT, is a central latitude (degrees north) for the grid, from which a representative value of the coriolis parameter is computed. It must be nonzero.

12.4 MM4 UPDATE file

Immediately following the INITAL UPDATE file is a file containing UPDATE instructions for editing the MM4. These modifications serve to define grid dimensions and to make the model adiabatic, so that the nonlinear initialization is an adiabatic one. Also,
Fig. 12.3: The file of CRAY UPDATE mods for the initialization program for the specified resolution.
*ID MOD
*/
*/ MODS BASED ON MM4V3.V1:
*/ SET GRID DIMENSIONS
*/
*D PARAME.5
   PARAMETER (IX=46,JX=61,KX=10)
*D PARAME.12
   PARAMETER (MIX=46,MJX=61, MKX=10)
*/
*/ MODIFY OUTPUT TIME, AND STOP SAVE FROM BEING WRITTEN
*/
*D OUTPUT.18,28
   IF (KTAU.EQ.1) CALL OUTTAP (IUT20, IUT30)
*/
*/ REMOVE HORIZONTAL DIFFUSION
*/
*I DIFFU.46
   RETURN
*I DIFFUT.41
   RETURN
*/
*/ REMOVE CONVECTIVE AND NONCONVECTIVE LATENT HEATING
*/
*D CUPARA.223
*D NCONVP.85,91
*/
*/ SPECIFY TIME STEP (MUST BE SAME IN INITIAL)
*/
*I PARAM.887
   DT=60.
*/
*/ ANY MODEL CORRECTIONS GO HERE
*/
*/ END MODS TO MM4
*/
\EOF

Fig. 12.4: The file of CRAY UPDATE mods for the adiabatic initialization of the MM4 with indicated resolution.
the output of forecast data is modified so that only data corresponding to the first model
time step is written on the output file.

IX, JX, KX, defined identically in INITAL;

DX, the grid separation in meters, defined identically in the MM4;

DT, a time step used to compute field time-tendencies, which should be defined identically in INITAL (See Section 12.3). This value need not be the same in subsequent forecasts made with the initialized data. Therefore, DT=60. (seconds) is suitable in most cases;

12.5 NAMELIST data input files

Immediately following the MM4 UPDATE file are three files of NAMELIST data.

The first file contains data input for execution of the starting iterate of the program INITAL. This data is in the form of a set (named LINIT) of NAMELIST variables. The FORTRAN unit number specified by values of IUNIT2 and IUNIT4 are both set to 61, indicating that initial data (on the volume iiiiii; Section 12.1) is found on unit 61. The value of NMODES is set to the number of vertical modes to be initialized (For a discussion of appropriate values see Section 12.6). The value of IFUNCT is set to 0 since this is a starting iteration (the only other possibility for a starting iterate is 1, indicating a linear initialization to start, but this is not recommended). All other data input assume default values.
Fig. 12.5: Data input files for a starting iteration of the program INITAL, the MM4, and a nonlinear iteration in the program INITAL.
The second file contains data input for execution of the MM4. This data is in the form of three sets (named OPARAM, LPARAM, and PPARAM) of NAMELIST variables. The first set is specified to limit printer output (from the MM4) and to execute one time step only. The second set is specified to render the MM4 adiabatic (but using virtual temperature to compute the hydrostatic geopotential). Grid characteristics and σ-levels are specified in the the last set. These last must be set equal to similar parameters in the UPDATEs of VMODES and INITAL (Sections 12.2 and 12.3, respectively). TIMAX should be equal to or slightly greater than DT. Other namelist variables, not shown, assume their default values.

The last file contains data input for execution of a non-starting, nonlinear iterate of the program INITAL. This data is in the form of a set (named LINIT) of NAMELIST variables. For this case, all NAMELIST variables may assume their default values, except NMODES should be set equal to its value specified for the starting iteration (default value is 2), as described in the beginning of this Section, and IFUNCT=2 must be specified since this is a nonlinear initialization.

12.6 The choice of NMODES and J1

The number of vertical modes is the same as the value of KX, the number of data σ-levels. The modes are ordered from largest equivalent depth to smallest. The value of NMODES specifies that the first thru the NMODEth modes are to be initialized. That means those portions of the dynamic fields which are described by the remaining modes
are not modified by the initialization scheme when IFUNCT=1 or 2 (no modification is done when IFUNCT=0). NMODES ≤ KX.

There are two primary considerations for choosing a particular value of NMODES. The purpose of the initialization is to balance the initial data so that relatively large-amplitude, high-frequency oscillations are absent from the initial period of a forecast. Therefore, a sufficient number of modes must be initialized so that this is the case. Oscillations of the surface pressure are most affected by external gravity waves, so most of such oscillations will be removed if only the first mode is initialized (NMODES = 1). However, further improvement is observed when the first internal mode is initialized as well (NMODES = 2). Of course, if gravity waves of a particular vertical scales are to be removed by modification of the initial data, the corresponding vertical modes must be initialized. The more modes initialized, the more the initial data are likely to be altered by the initialization procedure.

The other primary consideration for choosing a value for NMODES is that the Machenhauer (1977) scheme used to iterate the nonlinear balance equations does not always converge to a solution (for reasons described in Errico, 1983). The lack of convergence occurs especially when shallow modes are being initialized; i.e., for this program when NMODES is large. Therefore, it is not advisable to set NMODES greater than the value 3.

If the iteration scheme converges, increasing the value of J1 beyond some value will neither change the fields significantly or decrease the degree of oscillations observed. The balance condition described in Section 2.2 is only an approximation (for further discussion of this point, see Leith, 1981; Lorenz, 1981; Errico, 1982; Tribbia, 1984). For these reasons,
and to limit costs, it is appropriate to limit the value of $J_1$. For most purposes, a value of 2 or 3 is sufficient. If the former value is chosen but the forecast appears to be noisy still, the user should try to increase either or both of $J_1$ and NMODES by 1.

The CRAY job actually executes the program INITAL $J_1+2$ times. The first time is the starting (0th) iterate. The purpose of the last execution is only to compute diagnostics (described in Section 12.7). During the $N$th execution, diagnostics which describe the convergence of the initialization scheme after the previous $((N-1)\text{th})$ execution are computed and printed. Therefore, convergence properties of the created balanced dataset are not printed unless the number of (nonlinear) executions is actually one more than the number of desired iterations.

12.7 Explanation of printed diagnostics

The program INITAL calls subroutines to compute and print various diagnostics. These diagnostics should be examined after each run to ensure that the program execution was successful (although not all possible problems will be revealed by the limited set of diagnostics). An example of such printed output appears in Section 14.

First, the statement “INITIALIZATION PROGRAM EXECUTING” is printed at the top of a new page. This simply indicates that the program INITAL has been entered. This statement is then followed by the namelist output of the list named LINIT (described in detail in Section 10.1.2) Next is a statement indicating the function of this particular
This preliminary information is followed by some data tables. Each table describes the root-mean-squared (rms) values for some fields. They are presented as a function of vertical mode index $K$. Values of the corresponding equivalent depths ($H$, multiplied by the gravitational constant $g = 9.8\text{ms}^{-2}$) of the modes are also printed, denoted by $ED$ (units of $\text{m}^2\text{s}^{-2}$). The rms values are calculated as in Eq. 4.2.12.

The first set of data includes values for the fields of vorticity ($\zeta$ denoted by $Z$ for zeta), horizontal velocity divergence ($\delta$ denoted by $D$), pseudo-geopotential ($\hat{h}$ denoted by $G$), and linearized potential vorticity ($\zeta - (f_0/gH)\hat{h}$) denoted by $Z-(F/H)G$). These fields are computed as defined by Eqs. (4.2.2), (4.2.3), (2.1.26), and (2.2.6), and units are $\text{s}^{-1}$, $\text{s}^1$, $\text{m}^2\text{s}^{-2}$, and $\text{s}^{-1}$, respectively. The hat indicates values with respect to vertical modes (2.3.10), and not $\sigma$-levels.

The remaining printout includes rms values of the time derivatives ($D/DT$) and changes ($\Delta$) of the fields in the respective columns of the first set of data. The time derivatives are those derived from the fields determined for the previous iteration, while the changes ($\Delta$) are those determined for the current iteration. Units are as discussed above, except the time derivatives are per second. The maximum absolute change in surface pressure ($\Delta p$) is also printed. The values of $WP=WT=0$ indicate that $\Delta p$ was computed as described in Section 2.5. Finally, the date read from the input data file is printed, along with messages that a file has been written and that this particular iteration terminated successfully.
12.8 Use of subroutine PRINTF

If the user desires contour plots and printed arrays of fields in the I,J domain, the user may simply insert FORTRAN CALLs to the subroutine PRINTF (PRINT Fields) using CRAY UPDATE instructions to the decks INITAL or FSUBS. Contour plots are generated using the NCAR library routine CONREC. This output is especially useful for program checking purposes.

Unless the routine PRINTF is modified for 3-dimensional fields, not more than 3 I,J plots will be produced (this restriction reduces the quantity of output). The third dimension K will typically be either \( \sigma \)-levels or vertical-mode indecies. All values of I and J appear in the contour plots. However, the printed arrays will not be larger than \( 10 \times 12 \) (I \( \times \) J) values. Values of I, J, or K are spaced at equal intervals (distinct intervals for the three indecies).

A label is written for each array or plot output. A field name 1-8 characters long is written along with the value of the third dimension index K (K=1 for single level fields). In printed output, the intervals between successive I and J values are also indicated. The value of a variable denoted INDEX is also written. It is set by the user in the CALL to PRINTF. Specification of this variable allows the user to indicate the same field name in different parts of the program without confusion.

For instructions on how to CALL the routine PRINTF the user should examine the subroutine listing from the deck INITAL. Examples of such calls appear in the decks INITAL and FSUBS. Some sample output appears in Section 15.
PART 5. SOFTWARE DEMONSTRATION

In Sections 13-16, a demonstration execution of the initialization software is discussed. The intention is to describe some of the output and performance of the various programs described in this report. Also the effects of an initialization on forecast performance is discussed briefly in order to highlight some properties of the initialized fields.

The vertical modes of two versions of the MM4 are presented in Section 13. The execution and convergence of the initialization program is described in Section 14 for one particular case. Some of the initial analyzed fields and the differences added to them by the initialization procedure are presented in Section 15 for that case. Then the effects of the initialization on a model forecast for that case are described briefly in Section 16.

Section 13. Vertical Modes of the MM4

The output of VMODES as applied to the MM4 is discussed in this section. The relevant finite differencing of the MM4 has been discussed in Section 4.1. The program VMODES has been discussed in Part 2.

The output described here has been computed using the default values for the various physical constants. These are $R = 287 \text{J/K}$, $\kappa = 0.287$, $p_T = 10 \text{ cb}$, and $\rho = 100 \text{ cb}$. Results are presented for the 10 and 15 level versions of the MM4. The specified $\sigma$-levels, the derived (linearly interpolated) data $\sigma$-levels, the corresponding (standard atmospheric)
values of $\bar{T}$, and the determined set of equivalent depths $H_i$ for each version appear in Tables 13.1 and 13.2 respectively.

The matrices $\mathbf{B}$ and $\mathbf{B}^{-1}$ (the hydrostatic matrix defined by (6.1) and its inverse), and $\mathbf{Z}$ and $\mathbf{Z}^{-1}$ (the matrix of vertical structure functions (modes) defined by (2.3.8) and its inverse) for the 10-level version are presented in Tables 13.3-13.6, respectively. Each column of $\mathbf{Z}$ describes a single vertical mode. The product of a mode's amplitude and each component of such a column is the value of the mode at the corresponding data $\sigma$-level.

The vertical modes are plotted as a function of $\sigma$-level for the 10 and 15 level model versions in Figs. 13.1-13.2, respectively. For these plots, the amplitude of each mode is assumed to be 1. For further discussion of the modes see Sections 2.3 and 7.
<table>
<thead>
<tr>
<th>( k ) or ( i )</th>
<th>( \sigma_k )</th>
<th>( \sigma_{k+\frac{1}{2}} )</th>
<th>( \bar{T}_{k+\frac{1}{2}} ) (K)</th>
<th>( H_i ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.05</td>
<td>218.15</td>
<td>6889.61</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.15</td>
<td>218.18</td>
<td>476.60</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>0.25</td>
<td>232.07</td>
<td>111.23</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>0.35</td>
<td>243.12</td>
<td>34.04</td>
</tr>
<tr>
<td>5</td>
<td>0.4</td>
<td>0.45</td>
<td>252.38</td>
<td>13.77</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>0.55</td>
<td>260.38</td>
<td>6.36</td>
</tr>
<tr>
<td>7</td>
<td>0.6</td>
<td>0.65</td>
<td>267.46</td>
<td>3.04</td>
</tr>
<tr>
<td>8</td>
<td>0.7</td>
<td>0.75</td>
<td>273.82</td>
<td>1.40</td>
</tr>
<tr>
<td>9</td>
<td>0.8</td>
<td>0.85</td>
<td>279.60</td>
<td>0.54</td>
</tr>
<tr>
<td>10</td>
<td>0.9</td>
<td>0.95</td>
<td>284.92</td>
<td>0.13</td>
</tr>
<tr>
<td>11</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13.1. Specified \( \sigma_k \)-levels, derived (interpolated) \( \sigma_{k+\frac{1}{2}} \)-levels, standard atmospheric temperatures at data \( \sigma_{k+\frac{1}{2}} \)-levels, and calculated equivalent depths for the 10-level MM4.
<table>
<thead>
<tr>
<th>$k$ or $i$</th>
<th>$\sigma_k$</th>
<th>$\sigma_{k+\frac{1}{2}}$</th>
<th>$\overline{T}_{k+\frac{1}{2}}$ (K)</th>
<th>$H_i$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.00</td>
<td>.050</td>
<td>218.15</td>
<td>6889.302</td>
</tr>
<tr>
<td>2</td>
<td>.10</td>
<td>.150</td>
<td>218.18</td>
<td>476.852</td>
</tr>
<tr>
<td>3</td>
<td>.20</td>
<td>.250</td>
<td>232.07</td>
<td>111.532</td>
</tr>
<tr>
<td>4</td>
<td>.30</td>
<td>.350</td>
<td>243.12</td>
<td>34.365</td>
</tr>
<tr>
<td>5</td>
<td>.40</td>
<td>.450</td>
<td>252.38</td>
<td>14.068</td>
</tr>
<tr>
<td>6</td>
<td>.50</td>
<td>.550</td>
<td>260.38</td>
<td>6.670</td>
</tr>
<tr>
<td>7</td>
<td>.60</td>
<td>.650</td>
<td>267.46</td>
<td>3.391</td>
</tr>
<tr>
<td>8</td>
<td>.70</td>
<td>.740</td>
<td>273.21</td>
<td>1.773</td>
</tr>
<tr>
<td>9</td>
<td>.78</td>
<td>.810</td>
<td>277.35</td>
<td>.926</td>
</tr>
<tr>
<td>10</td>
<td>.84</td>
<td>.865</td>
<td>280.43</td>
<td>.474</td>
</tr>
<tr>
<td>11</td>
<td>.89</td>
<td>.910</td>
<td>282.84</td>
<td>.233</td>
</tr>
<tr>
<td>12</td>
<td>.93</td>
<td>.945</td>
<td>284.66</td>
<td>.107</td>
</tr>
<tr>
<td>13</td>
<td>.96</td>
<td>.970</td>
<td>285.94</td>
<td>.044</td>
</tr>
<tr>
<td>14</td>
<td>.98</td>
<td>.985</td>
<td>286.69</td>
<td>.015</td>
</tr>
<tr>
<td>15</td>
<td>.99</td>
<td>.995</td>
<td>287.18</td>
<td>.003</td>
</tr>
<tr>
<td>16</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13.2. Specified $\sigma_k$-levels, derived (interpolated) $\sigma_{k+\frac{1}{2}}$-levels, standard atmospheric temperatures at data $(\sigma_{k+\frac{1}{2}})$-levels, and calculated equivalent depths for the 15-level MM4.
<table>
<thead>
<tr>
<th>Table 13.3. The hydrostatic matrix for the MM4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.241 0.404 0.284 0.220 0.180 0.152 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.162 0.284 0.220 0.180 0.152 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.122 0.220 0.180 0.152 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.098 0.180 0.152 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.082 0.152 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.070 0.132 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.062 0.117 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.055 0.104 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.049 0.096</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 13.4. Inverse of the hydrostatic matrix for the MM4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000 0.000 8.182 -18.371 22.384 -26.394 30.400 -34.406 38.410 -41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 10.190 -22.384 26.394 -30.400 34.406 -38.410 41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 12.195 -26.394 30.400 -34.406 38.410 -41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 14.199 -30.400 34.406 -38.410 41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 18.204 -38.410 41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 20.206 -41.924</td>
</tr>
<tr>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 21.718</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 13.5. The matrix of vertical modes of the MM4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.658 2.522 -1.308 -0.441 0.282 0.177 0.135 -0.091 0.062 0.031</td>
</tr>
<tr>
<td>0.858 0.965 1.136 1.623 -1.448 -1.234 -0.983 0.740 -0.498 -0.259</td>
</tr>
<tr>
<td>0.957 0.149 1.670 0.612 0.727 1.483 1.674 -1.491 1.102 0.538</td>
</tr>
<tr>
<td>1.010 -0.202 1.126 -0.998 1.592 0.488 -0.814 1.502 -1.456 -0.896</td>
</tr>
<tr>
<td>1.041 -0.428 0.485 -1.515 0.129 -1.462 -0.898 -0.639 1.493 1.190</td>
</tr>
<tr>
<td>1.061 -0.574 -0.070 -1.138 -1.206 -0.570 1.322 -0.740 -1.015 -1.421</td>
</tr>
<tr>
<td>1.073 -0.667 -0.486 -0.389 -1.270 1.070 0.414 1.387 0.062 1.512</td>
</tr>
<tr>
<td>1.081 -0.724 -0.767 0.342 -0.424 1.138 -1.256 -0.386 0.970 -1.386</td>
</tr>
<tr>
<td>1.086 -0.756 -0.932 0.855 0.543 -0.051 -0.567 -1.113 -1.344 0.998</td>
</tr>
<tr>
<td>1.089 -0.770 -1.004 1.100 1.098 -1.061 0.969 0.835 0.636 -0.367</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 13.6. Inverse of the matrix of vertical modes of the MM4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.113 0.107 0.105 0.103 0.101 0.100 0.098 0.097 0.096 0.095</td>
</tr>
<tr>
<td>0.279 0.096 0.041 0.005 -0.020 -0.037 -0.048 -0.056 -0.062 -0.065</td>
</tr>
<tr>
<td>-0.135 0.134 0.156 0.110 0.050 -0.004 -0.046 -0.074 -0.091 -0.098</td>
</tr>
<tr>
<td>-0.062 -0.127 0.048 0.151 0.021 -0.126 -0.142 -0.052 0.060 0.129</td>
</tr>
<tr>
<td>0.025 -0.093 0.105 0.054 -0.147 -0.071 0.123 0.146 -0.001 -0.138</td>
</tr>
<tr>
<td>0.016 -0.065 0.111 -0.062 -0.095 0.147 0.057 -0.167 -0.084 0.142</td>
</tr>
<tr>
<td>-0.010 0.043 -0.090 0.114 -0.054 -0.088 0.174 -0.047 -0.173 0.132</td>
</tr>
<tr>
<td>0.006 -0.025 0.059 -0.102 0.129 -0.103 0.002 0.137 -0.207 0.104</td>
</tr>
<tr>
<td>0.003 -0.011 0.027 -0.054 0.090 -0.130 0.162 -0.171 0.139 -0.055</td>
</tr>
</tbody>
</table>
Fig. 13.1: The structures of the 10 vertical modes of the 10-level MM4, with \( \sigma \)-spacing and \( \bar{T} \) defined in Table 13.1. The vertical mode numbers (indices) are indicated in the figures.
Fig. 13.2: The structures of the 15 vertical modes of the 15-level MM4, with σ-spacing and \( \bar{T} \) defined in Table 13.2. The vertical mode numbers (indices) are indicated in the figures.
Section 14. Convergence of Initialization Algorithm

In Sections 14-16, results obtained from the initialization of input data for the MM4 are described. Version 3 of the MM4 is used, with IX=46, JX=61, and KX=10. The spacing of the σ-levels for this version has been discussed in Section 13. Non-initialized input data are obtained from an analysis produced by the National Meteorological Center (NMC) and enhanced by a Cressman-type re-analysis of rawindsonde data (Seaman and Haagenson, 1985). Initial data are for 0Z 22 April 1981. The fields of data are described in Section 15.

The initialization is applied to the 3 largest-scale vertical modes only (NMODES=3 in the namelist LINIT input). There are 3 iterations performed (J1=3 in the JCL). The iterations are begun from the non-initialized data (IFUNCT=0 in the first namelist LIST1, for no modifications to start). The initialization is made adiabatic by eliminating all diabatic processes from the model version which is coupled to the initialization software (Section 12.4). The job deck used to produce the initialization appears in Figs. 12.1-12.5, although specific values of the various parameters (as described above) are not indicated there.

A sample of output from execution of the initialization software appears in Fig. 14.1. Only that output from execution of the program INITAL is shown, and only for a single iteration (the first nonlinear iteration). The first line is an indication that the program INITAL has begun. This is followed by a listing of the namelist LINIT as input to the program. Diagnostics follow. These are those produced during the first execution of the
**Figure 14.1** Sample output from the program `INITAL` during execution of the first nonlinear iteration.

```
INITIALIZATION PROGRAM TERMINATED SUCCESSFULLY

MAXIMUM ABSOLUTE CHANGE IN PS IS 5.20E-01 CBARS, WITH WEIGHTS WP= 0.00E+00, WT= 0.00E+00
FILE READ FOR IDATE = 04/22/81 AT 00Z
FILE WRITTEN FOR IDATE = 04/22/81 AT 00Z
```
nonlinear portion of an initialization procedure. The diagnostics output are described in
Section 12.7. For this particular iteration, the first set of values (for rms $\zeta$, $\delta$, $\hat{h}$, and $\hat{P}$ are
those for the non-initialized data. The corresponding time tendencies are those evaluated
for the fields of non-initialized data. The corresponding values of field changes are those
produced for the first nonlinear iteration. Note that only the first three vertical modes
have nonzero changes.

Convergence of the initialization algorithm is illustrated in Figs. 14.2 and 14.3. In Fig.
14.2, values for the rms time tendencies of $\zeta$, $\delta$, and $\hat{h}$ are presented as a function of iteration
number and vertical mode index. Only the first 4 modes are presented, since values for
the others do not change significantly with iteration number since NMODES=3 (e.g., see
values for $\partial \delta / \partial t$). Therefore those other values may be simply read from Fig. 14.1. Values
for iteration number 1 which are plotted in Fig 14.2 are taken from corresponding values
printed in Fig 14.1.

In Fig. 14.3, rms values of $\Delta \zeta$, $\Delta \delta$, and $\Delta \hat{h}$ are presented as a function of iteration
number and vertical mode index. Only the first 3 modes are presented, since the others
are zero valued. The rms values of $\Delta p$ as a function of iteration number also appear in
Fig. 14.3. Note that for all fields, the values decrease with increasing iteration number,
although for some fields, the decreases are greater. It also appears that for more iterations,
further decrease may not be observed. Values for only 3 iterations are presented in Fig.
14.3, since those computed for the last iteration do not correspond to changes produced
in the initialized dataset (Section 12.6). Values for iteration number 1 which are plotted
in Fig 14.3 are taken from corresponding values printed in Fig 14.1.
Fig. 14.2: The root-mean-squared time tendencies of vorticity, divergence, and pseudo-geopotential as a function of iteration number (indicated in figure) and vertical mode index (abscissa). $\partial \zeta / \partial t$ and $\partial \delta / \partial t$ in units of $s^{-2}$. $\partial h / \partial t$ in units of $10^7 m^2 s^{-3}$. 
Fig. 14.3: The rms adjustments of vorticity, divergence, pseudo-geopotential, and surface pressure as a function of iteration number (indicated in figure) and vertical mode index (abscissa). $\Delta \zeta$ and $\Delta \delta$ in units of s$^{-1}$. $\Delta h$ in units of $10^8$ m$^2$s$^{-2}$. $\Delta p$ in units of $10^6$ cb.
Section 15. Initialized Modifications to Fields

In this section, some field modifications are presented, as produced by the particular experiment described in Section 14. The domain of the cross grid used in this test appears in Fig. 15.1. The domain of the dot grid is only slightly larger, with both grids having identical centers.

Figure 15.2 shows the hydrostatic thickness \( \left( \frac{\phi - \phi_S}{g} \right) \) between the surface and the 0.25 \( \sigma \)-level determined from the non-initialized analysis at the initial time. The field is characterized by two troughs: one over the Rocky Mountains; the other over Quebec. The differences produced by the initialization also appear in Fig. 15.2. Note that the magnitudes of the differences are less than 40 m, resulting from vertical mean temperature differences of less than 1.5 K.

Figure 15.3 shows the \( u \) field at the 0.25 \( \sigma \)-level determined from the non-initialized analysis at the initial time. The field is characterized by westerlies almost everywhere, with a maximum value of 60.8 m/s. The differences produced by the initialization also appear in Fig. 15.3. Note that the magnitudes of the differences are less than 5.1 m/s. Differences for the \( v \) field are comparable.

Figure 15.4 shows the \( p_S = p^* + p_T \) field determined from the non-initialized analysis at the initial time. The field is strongly influenced by topography. The Rocky Mountains, Mexican Plateau, and Appalachian Mountains are particularly evident. The differences produced by the initialization also appear in Fig. 15.4. Note that the magnitudes of the differences are less than 4.5 mb. The largest differences are over mountains, where both
the topography and $p_S$ vary most greatly, and where the analyzed pressure probably has least accuracy. The rms difference is 1.56 mb.

Figure 15.5 shows the $T$ field at the 0.75 $\sigma$-level determined from the non-initialized analysis at the initial time. The structure of the field is similar to that of the thickness shown in Fig. 15.2, since both are related hydrostatically. The differences produced by the initialization also appear in Fig. 15.5. Note that the magnitudes of the differences are less than 0.8 K.

The rms differences (initialized minus analyzed) of the $u$, $v$, $T$, and $\phi/g$ fields for each $\sigma$-level are presented in Table 15.1.
Fig. 15.1: The domain of the cross grid for the demonstration experiment.
Fig. 15.2: (a.) The hydrostatic thickness between the surface and the 0.25 $\sigma$-level determined from the non-initialized analysis at the initial time. Contour interval is 100m. (b.) The difference in that field produced by the initialization. Contour interval is 8m.
Fig. 15.3: (a.) The $u$ field at the 0.25 $\sigma$-level of the non-initialized analysis at the initial time. Contour interval is 10 m/s. (b.) The difference in that field produced by the initialization. Contour interval is 1 m/s.
Fig. 15.4: (a.) The surface pressure field of the non-initialized analysis at the initial time. Contour interval is 30mb. (b.) The difference in that field produced by the initialization. Contour interval is 0.7mb.
Fig. 15.5: (a.) The $T$ field at the 0.75 $\sigma$-level of the non-initialized analysis at the initial time. Contour interval is 3K. (b.) The difference in that field produced by the initialization. Contour interval is 0.1K.
<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$u$(m/s)</th>
<th>$v$(m/s)</th>
<th>$T$(K)</th>
<th>$\phi/g$(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.05</td>
<td>1.73</td>
<td>1.56</td>
<td>0.67</td>
<td>17.0</td>
</tr>
<tr>
<td>.15</td>
<td>1.41</td>
<td>1.31</td>
<td>0.68</td>
<td>13.1</td>
</tr>
<tr>
<td>.25</td>
<td>1.54</td>
<td>1.45</td>
<td>0.35</td>
<td>11.2</td>
</tr>
<tr>
<td>.35</td>
<td>0.98</td>
<td>0.92</td>
<td>0.46</td>
<td>8.4</td>
</tr>
<tr>
<td>.45</td>
<td>0.43</td>
<td>0.39</td>
<td>0.47</td>
<td>5.8</td>
</tr>
<tr>
<td>.55</td>
<td>0.37</td>
<td>0.32</td>
<td>0.42</td>
<td>3.6</td>
</tr>
<tr>
<td>.65</td>
<td>0.68</td>
<td>0.63</td>
<td>0.34</td>
<td>2.1</td>
</tr>
<tr>
<td>.75</td>
<td>0.93</td>
<td>0.86</td>
<td>0.24</td>
<td>1.1</td>
</tr>
<tr>
<td>.85</td>
<td>1.07</td>
<td>1.00</td>
<td>0.15</td>
<td>0.5</td>
</tr>
<tr>
<td>.95</td>
<td>1.14</td>
<td>1.06</td>
<td>0.10</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 15.1 The rms differences between initialized and non-initialized fields at each $\sigma$–level at the initial time.
Section 16. Comparison of MM4 Forecasts

As stated in the Introduction, the primary motivation for initialization is to reduce the amount of gravity-wave noise which otherwise would be present in a forecast. Therefore, a primary check of the initialization procedure is to compare time series of various fields at various points as produced by forecasts begun from initialized and non-initialized data. In this section, two such forecasts are described, starting from the data described in Sections 14-15. The forecast model is the diabatic MM4 (resolution as described in Section 14) with time-dependent lateral boundary conditions interpolated from non-initialized analyses produced for hours 0, 12 and 24.

Time series of $p_s$, $u$, $T$, and $\omega$ (the last three at the 0.45 $\sigma$-level) at the point $I=28$, $J=42$ (near Detroit, Michigan) for 15-hour forecasts begun from the initialized and non-initialized datasets for 0Z 22 April 1981 appear in Fig 16.1. Comparisons for time series produced at other points yields qualitatively similar results.

Note that, for the non-initialized data, $p_s$ changes by as much as 2.5 mb in one hour, whereas the time series for the initialized forecast is much smoother. The time series of $u$ from the initialized forecast is also much smoother than that from the non-initialized forecast, although the oscillations for the latter are not as dramatic as those for $p_s$. Presumably, the oscillations in $u$ are due to oscillations in the divergent part of the wind due to the presence of gravity waves. The differences between the other time series are qualitatively similar. Note that for each variable, the low frequency behavior of each forecast is quite similar, so that differences between initialized and non-initialized
Fig. 16.1: Time series of $p_s$, $u$, $T$, and $\omega$ (the last three at the 0.45 $\sigma$-level) at the point $I=28$, $J=42$ (near Detroit, Michigan) for 15-hour forecasts begun from the initialized (solid line) and non-initialized (dashed) datasets for OZ 22 April 1981.
Fig. 16.2: (a.) The difference of the fields of hydrostatic thickness as forecast at hour 12 starting from initialized and non-initialized datasets. Contour interval is 3m. (b.) The error in the hydrostatic thickness field (non-initialized forecast minus analyzed field at verification time). Contour interval is 10m. The hydrostatic thickness fields are determined between [111]
forecasts at hour 15 are quite small. In other words, the oscillations have been removed
without substantially affecting the forecast dynamic fields.

In general, the initialization procedure is quite effective at reducing oscillations of the
$ps$ field. Those oscillations are primarily due to gravity waves of large vertical scales. These
scales are the ones initialized. In contrast, oscillations of $\omega$ are not reduced as much since
that field is more greatly influenced by gravity waves of smaller internal scales. Those
scales are more difficult to initialize because of convergence problems of the algorithms
used in the initialization procedure (Section 12.6). In Fig. 16.1, the noise reduction looks
substantial, although this very good result may be atypical.

Figure 16.2 shows the differences between the initialized and non-initialized forecasts
of the hydrostatic thickness $((\phi - \phi_s)/g)$ evaluated between the surface and the 0.25 $\sigma$-level
at hour 12. The range of values is one-half that at the initial time (Fig. 15.2). The non-
initialized forecast is geostrophically adjusting by propagating unbalanced gravity waves
out of the forecast domain. Therefore, as time proceeds, it approaches the initialized
forecast.

The errors in the non-initialized forecast field of the same hydrostatic thickness at hour
12 (forecast minus analyzed field at the verification time) also appear in Fig. 16.2. Note
that the differences between forecasts are much smaller than the non-initialized forecast
ers. This indicates that the initialized and non-initialized forecasts are much more
similar to each other at hour 12 than either is to the analysis at hour 12. In other words,
the initialization is not significantly changing the meteorologically significant portion of the
fields, so that after a period of geostrophic adjustment, the forecasts are similar. Also note that the analysis at hour 12 is not balanced (i.e., has not been initialized), so that some of the forecast error is an indication of the inbalances present in the analysis but less present in the geostrophically-adjusted forecast.

Figure 16.3 shows the differences between the initialized and non-initialized forecasts of the $u$ field at the 0.25 $\sigma$-level at hour 12. The range of values is larger than that at the initial time (Fig. 15.3). Here there is some indication of predictability error growth (Errico and Baumhefner, 1986; Anthes et al., 1985). The errors in the non-initialized forecast field of the same $u$ field at hour 12 (forecast minus analyzed field at the verification time) also appear in Fig. 16.3. Note that the former differences are typically much smaller than the latter.

Figure 16.4 shows the differences between the initialized and non-initialized forecasts of the surface pressure ($p_s$) field at hour 12. The range of values is much smaller than that at the initial time (Fig. 15.4). The errors in the non-initialized forecast field of the $p_s$ field at hour 12 (forecast minus analyzed field at the verification time) also appear in Fig. 16.4. Remarks similar to those discussed for the thickness field apply here as well.

Figure 16.5 shows the differences between the initialized and non-initialized forecasts of the $T$ field at the 0.75 $\sigma$-level at hour 12. The range of values is similar to that at the initial time (Fig. 15.5). The errors in the non-initialized forecast field of the same $T$ field at hour 12 (forecast minus analyzed field at the verification time) also appear in Fig. 16.5. As for the other fields presented, the errors in either forecast are typically larger than the differences between the forecasts.
Fig. 16.3: (a.) The difference of the u fields at the 0.25 σ-level as forecast at hour 12 starting from initialized and non-initialized datasets. Contour interval is 0.7 m/s. (b.) The error in the u field at the 0.25 σ-level (non-initialized forecast minus analyzed field at verification time). Contour interval is 3.0 m/s.
Fig. 16.4: (a.) The difference of surface pressure as forecast at hour 12 starting from initialized and non-initialized datasets. Contour interval is 0.2mb. (b.) The error in the surface pressure field (non-initialized forecast minus analyzed field at verification time). Contour interval is 0.9mb.
Fig. 16.5: (a.) The difference of the $T$ fields at the 0.75 $\sigma$-level as forecast at hour 12 starting from initialized and non-initialized datasets. Contour interval is 0.1K. (b.) The error in the $T$ field at the 0.25 $\sigma$-level (non-initialized forecast minus analyzed field at verification time). Contour interval is 0.7K.
These results indicate that the initialization software is performing as intended. This is not to suggest that there are not problems which remain. Further discussion of these other problems or of the effects of initialization are beyond the scope of this report.
REFERENCES


1982a: Normal mode initialization and the generation of gravity waves by quasi-


