Naval Research Laboratory
Monterey, CA 93943-5502

NRL/MR/7533--93-7214

User's Guide for the Coupled Ocean/ Atmosphere Mesoscale Prediction System (COAMPS)

TRACY HAACK

Prediction Systems Branch
Marine Meteorology Division

November 1993

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*	2. Report Date.	3. Report Type and Dates Covered.
	November 1993	Final

4. Title and Subtitle.
User's Guide for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)

5. Funding Numbers.
PE 0602435N
PN R035E71
AN DN153155

6. Author(s).
Tracy Haack

7. Performing Organization Name(s) and Address(es).
Naval Research Laboratory
Marine Meteorology Division
Monterey, CA 93943-5502

8. Performing Organization Report Number.
NRL/MR/7533--93-7214

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Office of Naval Research
Arlington, VA 22217-5660

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) is a fully interactive coupled mesoscale model. It has been designed as a research tool for the simulation and study of a broad range of oceanic, atmospheric, and coupled ocean/atmosphere processes. Although the user's guide provides a brief description of the model equations and Fortran coding, more emphasis is placed on the preparation and execution of a COAMPS model run. The code has been written to allow the user to customize each simulation. This is accomplished by specifying a combination of input parameters, coding options and initialization features that best describe the particular process being studied. A flow chart of the COAMPS code and of the steps required to perform a simulation and submit a job are presented.

14. Subject Terms.		15. Number of Pages.
COAMPS Mesoscale model	Coupled ocean/atmosphere model	52
		16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Same as report

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. Agency Use Only (Leave blank).	2. Report Date. November 1993	3. Report Type and Dates Covered. Final

4. Title and Subtitle. User's Guide for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)	5. Funding Numbers. PE 0602435N PR R0135C71 AN DN153155
6. Author(s). Tracy Haack	

7. Performing Organization Name(s) and Address(es). Naval Research Laboratory Marine Meteorology Division Monterey, CA 93943-5502	8. Performing Organization Report Number. NRL/MR/7533--93-7214
9. Sponsoring/Monitoring Agency Name(s) and Address(es). Office of Naval Research Arlington, VA 22217-5660	10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.	
12a. Distribution/Availability Statement. Approved for public release; distribution unlimited.	12b. Distribution Code.

13. Abstract (Maximum 200 words).

The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) is a fully interactive coupled mesoscale model. It has been designed as a research tool for the simulation and study of a broad range of oceanic, atmospheric, and coupled ocean/atmosphere processes. Although the user's guide provides a brief description of the model equations and Fortran coding, more emphasis is placed on the preparation and execution of a COAMPS model run. The code has been written to allow the user to customize easy simulation. This is accomplished by specifying a combination of input parameters, coding options and initialization features that best describe the particular process being studied. A flow chart of the COAMPS code and of the steps required to perform a simulation and submit a job are presented.

14. Subject Terms. COAMPS Coupled ocean/atmosphere mesoscale model			15. Number of Pages. 52
			16. Price Code.
17. Security Classification of Report. UNCLASSIFIED	18. Security Classification of This Page. UNCLASSIFIED	19. Security Classification of Abstract.	20. Limitation of Abstract. Same as report

Contents

Acknowledgements

Many helpful suggestions were made by reviewers Dr. Richard Hodur, Dr. James Doyle, and Mr. William Thompson of NRL Monterey. The support of the sponsor, the Office of Naval Research, under program element 0602435N, is also gratefully acknowledged.

User's Guide for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)

1. Introduction

This report summarizes the procedure for running the Navy's nonhydrostatic Coupled Ocean/Atmosphere Mesoscale Prediction System (**COAMPS**). Developed and coded by Dr. Richard Hodur of NRL Monterey, **COAMPS** is constructed in a flexible way so that the user may specify many of the model parameters (e.g., grid point spacing, horizontal and vertical domain sizes, time steps, boundary conditions, and output fields, etc.).. Thus, the user selects appropriate input parameters that allow the model to simulate a particular atmospheric, oceanic, or coupled mesoscale phenomenon. Section 2 briefly describes the **COAMPS** model equations and parameterizations. Appendix A is frequently referred to for the definitions of the many model variables that appear in this section. Section 3 describes how to specify user input to a model run. Table 1 in this section serves as a reference list of the case specific model parameters, their definitions and default values. In Section 4, the **COAMPS** analysis code is outlined with particular attention given to generating a set of initial conditions. The **COAMPS** model code is also outlined in Section 4. Finally, compiling, linking and executing the analysis and model codes are performed via the use of a script file which is described in Section 5. Section 6 gives a quick summary of how to produce a variety of output and graphics files. Table 2 in this section provides a reference list of the output parameters that may be specified by the user.

2. Description

COAMPS couples a nonhydrostatic, fully compressible atmospheric system and a hydrostatic, incompressible oceanic system. Information is exchanged between the fluids via the surface fluxes of heat, moisture and momentum. These quantities are represented in the model using Monin-Obukhov similarity theory and the flux-profile relationships for temperature, humidity and moisture derived by Businger et al. (1971). The two surface parameterization schemes available in **COAMPS** are described by Barker and Baxter (1975) and Louis (1979).

In subsection 2, the atmospheric model equations and subgrid scale parameterization schemes are reviewed. In subsection 2b, the oceanic system of equations and subgrid scale parameterization are described.

a. The atmospheric model

1. Model equations

The atmospheric model is formulated after that of Klemp and Wilhelmson (1978) and includes predictive equations for the momentum components u_i

$$\frac{\partial u_i}{\partial t} + c_p\overline{\theta}_v\frac{\partial \pi}{\partial x_i} - D_{div_{d_i}} = -u_j\frac{\partial u_i}{\partial x_j} + \delta_{i3}\frac{g}{\overline{\theta}}\left[\theta' + \overline{\theta}(0.61q_v' - q_c' - q_r' - q_i' - q_s')\right] + \epsilon_{ij3}f(u_j - u_{g_j}) + D_{diff} + D_{Rayd} + D_{turb_{u_i}} \quad , \tag{1}$$

the nondimensional pressure perturbation π

$$\frac{\partial \pi}{\partial t} + \frac{\overline{c}^2}{c_p\overline{\rho}\overline{\theta}_v^2}\nabla_3\cdot(\overline{\rho}\overline{\theta}_v u_j) = -u_j\frac{\partial \pi}{\partial x_j} + \frac{R_d\pi}{c_v}\frac{\partial u_j}{\partial x_j} + \frac{c^2}{c_p\theta_v^2}\frac{d\theta_v}{dt} + D_{diff} + D_{turb_\pi} \quad , \tag{2}$$

and the potential temperature θ, and mixing ratios of water vapor q_v, clouds q_c, rain q_r, ice q_i and snow q_s, which are written in the form

$$\frac{\partial \phi}{\partial t} = -u_j\frac{\partial \phi}{\partial x_j} + M_\phi + D_{diff} + D_{Rayd} + D_{turb_\phi} \quad . \tag{3}$$

In the above equations the barred quantities are initial mean vertical profiles and the primed quantities are deviations from the mean. As each of the terms in equation (1)-(3) are described below, refer to Appendix A for variable definitions and explanations.

Equation (1) represents temporal changes to u_i that depend upon the pressure gradient and divergence damping on the left-hand-side, and on advection, buoyancy, Coriolis forcing, diffusion, Rayleigh damping and turbulent mixing on the right-hand-side. Equation (2) is derived from the compressible continuity equation and represents temporal changes to π. It includes a divergence term on the left-hand-side of the equation and terms for advection, divergence, heating, diffusion and turbulent mixing on the right-hand-side. Terms on the right-hand-side of (2) are typically quite small; however, in **COAMPS** they have been retained for completeness. Referring to equation (3), the terms producing temporal changes to the thermodynamic variables comprise advection, microphysical processes, diffusion, Rayleigh damping and turbulent mixing. Explanation of the microphysical parameterization, represented by M_ϕ, is provided in Rutledge and Hobbs (1983). Definitions for the diffusion D_{diff}, divergence damping D_{divd}, and Rayleigh damping D_{Rayd} are discussed in Section 2c. Representation of the subgrid scale turbulent mixing D_{turb} is described below.

2. Subgrid scale parameterization

COAMPS handles subgrid scale motions with a 1½ order, level-2.5 closure scheme for which a prognostic equation is derived for the turbulent kinetic energy. This scheme is described by Klemp and Wilhelmson (1978) and Deardorff (1980). Closure is achieved by parameterizing the turbulence terms D_{turb} in equations (1)-(3) by gradient approximations where

$$D_{turb_{u_i}} = \frac{\partial}{\partial x_j}\left(\overline{u_i' u_j'}\right) \quad ; \quad \overline{u_i' u_j'} = -K_m\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{2}{3}\delta_{ij} e \quad , \tag{4}$$

and

$$D_{turb_\phi} = \frac{\partial}{\partial x_i}\left(\overline{u_i' \phi'}\right) \quad ; \quad \overline{u_i' \phi'} = -K_h \frac{\partial \phi}{\partial x_i} \quad . \tag{5}$$

The mixing contribution to term D_{turb_π} is set to zero because it has virtually no influence on the solutions (Klemp and Wilhelmson, 1978). In equations (4)-(5), the overbar represents a grid volume average and the prime is a deviation from that average. The eddy mixing coefficients K_m for momentum and K_h for heat and moisture are defined by a turbulent length scale, l and a turbulent velocity scale, $e^{1/2}$

$$K_m = S_m l e^{1/2} \quad ; \quad K_h = S_h l e^{1/2} \quad , \tag{6}$$

modulated by the coefficients S_m and S_h. The model can be sensitive to the specifications of the turbulent mixing length l and the coefficients S_m and S_h, especially for large-eddy-simulations (LES).

The various options for defining these turbulence parameters are given in Section 3.

LES phenomenon are characterized by isotropic turbulence, whereas non-LES have vertical and horizontal turbulence scales differing by more than an order of magnitude. Consequently, non-LES applications of **COAMPS** utilize different horizontal and vertical expressions for K_m and K_h. While the vertical fluxes are parameterized with K_m and K_h defined by equation (6), the horizontal fluxes are parameterized with eddy mixing coefficients that depend on the horizontal deformation field:

$$K_m = \frac{b^2 \Delta x \Delta y}{\sqrt{2}} \left| \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right| \quad ; \quad K_h = 3K_m \quad , \tag{7}$$

where b is given in Appendix A (Smagorinsky, 1963). Later in this section and in Section 3(II), we discuss additional differences between the LES and non-LES turbulence parameterizations.

Returning to equation (6), we note that the eddy mixing coefficients depend on the turbulent kinetic energy (TKE) denoted by e. This quantity is determined by integration of a prognostic equation formulated after that of Klemp and Wilhelmson (1978) and Deardorff (1980):

$$\frac{\partial e}{\partial t} = - u_i \frac{\partial e}{\partial x_i} - \frac{\partial u_i}{\partial x_j} \overline{u_i' u_j'} + \frac{g}{\overline{\theta}} \overline{\left[w' \left(\theta' + \overline{\theta} (0.61 q_v' - q_c' - q_r' - q_i' - q_s') \right) \right]} - \frac{\partial}{\partial x_i} \overline{\left[u_i' \left(e' + p' / \overline{\rho} \right) \right]} - C_e \quad . \tag{8}$$

TKE is generated by contributions from advection, shear, buoyancy, diffusive and dissipative processes. Subgrid scale parameterizations for the three fluxes in equation (8) must also be specified. The diffusive flux is given by

$$\overline{u_i' \left(e' + p' / \overline{\rho} \right)} = -2K_m \frac{\partial e}{\partial x_i} \quad , \tag{9}$$

(Deardorff, 1980). The shear flux $(\overline{u_i' u_j'})$ is given by equation (4), and the buoyancy flux is given by

$$\overline{w' \left(\theta' + \overline{\theta} (0.61 q_v' - q_c' - q_r' - q_i' - q_s') \right)} = \begin{cases} -K_h(\gamma) & \textit{unsaturated motion,} \\ -K_h \left(A\gamma - \overline{\theta} \dfrac{\partial q_l}{\partial z} \right) & \textit{saturated motion.} \end{cases} \tag{10}$$

During saturation, the local lapse rate γ, is modulated by the moisture coefficient A, and corrected by the vertical gradient of

q_l, which is conserved in the absence of rain, snow and turbulent mixing (Klemp and Wilhelmson, 1978). The dissipation of TKE is defined by

$$C_e = Ce^{1.5}/l_{dis} \quad . \tag{11}$$

Expressions are given in Table 1 for the coefficient C and the dissipative mixing length l_{dis}. See Appendix A for the other variable definitions.

For non-LES applications of **COAMPS**, a few changes to the subgrid scale parameterization, based on a study by Therry and Lacarrère (1982), are emphasized here. First, two separate mixing lengths, l and l_{dis}, are used in the non-LES TKE formulation (in contrast, $l=l_{dis}$ for LES). These lengths are computed from an empirically derived function of the lapse rate, Monin-Obukhov length, TKE, and the boundary layer depth. Secondly, the parameterization for the buoyancy term for unsaturated motion (eqn. (10)) contains a countergradient correction γ_{cg}, to the lapse rate that is nonzero only in the unstable boundary layer. Finally, as described above, separate horizontal and vertical definitions are used for the eddy mixing coefficients during non-LES. In addition for non-LES model runs, the horizontal gradients from the shear term in equation (8) are eliminated, and the horizontal gradients and the term $2/3\delta_{ij}e$ in equation (4) are also eliminated.

The following procedure is used to obtain closure of the atmospheric system of equations (1)-(3). After appropriate expressions for l, S_m, S_h, and C are chosen by the user, e is predicted from equation (8) using the flux parameterizations (eqns. (4), (9)-(11)). Then, the eddy mixing coefficients K_m and K_h, are determined from the appropriate definitions (eqns. (6)-(7)). Finally, the subgrid scale turbulence terms D_{turb}, are computed from the gradient approximations (eqns. (4)-(5)) thereby closing the atmospheric system.

b. The oceanic model

1. Model equations

The ocean model is formulated after that developed by Chang (1985). It comprises predictive equations for the horizontal currents $u=u'+\overline{u}$ and $v=v'+\overline{v}$, for which the flow has been separated into a baroclinic response represented by the primes and a vertically averaged barotropic response represented by the overbars:

$$\frac{\partial u'}{\partial t} = \left[-u_j\frac{\partial u}{\partial x_j} + fv + D_{diff} + D_{turb_u}\right]' - \frac{1}{\rho_0}\frac{\partial p_r'}{\partial x} \quad . \tag{12}$$

$$\frac{\partial v'}{\partial t} = \left[-u_j\frac{\partial v}{\partial x_j} - fu + D_{diff} + D_{turb_v}\right]' - \frac{1}{\rho_0}\frac{\partial p_r'}{\partial y} \quad , \tag{13}$$

$$\frac{\partial \overline{u}}{\partial t} + \frac{1}{\rho_0}\frac{\partial \overline{p_r}}{\partial x} = \overline{\left[-u_j\frac{\partial u}{\partial x_j} + fv + D_{diff} + D_{turb_u}\right]} \quad , \tag{14}$$

$$\frac{\partial \overline{v}}{\partial t} + \frac{1}{\rho_0}\frac{\partial \overline{p_r}}{\partial y} = \overline{\left[-u_j\frac{\partial v}{\partial x_j} - fu + D_{diff} + D_{turb_v}\right]} \quad , \tag{15}$$

Prognostic equations are also derived for the perturbation height of the ocean h, or the height of the free surface,

$$\frac{\partial h}{\partial t} - \frac{\partial[h(u'+\overline{u})]}{\partial x} - \frac{\partial[h(v'+\overline{v})]}{\partial y} - H\left(\frac{\partial \overline{u}}{\partial x}+\frac{\partial \overline{v}}{\partial y}\right) = 0 \quad , \tag{16}$$

the temperature T

$$\frac{\partial T}{\partial t} = -u_i\frac{\partial T}{\partial x_i} + D_{diff} + D_{turb_T} \quad , \tag{17}$$

and the salinity S

$$\frac{\partial S}{\partial t} = -u_i\frac{\partial S}{\partial x_i} + D_{diff} + D_{turb_S} \quad . \tag{18}$$

The ocean pressure, vertical motion and density are respectively diagnosed from the hydrostatic equation:

$$\Delta p = \rho g \Delta z \quad , \tag{19}$$

the continuity equation:

$$\Delta w = -\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\Delta z \quad , \tag{20}$$

and the equation of state (Gill, 1982):

$$\rho = fn(p, S, T) \quad . \tag{21}$$

As mentioned previously, Appendix A contains the variables and

their definitions. In the above equations, the term D_{diff} represents numerical diffusion that is described in Section 2c. The term D_{turb} represents subgrid scale turbulent mixing for the ocean model that is summarized below.

2. Subgrid scale parameterization

Subgrid scale mixing within the ocean model is based on the 1st order, level-2 closure scheme of Mellor and Yamada (1974) for which the turbulent kinetic energy is computed diagnostically. They derive a steady state expression for TKE by assuming that the left-hand side, and the first and fourth terms on the right-hand side of equation (8) are small and therefore may be neglected. The resultant turbulent kinetic energy equation, after substituting definitions (4)-(6) (Section 2a) and simplifying, is given by

$$e = 15\, l^2 \left\{ \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right] S_m + \left[\frac{g}{\rho} \frac{\partial \rho}{\partial z} \right] S_h \right\} \quad , \tag{22}$$

(Clancy and Pollak, 1983). Note that in the ocean model, horizontal mixing is negligible and so the horizontal derivatives are not retained in the ocean TKE equation. In equation (22), the oceanic mixing length l is defined by

$$l = \frac{\kappa z}{1 + \frac{\kappa z}{l_{ocn}}} \quad , \tag{23}$$

where l_{ocn} provides an asymptotic limit on l given by

$$l_{ocn} = 0.1 \frac{\int_{-H}^{0} |z| e^{\frac{1}{2}} dz}{\int_{-H}^{0} e^{\frac{1}{2}} dz} \quad , \tag{24}$$

after Mellor and Yamada (1974). The coefficients S_m and S_h in equation (22) are stability functions that depend upon the flux Richardson number (Mellor and Yamada, 1974). These coefficients, along with the mixing length (eqns. (23)-(24)) are used to diagnose e. The above variables are then needed to compute the eddy mixing coefficients K_m and K_h (eqns. (6)). Finally, the ocean turbulence mixing terms D_{turb} are determined from eqnations (4) and (5), which closes the ocean system of equations (12)-(18).

c. Numerical techniques

1. Model Integration

Spatially, most of the terms and quantities in **COAMPS** are computed using second-order centered differencing on a scheme C staggered grid (Arakawa and Lamb, 1974) as shown in Figure 1. Note that the velocity components are displaced by one-half grid interval from a corresponding mass (π) grid point, thereby producing more accurate pressure gradient and divergence computations. All of the other model variables are computed at mass (π) points. The advection terms are computed with the flux-divergence form using second- or fourth-order differencing, except for potential temperature advection. The advection of potential temperature is computed with either the advective form, using the second- or fourth-order differencing schemes given by Klemp and Wilhelmson (1978), or using second-order upstream differencing.

Centered temporal differencing is used to integrate the terms on the right-hand-sides of the model equations (i.e., eqns. (1)-(3), (8) for the atmosphere and eqns. (12)-(18) for the ocean), except for the turbulent mixing terms (D_{turb}) which are integrated implicitly, and the diffusion and damping terms (D_{diff}, D_{Rayd}, D_{divd}) which are integrated with a forward time step to insure stability. Terms on the left-hand-side of each of the model equations incorporate rapidly propagating sound/gravity waves in the atmosphere and the faster moving barotropic response in the ocean. They are therefore treated with a time-splitting technique in which the larger time step increments Δt_a and Δt_o, are split into smaller time steps $\Delta \tau_a$ and $\Delta \tau_o$. Klemp and Wilhelmson (1978) give a detailed description of the time-splitting method. Additionally, the smaller, atmospheric time step integrations may be performed semi-implicitly in the vertical, allowing for a less restrictive stability condition on $\Delta \tau_a$. The stability limitations associated with each of the above time steps for the atmospheric and oceanic models are shown in Table 1 of Section 3.

2. Diffusion

To suppress the growth of numerical instabilities and the accumulation of energy at the smallest scales, several computational forms of damping and smoothing have been incorporated into the model code. The damping of spurious short wavelengths is accomplished with the numerical diffusion term D_{diff}. This quantity acts as a horizontal smoother via either a 2nd or 4th order laplacian operator on each of the predicted atmospheric and oceanic variables. For fourth-order diffusion this term is of the form

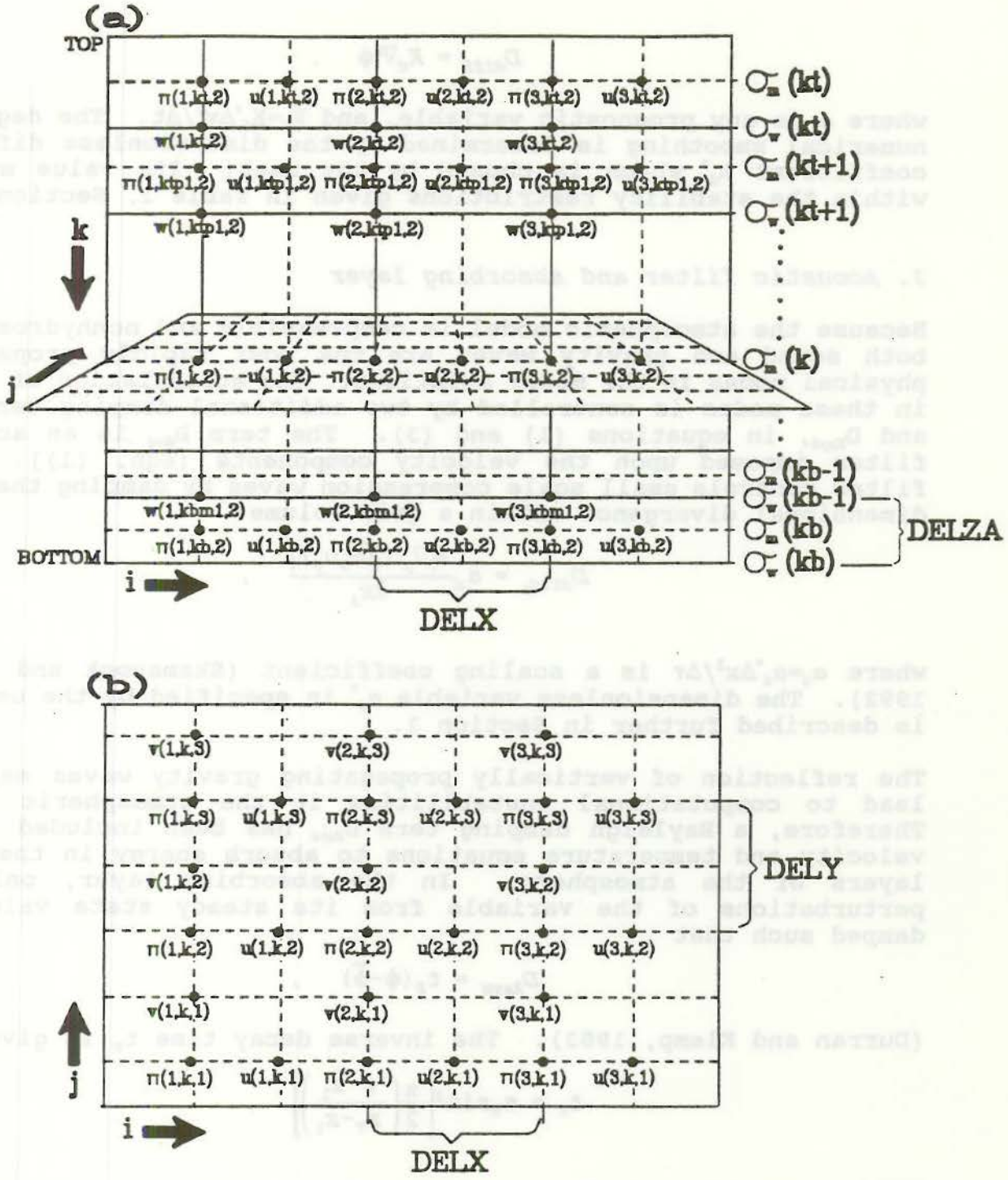

Figure 1: (a) Vertical and (b) horizontal grid structure of **COAMPS**. The σ_m and σ_w are the model height arrays; DELX, DELY and DELZA are the grid increments in the (i,j,k) directions respectively; the u,v,w are the velocity component arrays and the π is the dimensionless pressure perturbation array (indexed (i,k,j)).

$$D_{diff} = K_d \nabla^4 \phi \quad , \tag{25}$$

where ϕ is any prognostic variable, and $K_d = K_d^* \Delta x^4 / \Delta t$. The degree of numerical smoothing is determined by the dimensionless diffusion coefficient K_d^* which is chosen by the user. Its value must be within the stability restrictions given in Table 1, Section 3.

3. Acoustic filter and absorbing layer

Because the atmospheric model is compressible and nonhydrostatic, both sound and gravity waves are the most rapidly propagating physical modes in the model solutions. The accumulation of energy in these modes is controlled by two additional damping terms D_{divd} and D_{Rayd}, in equations (1) and (3). The term D_{divd} is an acoustic filter imposed upon the velocity components (eqn. (1)). This filter controls small scale compression waves by damping the three dimensional divergence within a grid volume:

$$D_{divd_i} = \alpha_d \frac{\partial [\nabla_3 \cdot (\bar{\rho} \bar{\theta}_v u_j)]}{\partial x_i} \quad , \tag{26}$$

where $\alpha_d = \alpha_d^* \Delta x^2 / \Delta \tau$ is a scaling coefficient (Skamarock and Klemp, 1992). The dimensionless variable α_d^* is specified by the user and is described further in Section 3.

The reflection of vertically propagating gravity waves may also lead to computational instabilities in the atmospheric model. Therefore, a Rayleigh damping term D_{Rayd} has been included in the velocity and temperature equations to absorb energy in the upper layers of the atmosphere. In the absorbing layer, only the perturbations of the variable from its steady state value are damped such that

$$D_{Rayd} = t_R (\phi - \bar{\phi}) \quad , \tag{27}$$

(Durran and Klemp, 1983). The inverse decay time t_R is given by

$$t_R = \alpha_R \sin^2 \left[\frac{\pi}{2} \left(\frac{z - z_1}{z_T - z_1} \right) \right] \quad , \tag{28}$$

(Klemp and Lilly, 1978), where α_R (units s^{-1}) is a user defined time scale discussed in Section 3. The other variables are defined in Appendix A.

4. Time filter

High frequency temporal oscillations may also be excited in the atmospheric model solutions. This result is due to the decoupling of the time levels associated with the leapfrog integration scheme. To keep the time levels coupled, a Robert (1966) time filter is applied to each atmospheric prognostic variable by weighting it against its previous and predicted values using the expression

$$\phi^t = \phi^t_* + \alpha_t(\phi^{t-1} - 2\phi^t_* + \phi^{t+1}_*) \quad , \tag{29}$$

where the (*) represents the prefiltered value. Shamarock and Klemp (1992) suggest giving the weighting coefficient α_t a value of 0.05 or less when divergence damping is enabled and a value of 0.1-0.2 otherwise.

3. Case specific parameters

In this section, model parameters that the user may specify are listed and described. Because of this feature, the user may customize **COAMPS** to simulate a particular physical process. The **COAMPS** include file *nhcd04.h* provides the input to the analysis and model executables. It contains a complete list of model input variables and default values. The user may change only those variables that are passed through the namelist **/nonhyd/** shown in Appendix B. Tables 1 and 2 contains those variable names, definitions (variable type), written descriptions, and default values most often customized by the user. Table 1 lists the important model initialization and runtime parameters, while those listed in Table 2 (see Section 6) are only for specifying various forms of output and graphical display data. Note that certain critical parameters are labeled with an asterisk (*). These are discussed in detail following the table. Also, parameters ending in **'a'** are generally for the atmospheric model, **'o'** for the ocean model. This section is to be used as a reference of model parameters and their definitions. A sample script file, provided in Appendix C and discussed in Section 5, shows how the user specifies these parameters in a **COAMPS** simulation.

Table 1: Case specific parameters
(See Appendix A for variable definitions.)

PARAM	DEFINITION (TYPE)	DESCRIPTION	DEFAULT
*AL0	l_{atm}	mixing length coef or upper limit (atmos)	100.0 m
ALPHA	(1) 1.0 (equal avg) (2) 0.0 (explicit) (3) 0.4 (damping)	semi-implicit time averaging coef: (1)Ikawa88 (2)Klemp&Wilhelm78 (3)Dudhia93	0.4
*CBAR	$\overline{c} = \sqrt{\frac{c_p R \overline{T_v}}{c_v}} \geq \frac{\Delta z}{\Delta \tau_a}$	speed of sound (atmos)	100.0 m/s
*CSTAR	$c_* = Nz_T/pi$	gravity wave speed	30.0 m/s
*DELTA *DELTO	$\Delta t \leq 0.7\Delta x/(2u)$ *and* $\Delta t \leq \Delta z/w$	large time step (atmos & ocean)	2.0 sec 2.0 sec
DELX DELY	Δx Δy	horiz grid point spacing	500.0 m 500.0 m

Table 1, continued.

DELZA	Δz	vert grid point spacing:top→bottom (atmos & ocean)	KK*500.0 m
*DIF2ND	$K_d^* = \frac{K_d \Delta t}{\Delta x^2} \le \frac{1}{8}$	2nd order dimensionless diffusion coef	0.025
*DIF4TH	$K_d^* = \frac{K_d \Delta t}{\Delta x^4} \le -\frac{1}{32}$	4th order dimensionless diffusion coef	-0.0025
*DIVDMP	$\alpha_d^* = \alpha_d \Delta \tau_a / \Delta x^2 \le \frac{1}{2}$	dimensionless diverg damping coef	0.05
FLAT	---	latitude	0.0°
IADVCT	(1) 4th ord advect (2) 2nd ord advect (3) upstream	temp advection scheme (atmos): (1)Klemp&Wilhelm78 (2)Klemp&Wilhelm78 (3)Haltiner&Will80	(6)
*IAHSGM	(1) deformation (2) tke equation	horiz subgrid mixing (atmos): (1)Smagorinsky63 (2)Deardorff80	(1)
*IAMXGL	(1) $l = \kappa z/(1 + \kappa z/l_{atm})$ (2) $l = \Delta s$ (3) $l = \Delta z\ l_{atm}$ (4) $l = fn(z_i, L, \gamma, e)$ (5) $l \begin{cases} = .76\sqrt{e}/N \rightarrow stable \\ = \Delta s \rightarrow unstable \end{cases}$	turb mixing length (atmos): (1)Mellor&Yamada74 (2)Klemp&Wilhelm78 (3) --- (4)Therry&Lacarr82 (5)Deardorff80	(1)
*IASHSM	(1) $S_m = fn(Ri\#)$ $S_h = fn(Ri\#)$ (2) $S_m = S_{m0}$ $S_h = S_{h0}$ (3) $S_m = S_{m0}$ $S_h = (1+2l/\Delta s) S_m$ (4) $S_m = S_{m0}$ $S_h = 1.35 S_m$	coef of K_m and K_h (atmos): (1)Mellor&Yamada74 (2)Klemp&Wilhelm78 (3)Deardorff80 (4)Therry&Lacarr82	(1)

Table 1, continued.

*IBDYA *IBDYO	(1) fixed (2) extrapolated (3) periodic (4) radiation (5) radia/extrap	x-boundary condition (5)Klemp&Wilhelm78	(1) (1)
ICASE	1 → 99	case number identifier	(1)
*ISFLX	 (1) (2)	surface flux parameterization: (1)Barker&Baxter75 (2)Louis79	(2)
*JBDYA *JBDYO	(1) fixed (2) extrapolated (3) periodic (4) radiation (5) radia/extrap	y-boundary condition (5)Klemp&Wilhelm78	(1) (1)
KK	1 → 99	total number of vertical levels	none
KATMOS	0 → 99	number of atmos levels	KK
KTAUST	0 → 9999999 sec	model beg time	0 sec
LATMOS	(logical)	atmos model	true
LCOOL	(logical)	radiative cooling	false
*LDDAMP	(logical)	divergence damping	true
*LDIFF	(logical)	diffusion	true
*LFLUX	(logical)	surface fluxes	true
LICE	(logical)	ice physics	true
*LLES	(logical)	TKE: large-eddy-simulation (atmos)	false
LLHT	(logical)	latent heating	true
LMOIST	(logical)	moist physics	true

Table 1, continued.

LOCEAN	(logical)	ocean model	false
*LRALEE	(logical)	Rayleigh damping absorbing layer	false
LRDTN	(logical)	radiation (coding in development)	false
*LSPLIT	(logical)	time splitting ($\Delta\tau$)	true
*LTKE	(logical)	subgrid scale parameterization (atmos)	true
LTWOD	(logical)	two-dimensional	true
LTHRW	(logical)	thermal wind	false
LVGEO	(logical)	geostrophic wind	false
*LWIMP	(logical)	semi-implicit vert integration	true
LWRITU	(logical)	unformatted graphics data	false
LWTOP	(logical)	vert motion at upper boundary	false
L4OADV	(logical)	4th order advection	true
M	1 → 9999	number of x-grid points	none
MAXTIM	DELTA → 9999999 sec	model end time	2400 sec
MFIELD	DELTA → 9999999 sec	time interval for saving restart data	-1 sec
*MTAUA= $\Delta t_a/\Delta\tau_a$	$\Delta\tau_a \leq 0.7\Delta x/(\bar{c}+u)$ and $\Delta\tau_a \begin{cases} \leq \frac{\Delta z}{(\bar{c}+w)} & \rightarrow LWIMP=f \\ \leq \frac{\Delta z}{w} & \rightarrow LWIMP=t \end{cases}$	number of $\Delta\tau_a$ per Δt (atmos)	5

Table 1, continued.

*MTAUO= $\Delta t_o/\Delta \tau_o$	$\Delta \tau_o \leq 0.7\Delta x/[2(\sqrt{gH}+u)]$ *and* $\Delta \tau_o \leq \Delta z/w$	number of $\Delta \tau_o$ per Δt (ocean)	50
N	0 → 9999	number of y-grid points	none
*NRDAMP	0 → KK	number of vert absorbing levels	8
*RDTIME	$\frac{1}{\alpha_R} \approx \frac{1}{3.5\,k\,\bar{u}}$	time scale for Rayleigh damping	120.0 sec
ROBERT	α_t	time-level coupling coef	0.05
*SM0	S_{m0}	coef of S_m	0.5
*SH0	S_{h0}	coef of S_h	0.675
UMEAN	---	constant wind	0.0 m/s
UGEO1D VGEO1D	u_g v_g	geostrophic wind components	KK*0.0 m/s KK*0.0 m/s

(I) **Time steps**
parameters: **CBAR, DELTA, DELTO, LWIMP, LSPLIT, MTAUA, MTAUO**

The parameters that specify the large time steps for the atmosphere **DELTA** and the ocean **DELTO** are computed using the stability criteria shown in Table 1. These large time steps depend on the advective speeds in the respective fluids. Note that both the vertical and horizontal grid resolutions impose a restriction on the large time step and both must be satisfied. To obtain Δt_a, compute a value from the first equation using an estimate of the **maximum** horizontal wind speed (or for the ocean model use an estimate of the maximum current speed). Compare that value to the time step obtained from the second equation using the **smallest** vertical grid increment and an estimate of the **maximum** vertical velocity (or vertical motion). To ensure stability, choose the smaller of the two values and truncate the digits after the decimal.

For example, suppose you are simulating the thunderstorm case shown in Appendix C. For that case, Δx=750 m and $u_{max} \approx 40$ m/s which yields an atmospheric time step of Δt_a=6.5625 s. From the second equation, we obtain a time step of Δt_a=10 s using Δz=300 m and $w_{max} \approx 30$m/s.

we obtain a time step of Δt_a=10 s using Δz=300 m and $w_{max} \approx 30$m/s. **DELTA** is therefore set to the lower of the two values and then truncated to 6 s. Typically, a Δt value that divides into 3600 s is preferable. For a coupled simulation, the parameter specifying the ocean time step **DELTO**, would need to be calculated as well. That value is obtained from the same stability criterion as is **DELTA** but using values for u_{max}, w_{max}, and Δz applicable to the ocean.

In a nonhydrostatic atmospheric simulation, sound and gravity waves are integrated over shorter time steps to maintain stability. These integrations are carried out with the time-splitting procedure described briefly in Section 2c1 when the logical parameter **LSPLIT** to set to true. Time-splitting is also used for integration of the barotropic mode in the ocean model. The smaller atmospheric and oceanic time steps are specified by parameters **MTAUA** and **MTAUO**, which indicate the number of small iterations per large time step iteration. These parameters are obtained by first computing a small time step $\Delta\tau_a$ for the atmosphere and $\Delta\tau_o$ for the ocean from the stability criterions shown in Table 1, in the same manner as that described above for Δt. Then the number of small iterations is determined by dividing $\Delta t_a/\Delta\tau_a$=**MTAUA** and $\Delta t_o/\Delta\tau_o$=**MTAUO**, rounding up to the next integer value.

Note that the small atmospheric time step also depends on the speed of the most rapidly propagating mode given by the speed of sound $\bar{c}$, specified in the model by the parameter **CBAR** (see Table 1). A typical value for $\bar{c}$ is approximately 330 m/s; however, a quasi-compressible simulation can be run for which smaller $\bar{c}$ values are acceptable provided that $\bar{c}$ satisfies the $\Delta z/\Delta\tau_a$ relation shown in the table. As a general rule, the value chosen for **CBAR** should be at least twice the fastest advective speed.

When semi-implicit vertical integration is enabled, the vertical dependance on $\bar{c}$ is eliminated. This allows for a more efficient model run since the vertically imposed stability condition on $\Delta\tau_a$ is less restrictive. To specify semi-implicit vertical integration set the logical variable **LWIMP** to true and obtain a small time step based on comparison of $\Delta\tau_a$ from the horizontally imposed stability restriction with the $\Delta\tau_a$ from the ratio $\Delta z_{min}/w_{max}$.

Using the thunderstorm case (Appendix C) as an example, we demonstrate how to compute the number of small time step iterations. From the appendix, we note that **LSPLIT** is true, **LWIMP** is false, and **CBAR** is the default value, 100 m/s. The stability criterion given by the horizontal grid spacing yields a time step of $\Delta\tau_a$=3.75 m/s while that from the vertical spacing gives a more restrictive value of 2.31 s. Therefore, the small atmospheric time step $\Delta\tau_a$ is truncated to 2 s, and **MTAUA** = 6/2 = 3 iterations. For a coupled simulation, similar steps are also taken to obtain the small oceanic time step $\Delta\tau_o$, which depends on the gravity wave speed $\sqrt{gH}$ where H is the depth of the fluid. That value is then used to

determine the number **MTAUO** of small oceanic iterations per large oceanic time step.

(II) **Turbulent kinetic energy: LES and Non-LES simulations**
parameters: **AL0, IAHSGM, IAMXGL, IASHSM, LLES, LTKE, SH0, SM0**

When performing a large-eddy-simulation (LES) special consideration must be given to specifying the subgrid scale turbulence parameterization. First, enable subgrid scale mixing by setting the logical variable **LTKE** to true. Second, set the logical variable **LLES** to true which flags certain LES coefficients, terms and calculations in the TKE equation. Third, choose appropriate options for the mixing length **IAMXGL** and **AL0**, and the turbulence coefficients **IASHSM**, **SH0** and **SM0**, that are defined in Table 1. An important consideration here is the model's sensitivity to these parameters. Certain definitions for the mixing length should be paired with corresponding definitions for the turbulence coefficients. For example, Deardorff (1979) uses **IAMXGL**=5 and **IASHSM**=3 with **SM0**=0.1 for his three-dimensional LES study of turbulent mixed layers. Mellor and Yamada (1974) use **IAMXGL**=1 (where **AL0** is dependant upon the ratio of two vertical TKE integrals) and **IASHSM**=1 to represent one-dimensional vertical mixing in their level-2 turbulence closure model.

For non-LES modeling, many of the steps listed above are repeated but with different options. As before, set the logical variables: **LTKE**=true and **LLES**=false. Then choose appropriate values for the parameters **IAMXGL**, **AL0**, **IASHSM**, **SH0** and **SM0**. Although non-LES simulations are less sensitive to the subgrid scale mixing, caution is still advised when specifying these values. Two types of non-LES parameterizations are coded in the model. One type is formulated after that of Klemp and Wilhelmson's (1978) three-dimensional study of a convective storm. They use options **IAMXGL**=2 and **IASHSM**=2, and coefficients **SM0**=0.2 and **SH0**=0.6. The second, more sophisticated boundary layer parameterization, derived by Therry and Lacarrère (1982), is based upon empirical relationships and experimental data for which **IAMXGL**=4 and **IASHSM**=4 with **SM0**=0.5.

(III) **Numerical diffusion**
parameters: **DIF2ND, DIF4TH, LDIFF**

During a **COAMPS** run, diffusion is needed to damp the accumulation of high frequency oscillations. The amount of diffusion is specified by the dimensionless coefficients **DIF2ND** and **DIF4TH**, which vary from case to case depending on both the spatial and temporal model increments. Simulations may be run without diffusion by setting the logical variable **LDIFF** to false; however, if left unchecked, numerical instabilities will likely amplify, nonlinearly contaminate the larger scales, and may eventually cause the model to crash. Either second- or fourth-order diffusion may

be applied to the dependent variables. In general, fourth-order diffusion significantly reduces scales $4\Delta x$ and less while maintaining the amplitude of the larger modes. The coefficient **DIF4TH** must be given a negative value in order to implement fourth-order diffusion. The maximum magnitude for **DIF4TH** may be computed from the expression given in Table 1, using the stability limit of -0.03125 to obtain a value for K_d. The actual value used for K_d however, is typically several orders of magnitude smaller and is chosen somewhat arbitrarily.

Second-order diffusion tends to damp and smooth the model solutions to a greater extent. For nonperiodic boundaries, second-order diffusion is always used to compute the smoothing at the first interior grid point. It may also be specified for diffusing the solutions at the remaining grid points by using a non-negative or zero value for **DIF4TH**. In that case, the coefficient **DIF2ND** regulates the amount of diffusion. As for fourth-order diffusion, this parameter may be adjusted for each case based on space and time scales. Its value must remain below the stability threshold of 0.125.

(IV) **Lateral boundary conditions**
parameters: CSTAR, IBDYA, IBDYO, JBDYA, JBDYO

Several options are available for designating boundary conditions at the east/west and north/south domain edges. East/west boundaries are indicated by the parameter **IBDYA** in the atmosphere and by **IBDYO** in the ocean. North/south boundaries are similarly specified by **JBDYA** for the atmosphere and **JBDYO** for the ocean. As shown in Table 1, options (1), (2), and (3) are straightforward and no further discussion on those conditions is presented. Instead, we refer the user to the model code for further detail (see Figures 2b and 2c for file names). More complicated radiative boundary conditions are given by options (4) and (5), described by Klemp and Wilhelmson (1978). The radiative condition allows the dominant gravity wave modes to move out through the domain with minimal reflection. The outward propagation of the gravity waves is governed by c_* defined in the model by the parameter **CSTAR** (see Table 1). The radiation condition advects the normal component through the boundary at a speed determined by (u_i+c_*) via the equation

$$\frac{\partial u_i}{\partial t} = -(u_i+c_*)\frac{\partial u_i}{\partial x_i} + TERMS_i \quad , \qquad (30)$$

where i=1,2 and $u_1=u$, $u_2=v$. In the above equation, *TERMS* represents the effects of Coriolis forcing, Rayleigh damping and subgrid scale mixing evaluated at the boundaries. For option (4), subgrid scale mixing is neglected at the lateral boundaries. Equation (30) holds for both outflow and inflow with the following exception: $(u_i+c_*)=0$

for inflow. Additionally, option (5) stipulates that boundary values for all of the other prognostic variables are extrapolated. Option (4) determines those boundary values by assuming that the normal derivatives are zero.

(V) **Rayleigh damping absorbing layer**
parameters: **LRALEE, NRDAMP, RDTIME**

To reduce reflection off of the upper boundary and the subsequent amplification of vertically propagating sound and gravity waves, the user can specify an absorbing layer at the top of the model domain. This is accomplished by setting the logical parameter **LRALEE** to true. The number of vertical levels in which absorption occurs is given by the integer variable **NRDAMP**, and the total depth of the absorbing layer should equal approximately one vertical wavelength. The decay time scale **RDTIME** is chosen such that the dominant horizontal wavelength is damped most efficiently. The expression in Table 1 may be used to calculate an appropriate value for this parameter.

(VI) **Divergence damping**
parameters: **DIVDMP, LDDAMP**

An acoustic mode filter has been added to **COAMPS** that diffuses divergence and reduces the accumulation of mass within a grid volume. Since nonhydrostatic simulations tend to excite high-frequency sound waves, the divergence damping flag **LDDAMP** is generally set to true. The dimensionless damping coefficient **DIVDMP** may be calculated from the expression given in Table 1. Although this parameter should be less than 0.5 for stability considerations, the default value is an order of magnitude smaller and has been chosen arbitrarily.

(VII) **Surface fluxes**
parameters: **ISFLX, LFLUX**

Adequate surface layer representation permits information at the lower atmospheric boundary, and upper oceanic boundary, to be transferred to adjacent model layers. It is also responsible for the coupling between the two fluids. **COAMPS** computes the parameterized surface fluxes when the logical parameter **LFLUX** is set to true. Two types of surface layer schemes have been incorporated into the model code. Both give similar results when simulating a heated planetary boundary layer. The first scheme, specified by setting **ISFLX**=1, is described in a paper by Barker and Baxter (1975), and the second, formulated by Louis (1979), is denoted by the option **ISFLX**=2.

4. Code

Performing a modeling study using **COAMPS** consists of running two executables: (1) the analysis code, called *coama*, containing the initialization of model variables, background fields, and other case specific parameters, and (2) the model code, called *coamm*, containing the time integration loops in which various calls are made to the atmospheric and oceanic model subroutines. Figure 2a-d shows a simplified flow chart of a **COAMPS** model execution including the important subroutine names and their function within the system.

In Subsection 4a the details of the analysis code, including how to specify case dependent initial conditions, are described. The details of the model code are described in Subsection 4b.

a. The analysis code

The main function of the analysis code is to provide the model with a set of initial conditions. Many of the constants, parameters and arrays required for a model run are assigned values in the main analysis program *coama.f*. This program calls a subroutine named *flds#.f* that must be provided by the user. It contains the case specific parameters, prognostic model variables and background profiles at the model's start time=**KTAUST**. The '#' in this filename is a case number identifier and corresponds to the parameter **ICASE** in the namelist input (See Table 1). We describe the two primary analysis files, *coama.f* and *flds#.f*, below.

Initially, *coama.f* reads a set of include files *nhcd0#.h* that contain dimension, declaration and common block statements for many of the **COAMPS** parameters and arrays. The namelist **/nonhyd/** is embedded within *nhcd04.h* (See Appendix B). It permits the transfer of user input from the script file to the executables. Secondly in *coama.f*, constants are assigned values and loop indices are specified. The latter is important for partitioning the vertical levels into top and bottom indices for the atmospheric and oceanic models. This is done using the total number of levels **KK**, and the number of atmospheric levels **KATMOS**, both specified by the user. From the partitioned vertical indices and the **DELZA** input read through the namelist, two sets of heights arrays, $\sigma_m(k)$ and $\sigma_w(k)$, are computed (k is the vertical array index). As shown in Figure 1, the **DELZA** values are the spacing between each $\sigma_w(k)$ height level, and k=1 is the first model level from the top. The $\sigma_w(k)$ heights correspond to vertical velocity grid points, while the $\sigma_m(k)$ heights are staggered upward by ½ grid interval and correspond to grid point locations of the remaining model variables.

The *coama.f* and *makefile.coama* files must both be edited to include

references to the user's initial conditions file *flds#.f*. When creating a *flds#.f* file, we recommend copying and modifying the *fldsample.f* file provided with the **COAMPS** code. This file contains the initialization of the following arrays (array names are shown in parentheses):

- 3D model variables (the number in the array names corresponds to a time level; number 2 indicates the current time level)
  - mixing ratios (*qc2, qi2, qr2, qs2, qv2*)
  - pressure (*p2*)
  - potential temperature (*th2*)
  - salinity (*s2*)
  - ocean temperature (*t2*)
  - atmospheric turbulent kinetic energy (*e2*)
  - velocity components/ocean currents (*u2, v2, w2*)
- 2D background profiles
  - density (*rbarm*)
  - mixing ratio (*qvbarm*)
  - pressure (*prbarm*)
  - salinity (*sbarm*)
  - temperatures (*tbarm, tvbarm, thbarm, thvbarm*)
- 2D surface variables
  - temperature (*tsea*)
  - mixing ratio (*qvsea*)
  - roughness (*z0*)
  - land=1/water=0 indicator (*xland*)
  - albedo (*albedo*)
  - ground wetness (*grdwet*)

Presently, the three-dimensional model arrays are indexed (i,k,j); however, this will soon be changed to (i,j,k) for use of **COAMPS** with real data, nested grids and multi-tasking. The background profiles are indexed (i,k), but typically vary only with height. The surface variables are indexed (i,j). The above arrays are specified by the user from a skew-T sounding, a published study, a data base, or any other available source. They are then interpolated to the model heights given by the $\sigma_m(k)$ array (See Figure 1a). An iterative technique is used to hydrostatically balance the background pressure with the temperature and moisture fields.

After calling *flds#.f*, *coama.f* calls subroutine *nhio.f* that writes out the initial conditions to an unformatted data file named *D#######*, where the '#'s correspond to the time in seconds. The analyzed data provides input to the **COAMPS** model run which is discussed in the next section.

b. The model code

The main model program *coamm.f* is the backbone of **COAMPS**. It utilizes the data given by the analysis code to integrate the prognostic variables between the user specified beginning and end times, **KTAUST** to **MAXTIM**. Throughout the integration, model output and graphics data are created at users specified time intervals. In Section 6, information pertaining to the model output is provided. The structure and details of the *coamm.f* main program are given below.

As shown in Figures 2c and 2d, *coamm.f* contains three integration loops. A primary integration loop 100 to leapfrog over a large time step, and embedded within it, two secondary loops 200 that integrate over the small time-splitting steps of the atmosphere and ocean. Before loop 100 begins, a series of calls are made to set up the model run (see Fig. 2a). These subroutines and lines of code perform the following necessary steps: 1) read the include files *nhcd0#.h*, 2) read the analysis data from the *D#######* file, 3) initialize other model arrays, 4) assign loop indices, 5) compute model runtime variables, flags and coefficients, 6) assign values to constants, and 7) define output/graphics parameters (see Fig. 2b).

Next, the primary forecast loop 100 begins. After the first forward time step, each pass through loop 100 performs a leap frog integration based upon the smaller of the two large time steps, Δt_a or Δt_o. The code within this loop is divided into two sections, one for the atmosphere and one for the ocean. Each section is flagged by a conditional that skips around the code when appropriate. For example, the ocean section is skipped 1) entirely during an atmospheric model run only, or 2) for several iterations during a coupled run (because typically the atmospheric time step is smaller than the oceanic time step). Similarly, the atmospheric section is skipped entirely during an oceanic model run only. For a large time step integration, the general structure of both the atmospheric and ocean sections is as follows:

```
-- shift time levels of prognostic variables
   (time levels correspond to the numbers in the variable
     array names (eg. u1,u2,u3) where:  1-previous value,
     2-current value, 3-predicted value)
-- compute surface fluxes
-- compute diagnostic variables
-- compute right-hand-sides of prognostic variables
-- perform small time step integrations
   (obtain left-hand-sides of prognostic variables;
    see loop 200 below)
-- update prognostic variables
-- implicitly mix prognostic variables
-- apply boundary conditions for prognostic variables
```

-- adjustments to prognostic variables
(ie. microphysical and radiation adjustments
to the temperature and moisture fields)
-- correct boundaries of prognostic variables
-- time filter prognostic variables
-- call printout, graphics, and statistics files

Refer to Figures 2c and 2d for a flow chart of the main model loop 100. At the end of loop 100 a restart data file is created (*D#######*) by subroutine *nhio.f* at user specified time intervals given by namelist parameter **MFIELD**. This capability allows the user to continue a forecast beyond the user specified **MAXTIM** or to restart the model from an earlier time (after having made changes to the output and graphics fields for example).

Within each of the atmospheric and ocean sections, an additional loop 200 integrates separately the model terms and equations that include the faster moving sound and gravity waves. These integrations are performed with a small forward time step $\Delta\tau_a$ for the atmosphere and a small centered time step $\Delta\tau_o$ for the ocean. Each loop is based upon the following structure:

-- compute left-hand-sides of prognostic variables
-- update prognostic variables
-- apply boundary conditions of prognostic variables.

Additionally within the oceanic loop 200, variables representing the barotropic responses $\overline{u}$ and $\overline{v}$, and the height of the free surface h, are time filtered and their time levels are shifted because those prognostic equations are computed entirely within the $\Delta\tau_o$ loop. Figures 2c and 2d show more complete diagrams of the small time step loops. In the next section, the script file that compiles, links and runs the **COAMPS** analysis and model code is discussed.

(2a) SETUP ROUTINES

```
nhcd01.h - specify grid dimensions (M,KK,N)
nhcd02.h_ specify variable and array dimension, declaration and
nhcd03.h   common block statements
                        specify default variable values
           nhcd04.h -   read user-defined values from /nonhyd/
                          (See Appendix B)
namel.f  - nhio.f - open and read initial conditions file
                    D####### (#s are the time in seconds)
           assign loop indices
           compute model runtime variables, flags and coefficients
           assign values to constants
           setup graphics parameters
setup.f  - assign vertical loop indices for the atmosphere/ocean
asetup.f - initialize arrays for atmospheric prognostic variables
             at each time level
osetup.f - initialize arrays for oceanic prognostic variables at
             each time level
CODE     _ setup graphics
           DATA ROUTINES (below) - specify output and graphics
```

(2b) DATA ROUTINES

```
aprt2d.f, oprt2d.f - print out 2D model data
aprt3d.f, oprt3d.f - print out 3D model data
asavxz.f, osavxz.f - send xz-plot data to graphics files
asavxy.f, osavxy.f - send xy-plot data to graphics files
asav1d.f, osav1d.f - send 1D-plot data to graphics files
stat2d.f, stat3d.f - compute and print out atmospheric statistics
atraj.f,  otraj.f  - compute trajectories
```

Figure 2: Outline flow chart of the main model routine *coamm.f*, including the important subroutines and their functions: (a) setup subroutines (b) data subroutines (c) atmospheric section (d) ocean section. See Appendix A for the variable definitions.

(2c) ATMOSPHERIC SECTION

COAMM.F (main model routine)

```
SETUP ROUTINES (see Figure 2a) - initialize model
LOOP 100 - large time step integration (Δt_a or Δt_o)
  ATMOSPHERIC SECTION
    setup.f   - assign vertical loop indices for the atmosphere
    CODE      - if check; atmospheric iteration flag (kmoda)
    ashift.f  - shift time levels for prognostic variables
                  (u, v, w, e, π, θ, q_v, q_c, q_r, q_s, q_i)
    adiv3d.f  - compute three dimensional divergence
    bwind.f   - compute wind at the horizontal boundaries
    sfcflx.f  - compute surface variables (u_*, θ_*, q_*)
    adiagk.f  - diagnose horizontal (K_m, K_h); for iahsgm=1 only
    afore.f   - predict (e)
    aforth.f ┐
    aforqv.f │
    aforqc.f ├ predict (θ, q_v, q_c, q_r, q_s, q_i)
    aforqf.f │
    aforqs.f │
    aforqi.f ┘
    amixtq.f  - implicitly mix (θ, q_v, q_c, q_r, q_s, q_i)
    abdye.f   - correct boundaries (e)
    abdytq.f  - correct boundaries (θ, q_v, q_c, q_r, q_s, q_i)
    arhsu.f  ┐
    arhsv.f  ├ - predict RHS (u, v, w)
    arhsw.f  ┘
    arhsp.f   - predict RHS (π)
    LOOP 200 - small time step integration (Δτ_a)
      ¦ alhsw.f  - predict LHS (u, v, w)
      ¦ uplhs1.f - update (u, v, w)
      ¦ abdyuv.f - correct boundaries (u, v, w)
      ¦ alhsp.f  - predict LHS (π)
      ¦ uplhs2.f - update (π)
      ↓ abdyp.f  - correct boundaries (π)
    END LOOP 200
    amixuv.f ┐ implicitly mix (u, v, w)
    amixw.f  ┘
    abdyuv.f  - correct boundaries (u, v, w)
    adjtq.f   - microphysical adjustments (θ, q_v, q_c, q_r, q_s, q_i)
    radiat.f  - radiation adjustments to (θ)
    abdytq.f  - correct boundaries (θ, q_v, q_c, q_r, q_s, q_i)
    atimef.f  - apply time filter (u, v, w, e, π, θ, q_v, q_c, q_r, q_s, q_i)
    DATA ROUTINES (see Figure 2b) - specify output and graphics
  END ATMOSPHERIC SECTION
  CONTINUE LOOP 100
```

(2d) OCEANIC SECTION

```
↓ CONTINUE LOOP 100 - large time step integration (Δt_a or Δt_o)
  OCEANIC SECTION
    setup.f  - assign vertical loop indices for the ocean
    CODE     - if check; oceanic iteration flag (kmodo)
    oshift.f - shift time levels for prognostic variables
                 (u, v, w, T, S, ρ)
    odiagp.f - diagnose (p)
    odiage.f - diagnose (e)
    ofort.f  - predict (T)
    ofors.f  - predict (S)
    omixst.f - implicitly mix (T, S)
    obdyts.f - correct boundaries (T, S)
    orhsu.f  - predict (u, v)
    orhsv.f
    omixuv.f - implicitly mix (u, v)
    odiagr.f - diagnose (ρ)
    obaro.f  - update (u', v')
               obdyuv.f - correct boundaries (u', v', T, S)
               otimef.f - apply time filter (u', v', T, S)
               LOOP 200 - small time step integration (Δτ_o)
                 CODE   - shift time levels (ū, v̄, h)
                          predict (ū, v̄, h)
                          update (ū, v̄, h)
                          correct boundaries (ū, v̄, h)
                          apply time filter (ū, v̄, h)
                 ↓
               END LOOP 200
               update (u, v)
    odiagw.f - diagnose (w)
    DATA ROUTINES (see Figure 2b) - specify output and graphics
  END OCEANIC SECTION
↓ nhio.f   - write model forecast data to file D#######
END LOOP 100
```

```
END COAMM.F
```

5. Execution: the script file

In order to run **COAMPS**, first the executables coama and coamm must be created by compiling and linking the source code. However, before the executables are made, the user must create three files: 1) a *flds#.f* file as described in Section 4a, 2) the model grid dimensions (**M,KK,N**) located in the include file *nhcd01.h*, and 3) an input file named *case#.data* containing the namelist *nonhyd*. Recall that *nonhyd* assigns the case specific model parameters that are defined by the user. The latter two files, *nhcd01.h* and *case#.data*, may be contained within a single script file named *case#.cc*, that not only creates them but also compiles and links the source code, and then runs the executables. A flow chart outlining the steps performed in the script file is provided in Figure 3.

A complete **COAMPS** execution is performed by submitting *case#.cc* as a batch process which is accomplished by typing the command '*qsub case#.cc*' on the CRAY. It is beneficial to perform each of the steps in the script file interactively in the following situations: when debugging the code or during the initial phases of a model run. This is suggested since preliminary, debugging model runs should be small and of short duration, and therefore are more efficiently processed interactively. In addition to Figure 3, a specific example of a script file, *case8.cc* created for the execution of a convective thunderstorm simulation on the CRAY, is given in Appendix C. In the following paragraphs, we discuss each of the steps performed by the *case8.cc* script file

Referring to Appendix C, any line beginning with a '*#*' is a comment statement except those containing a '*#@$-*' which are batch processing commands. The first of these commands '*q lsc*' indicates the system queue in which the job will run. The second '*lT 01:30:10*' and third '*lt 01:30:00*' indicate the maximum cpu time in hours, minutes and seconds for the entire process and for each subprocess. Similarly, the fourth and fifth commands indicate the maximum space requirements in megawords for the entire process and each subprocess. The user may wish to take advantage of the UNIX command '*size*' to determine the space requirements for a given executable. Time requirements are determined by running a short forecast, and noting the cpu time reported in the output file. That value is then used to gauge the cpu time required for a longer forecast. The sixth command '*eo*' combines the standard error with the standard output into one output file with the extension '*.log*'. This file is mentioned further in Section 6. The seventh command '*s /bin/sh*' indicates that the script will run in the bourne shell. Typically, the user will only change the second-fifth batch processing commands, and need only concern themselves with the values of '*lT*' and '*lM*' since the subprocess limits may be set to an equivalent or nearly equivalent value.

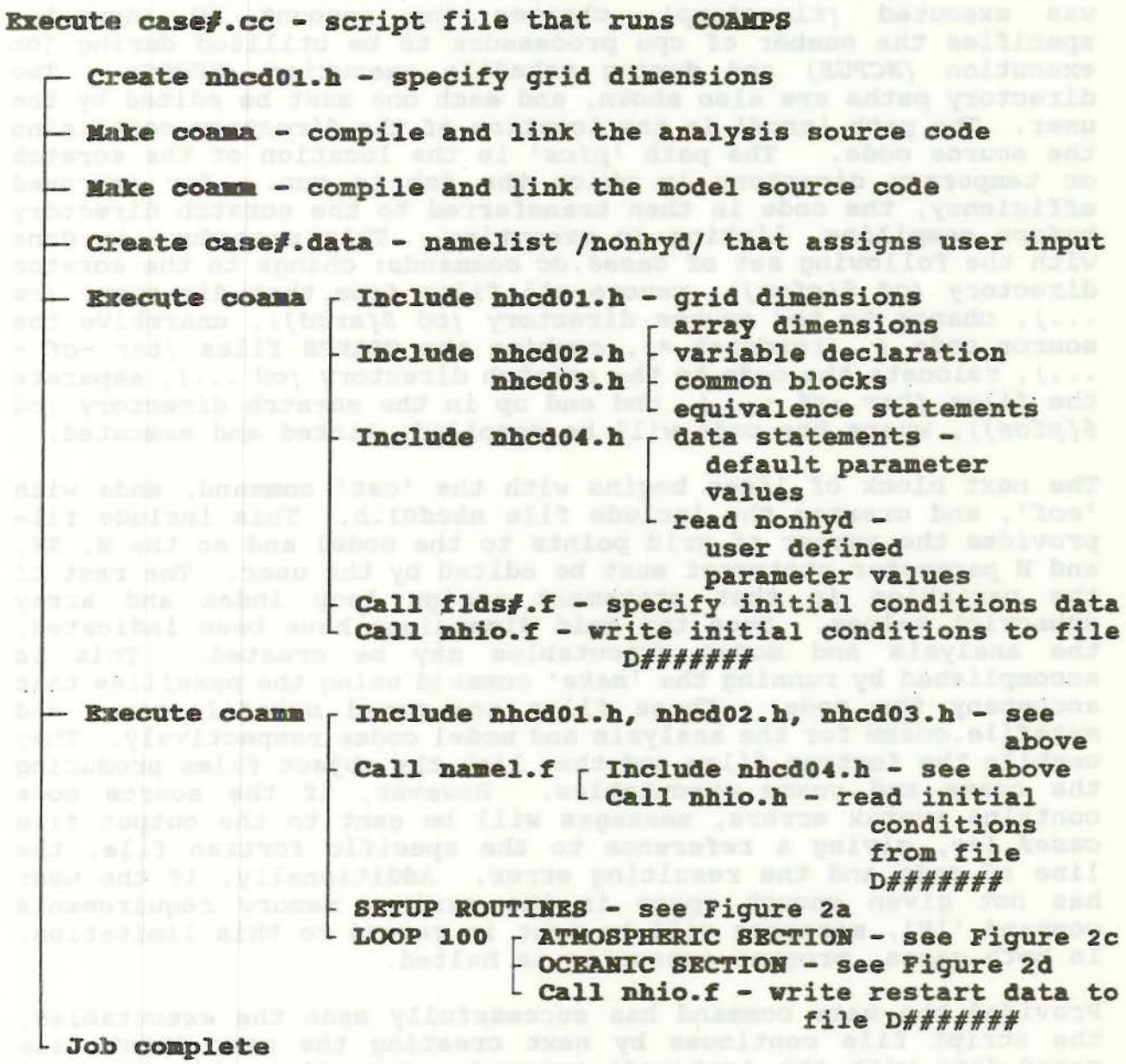

Figure 3: Outline flow chart of the script file *case#.cc* that performs each of the steps necessary for a COAMPS simulation.

The next set of lines in *case8.cc* do the following: echo to standard output all script file arguments *(echo)*, and the time each was executed *(timestamp)*, changes the account ID *(newact)*, specifies the number of cpu processors to be utilized during job execution *(NCPUS)* and during makefile execution *(NPROC)*. Two directory paths are also shown, and each one must be edited by the user. The path '*srcd*' is the location of the directory containing the source code. The path '*pfcs*' is the location of the scratch or temporary directory in which the job is run. For improved efficiency, the code is then transferred to the scratch directory before compiling, linking or executing. This procedure is done with the following set of *case8.cc* commands: change to the scratch directory *(cd ${pfcs})*, remove all files from that directory *(rm ...)*, change to the source directory *(cd ${srcd})*, unarchive the source code *(...remdmget *)*, combine the **COAMPS** files *(tar -cf - ...)*, relocate the code to the scratch directory *(cd ...)*, separate the files *(tar -xf -...)*, and end up in the scratch directory *(cd ${pfcs})*, where the code will be compiled, linked and executed.

The next block of lines begins with the '*cat*' command, ends with '*eof*', and creates the include file *nhcd01.h*. This include file provides the number of grid points to the model and so the **M**, **KK**, and **N** *parameter statement* must be edited by the user. The rest of the variables in that statement assign loop index and array subscript values. Once the grid dimensions have been indicated, the analysis and model executables may be created. This is accomplished by running the '*make*' command using the makefiles that accompany the code. These files are named *makefile.coama* and *makefile.coamm* for the analysis and model codes respectively. They compile the fortran files and then link the object files producing the *coama* and *coamm* executables. However, if the source code contains syntax errors, messages will be sent to the output file *case#.log*, giving a reference to the specific fortran file, the line of code and the resulting error. Additionally, if the user has not given enough space in the maximum memory requirements command '*lM*', messages will be sent in regard to this limitation. In both cases, program execution is halted.

Provided the *make* command has successfully made the executables, the script file continues by next creating the user input file *case8.data* with the '*cat/eof*' command. Note that this block of script is labeled **nonhyd**, and that each line assigns a value to a model input variable. The variable names must be among those listed in the namelist **/nonhyd/** in Appendix B for the proper transfer of user input to the model. An input variable is assigned a default value from the include file *nhcd04.h*, which may then be overwritten by the user defined value from the input file *case8.data*. The user's ability to make changes to these model parameters, without having to modify, recompile and relink the fortran code, provides for a more efficient modeling study. For reference to each of the namelist variables see Table 1 in Section 3 and Table 2 in Section 6.

At this point, all of the necessary input files have been created and the executables are ready to run. The last few lines in *case8.cc* perform this task. The '*ja*' commands turn on and off job accounting producing a detailed report of space and time usage by the executable. The command '*time*' reports the elapsed time, wall time and cpu time during execution. Should runtime errors occur, the script file will be halted and a short error message will be written to the output *case8.log* file. A typical one reads '*floating point exception*' which may indicate an out-of-bounds array, misaligned subroutine arguments, division by zero, etc. The user will need to isolate the location of the error by inserting print statements in the code, by using a fortran debugging program or by setting some of the logical variables to false. In the next section, we briefly review the contents of the '*.log*' output file, as well as the other forms of output and data.

6. Output/graphics

Analyzing model data and viewing graphical output is useful for monitoring **COAMPS** progress, debugging code, or comparing the results of various simulations. Several features have been added to the **COAMPS** model to produce user specified output. Each form of data will be reviewed briefly.

When the batch job has finished executing, a *case#.log* output file is produced by the *case#.cc* script. It will reside in the same directory from which the script was submitted. This output file contains all of the script file commands and arguments, including job accounting information, as well as the standard error and standard output that is produced by the two executables *coama* and *coamm*. The standard output is generated by the usual fortran write statements that are inserted into the code or contained in the various print subroutines (eg. *aprt3d.f*, *oprt3d.f* in Figure 2b). At time intervals specified by values of **MPRNTA** and **MPRNTO** (see Table 2), the print routines write out data for predetermined model variables at predetermined cross-section and horizontal level locations. The user may wish to tailor these output routines to printout locations and model fields relevant to their particular study. The *case#.log* files primary function is to provide a quick look at a simulation's progress.

For graphical display of model output, several other forms of data are written to files that are then used as input to NCAR graphics programs used for viewing plots and graphs. For example, cross-section data are written out by the subroutines *asavxz.f* and *osavxz.f*. Horizontal level data are written out by the subroutines *asavxy.f* and *osavxy.f*. One-dimensional grid point data are written out by the subroutines *asav1d.f* and *osav1d.f*. Trajectory data are written out by the main program *coamm.f*, and subroutines *atraj.f*, and *otraj.f*. The type of model output, as well as the variables, grid point locations, start and end times, and time intervals for which data files are created may be specified by the user. This is done by assigning values to the appropriate user input parameters, defined in Table 2 below and shown in the sample script file in Appendix C. For more information on creating one-dimensional data and for viewing and printing one-dimensional plots see the technical note *Graphical Display Procedure for Grid Point Model Data*. A technical note describing the two-dimensional graphics will be forthcoming.

Table 2: Output/graphics parameters
(See Appendix A for variable definitions)

PARAM	DEFINITION (TYPE)	DESCRIPTION	DEFAULT
CPATHI	(character*80)	path of model data input/output files (D#######)	' '
CPATHO	(character*80)	path of graphics data output files	' '
C1DFLD	(character*4 array)	1D-plot field names	' '
I1D	1 → M	x-grid point for 1D-plot files	1
IDADJ	1 → M	x-grid point for adjtq.f diagnostic files	21
IDAFE	1 → M	x-grid point for afore.f diagnostic files	21
IXYLVA IXYLVO	1 → KK (array)	levels for xy-plot files	0 0
IXZIEA IXZIEO	1 → M (array)	end x-grid points for xz-plot files	-1 -1
IXZISA IXZISO	1 → M (array)	begin x-grid points for xz-plot files	-1 -1
IXZJEA IXZJEO	1 → N (array)	end y-grid points for xz-plot files	-1 -1
IXZJSA IXZJSO	1 → N (array)	begin y-grid points for xz-plot files	-1 -1
J1D	1 → N	y-grid point for 1D-plot files	1
JDGADJ	1 → N	y-grid point for adjtq.f diagnostic files	1
JDGAFE	1 → N	y-grid point for afore.f diagnostic files	1

Table 2, continued.

MESSG	(character*80)	case identifying message	' '
MPRNTA MPRNTO	0 → 9999999 sec	time interval for printing model output to file *case#.log*	999999 sec 999999 sec
MSTATA	0 → 9999999 sec	time interval for printing statistics to file *case#.log*	600 sec
T1D	0 → 9999999 sec	time interval for creating 1D-plot files	100000 sec
TJTEA TJTEO	0 → 9999999 sec (array)	end times for trajectory files	0 sec 0 sec
TJTIA TJTIO	0 → 9999999 sec (array)	time intervals for creating trajectory files	0 sec 0 sec
TJTSA TJTAO	0 → 9999999 sec	begin times for trajectory files	0 sec 0 sec
TJXA0 TJXO0	1 → M (array)	starting x-grid points for trajectory files	1.0 1.0
TJYA0 TJYO0	1 → N (array)	starting y-grid points for trajectory files	1.0 1.0
TJZA0 TJZO0	1 → KK (array)	starting height levels for trajectory files	1.0 1.0
XYFLDA XYFLDO	(character*4 array)	xy-plot field names	' ' ' '
XYTEA XYTEO	0 → 9999999 sec (array)	end times for xy-plot files	0.0 sec 0.0 sec
XYTIA XYTIO	0 → 9999999 sec (array)	time intervals for creating xy-plot files	0.0 sec 0.0 sec

Table 2, continued.

XYTSA XYTSO	0 → 9999999 sec (array)	begin times for xy-plot files	0.0 sec 0.0 sec
XZFLDA XZFLDO	(character*4 array)	xz-plot field names	' ' ' '
XZTEA XZTEO	0 → 9999999 sec (array)	end times for xz-plot files	0.0 sec 0.0 sec
XZTIA XZTIO	0 → 9999999 sec (array)	time intervals for creating xz-plot files	0.0 sec 0.0 sec
XZTSA XZTSO	0 → 9999999 sec (array)	begin times for xz-plot files	0.0 sec 0.0 sec

References

Arakawa, A. and V.R. Lamb, 1974: *Computational design of the UCLA general circulation model*. Methods in Computational Physics, Vol. 17, Academic Press, 173-265.

Barker, E.H. and T.L. Baxter, 1975: A note on the computation of atmospheric surface layer fluxes for use in numerical modeling. *J. Appl. Meteor.*, **14**, 620-622.

Businger, J.A., J.C. Wyngaard, Y. Izumi, and E.F. Bradley, 1971: Flux-profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, **28**, 181-189.

Clancy, R.M. and K.D. Pollak, 1983: A real-time synoptic ocean thermal analysis/forecast system. *Prog. Oceanog.*, **12**, 383-424.

Chang, S.W., 1985: Deep ocean response to hurricanes as revealed by an ocean model with free surface. Part I: Axisymmetric case. *J Phys. Oceanogr.*, **15**, 1847-1858.

Deardorff, J.W., 1972: Numerical investigation of neutral and unstable planetary boundary layers. *J. Atmos. Sci.*, **29**, 91-115.

Deardorff, J.W., 1980: Stratocumulus-capped mixed layers derived from a three-dimensional model. *Bound. Layer Meteor.*, **18**, 495-527.

Dudhia, J., 1993: A nonhydrostatic version of the Penn State-NCAR mesoscale model: validation tests and simulation of an Atlantic cyclone and cold front. *Mon. Wea. Rev.*, **121**, 1493-1513.

Durran D.R. and J.B. Klemp, 1983: A compressible model for the simulation of moist mountain waves. *Mon. Wea. Rev.*, **111**, 2341-2361.

Gill, A.E., 1982: *Atmosphere-ocean dynamics*. Academic Press, 662 pp.

Haltiner, G.J. and R.T. Williams, 1980: *Numerical prediction and dynamic meteorology*. John Wiley & Sons, Inc. 477 pp.

Ikawa, M., 1988: Comparison of some schemes for nonhydrostatic models with orography. *J. Meteor. Soc. Japan*, **66**, 753-776.

Klemp, J.B. and D.K. Lilly, 1978: Numerical simulation of hydrostatic mountain waves. *J. Atmos. Sci.*, **35**, 78-107.

Klemp, J.B. and R.B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

Louis, J.F., 1979: A parametric model of vertical eddy fluxes in the atmosphere. *Bound. Layer Meteor.*, **17**, 187-202.

Mellor, G.L. and T. Yamada, 1974: A hierarchy of turbulence closure models for planetary boundary layers. *J. Atmos. Sci.*, **31**, 1791-1806.

Robert, A.J., 1966: The investigation of a low order spectral form of the primitive meteorological equations. *J. Meteor. Soc. Japan*, **44**, 237-245.

Rutledge S.A. and P.V. Hobbs, 1983: The mesoscale and microscale structure of organization of clouds and precipitation in midlatitude cyclones. VIII: A model for the "seeder-feeder" in warm-frontal rainbands. *J. Atmos. Sci.*, **40**, 1185-1206.

Shamarock, W.C. and J.B. Klemp, 1992: The stability of time-split numerical methods for the hydrostatic and the nonhydrostatic elastic equations. *Mon. Wea. Rev.*, **120**, 2109-2127.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations: 1. The basic experiment. *Mon. Wea. Rev.*, **91**, 99-164.

Therry, G. and T. LaCarr,re, 1983: Improving the eddy kinetic energy model for planetary boundary layer description. *Bound. Layer Meteor.*, **25**, 63-88.

Yamada T. and G. Mellor, 1975: A simulation of the Wangara atmospheric boundary layer data. *J. Atmos. Sci.*, **32**, 2309-2329.

Appendix A: Definitions of variables. The number in parentheses is the equation number in which the variable appears.

VAR	DEFINITION	DESCRIPTION
A	$=\frac{(1+1.608\epsilon q_v L_v/(R_d T))}{(1+\epsilon q_v L_v^2/(c_p R_d T^2))}$	moisture coef for TKE buoyancy flux--saturated motion (10)
α_d	Tbl (1): DIVDMP	divergence damping coef (26)
α_t	Tbl (1): ROBERT	time filter coef (29)
b	=0.0196 (unsaturated) =9*0.0196 (saturated)	horizontal deformation coef for K_m (7)
$\overline{C^2}$	$=(c_p R_d \overline{\pi}\overline{\theta}_v)/c_v$	speed of sound squared (2)
c_*	Tbl (1): CSTAR	gravity wave phase speed
c_p	=1004.64 J/(kgK) (air) =4180.00 J/(kgK) (water)	specific heat of gas with respect to pressure
c_v	$=c_p-R_d$	specific heat of gas with respect to volume
C	=0.19 (iamxgl=1,2,3) =0.125 (iamxgl=4) =0.19+0.51*l/Δs (iamxgl=5)	coef of C_ϵ
C_ϵ	$Ce^{1.5}/l_{dis}$	atmos TKE dissipation term (11)
D_{diff}	eqn (25)	numerical diffusion term (1-3,12-18)
D_{divd}	eqn (26)	divergence damping term (1)
D_{Rayd}	eqn (27)	Rayleigh damping term (1,3)
D_{turb}	eqn (4) and (5)	subgrid scale turbulence term (1-3,12-18)
Δs	$=(\Delta x \Delta y \Delta z)^{1/3}$	measure of the grid volume
e	eqn (8)--atmos; eqn (22)--ocean	turbulent kinetic energy (4,6)
ϵ_{ijk}	$=\epsilon_{123}=1$ $=\epsilon_{213}=-1$ otherwise=0	alternating unit tensor (1)
f	$=2\Omega\sin(\text{flat}*\pi/180°)$	Coriolis parameter

flat	---	latitude in degrees
g	$=9.806\ m/s^2$	gravitational acceleration
γ	$=\partial\overline{\theta}_v/\partial z$ (*unsaturated*) $=\partial\overline{\theta}_e/\partial z$ (*saturated*)	local lapse rate (10)
γ_{cg}	$=5Q_o/(z_i w_*)$	temperature countergradient (10)
h	eqn (16)	height of the ocean's free surface; perturb height from H
H	$\Sigma\ \sigma_w$ (ocean levels)	mean depth of the ocean (16)
k	$=2*pi/\lambda$	horizontal wavenumber
kk	---	number of vertical levels
K_d	Tbl (1): DIF2ND, DIF4TH	diffusion coef (25)
K_h	eqn (6) and eqn (7)	eddy mixing coef--heat (5)
K_m	eqn (6) and eqn (7)	eddy mixing coef--momentum (4)
κ	=0.4	von Karman constant
δ_{ij}	$=\delta_{i=j}=1$ $=\delta_{i\neq j}=0$	Kronecker delta (1),(4)
l	Tbl (1)--atmos; eqn (23)--ocean	turbulent mixing length (6)
l_{atm}	Tbl (1): AL0	atmos mixing length coef or asymptotic limit
l_{dis}	=l (iamxgl=1,2,3,5) $=fn(z_i, L, \gamma, e)$ (iamxgl=4)	mixing length for the TKE dissipation term C_ϵ
l_{ocn}	eqn (24)	ocean mixing length limit (23)
L	$=-u_*^3 T_v/\kappa g Q_o$	Monin-Obukhov length
L_v	=2500.3 J/g	latent heat of vaporization
λ	---	horizontal wavelength
m	---	number x-grid points
M_ϕ	Rutledge&Hobbs84	microphysical term (3)

n	---	number y-grid points
N	$=\sqrt{(g/\bar{\theta}_v)\partial\theta_v/\partial z}$	Brunt-Väisälä frequency
Ω	$=2*pi/day=7.272\times10^{-5}$ s^{-1}	angular velocity
p	eqn (19)	pressure
p_o	$=100000$ g/ms^2	atmos reference pressure
p_r	$=p-g\rho_o(H-z)$	ocean relative pressure (12-15)
pi	$=4*\arctan(1)=3.141592654$	180°
π	$=(p/p_o)^{R_d/c_p}=T/\theta$ eqn (2)	dimensionless atmos pressure perturbation
ϕ	---	general prognostic variable
q_c	eqn (3)	mixing ratio of cloud water
q_i	eqn (3)	mixing ratio of ice
q_l	$=q_v+q_c$	conserved moisture var (10)
q_r	eqn (3)	mixing ratio of rain water
q_s	eqn (3)	mixing ratio of snow
q_v	eqn (3)	mixing ratio of water vapor
q_*	Barker&Baxter75; Louis79	surface moisture scale
Q_o	$=\theta_* u_*$ (*dry*) $=\theta_* u_*+0.608T_v q_* u_*$ (*moist*)	surface heat flux
R_d	$=287.04$ J/kgK	gas constant for dry air
ρ	eqn (21)	density
ρ_o	$=1035$ kg/m^3	ocean reference density (12-15)
S	eqn (18)	salinity
S_h	Tbl (1): IASHSM	coef of K_h (6)
S_{h0}	Tbl (1): SH0	coef of S_h
S_m	Tbl (1): IASHSM	coef of K_m (6)
S_{m0}	Tbl (1): SM0	coef of S_m

σ_m	Fig (1)	height array for mass points
σ_w	Fig (1)	height array for w points
t	---	time; large time step
t_R	eqn (28)	coef for Rayleigh damping (27)
T	eqn (17)	temperature
T_v	$=T*(1+0.61q_v)$	virtual temperature
τ	---	time; small time step
θ	eqn (3)	potential temperature
θ_v	$=\theta*(1+0.61q_v)$	virtual potential temperature
θ_*	Barker&Baxter75; Louis79	surface temperature scale
u_{gi}	$=u_{g1}=u_g$ $=u_{g2}=v_g$	u-comp: geostrophic wind (1) v-comp: geostrophic wind (1)
u_i	$=u_1=u$ eqn (1)--atmos eqns(12)+(14)--ocean $=u_2=v$ eqn (1)--atmos eqns(13)+(15)--ocean $=u_3=w$ eqn (1)--atmos eqn (20)--ocean	u-comp: wind : ocean current v-comp: wind : ocean current w-comp: wind : ocean current
u_*	Barker&Baxter75; Louis79	surface velocity scale
u'	eqn (12)	u-comp perturbation ocean current (baroclinic mode)
$\overline{u}$	eqn (14)	u-comp vertical mean ocean current (barotropic mode)
v'	eqn (13)	v-comp perturbation ocean current (baroclinic mode)
$\overline{v}$	eqn (15)	v-comp vertical mean ocean current (barotropic mode)
w_*	$=(gQ_o z_i/\overline{\theta})^{1/3}$	convective velocity scale
x_i	$=x_1=x$ $=x_2=y$ $=x_3=z$	cartesian coordinate directions
z_i	---	height of the PBL inversion; 20% of the TKE maximum

z_T	$=\Sigma\ \sigma_w$ (ocean + atmos levels)	model domain top
z_1	---	lowest height of Rayleigh damping absorbing layer (28)

Appendix B: Lines of code extracted from the model's include file *nhcd04.h*. The namelist **/nonhyd/** reads from the input file *case#.data*, user defined values for model input parameters.

```
          •
          •
          •
c*****************************************************************
c
      namelist/nonhyd/
c
     1  cpathi,cpatho,messg,xzflda,xzfldo,c1dfld,xyflda,xyfldo
c
     1 ,almin,alpha,al0,cbar,cstar,delta,delto,delx,dely,delza
     2 ,dif2nd,dif4th,flat,psfc,rdtime,robert,sh0,sm0,tamp
     3 ,tjtea,tjtia,tjtsa,tjxa0,tjya0,tjza0,tjteo,tjtio,tjtso
     4 ,tjxo0,tjyo0,tjzo0,tprime,twidth,t1d,ugeo1d,umean
     5 ,vgeo1d,xytea,xyteo,xytia,xytio,xytsa,xytso,xztea,xzteo
     6 ,xztia,xztio,xztsa,xztso,zampa
c
     1 ,iadvct,iahsgm,iamxgl,iashsm,iavgen,iavgst,ibdya,ibdyo,icase
     2 ,idgadj,idgafe,ielead,isflx,islead,ixylva,ixylvo,ixziea
     3 ,ixzieo,ixzisa,ixziso,ixzjea,ixzjeo,ixzjsa,ixzjso,i1d,jbdya
     4 ,jbdyo,jdgadj,jdgafe,j1d,katmos,ktaust,krdtn,maxtim,mfield
     5 ,mforce,mframe,mprnta,mprnto,msavea,msaveo,mstata,mstato
     6 ,mtaua,mtauo,nrdamp
c
     1 ,lasym,lcool,lddamp,ldgadj,ldgafe,ldiff,lfft1d,lfft2d,lflux
     2 ,lice,linit,lles,llht,lmlmin,lmoist,losym,lpert,lralee,lrdtn
     3 ,lsplit,ltke,ltwod,lvgeo,lwimp,lwritu,lwtop,l4oadv
c
c*****************************************************************
c
      read (5,nonhyd,end=101)
  101 continue
          •
          •
          •
```

Appendix C: A sample script file named *case8.cc*, that creates the user input files (*nhcd01.h* and *case8.data*), compiles and links the source code, and runs the **COAMPS** executables (*coama* and *coamm*).

```
# ##############################################################
# ##############################################################
#    CASE 8:  South Park, CO convective storm
# ##############################################################
# ##############################################################

#@$-q lsc
#@$-lT 01:30:10
#@$-lt 01:30:00
#@$-lM 11Mw
#@$-lm 11Mw
#@$-eo
#@$-s /bin/sh

# ##############################################################
#    batch process commands; pathnames; transfer code to scratch
# ##############################################################

set echo
set timestamp
newacct NE8000
NCPUS=1
export NCPUS
NPROC=8
export NPROC
srcd=/u/b/hodur/coamps/src/test
pfcs=/scr/hodur/coamps/data/case8/output

cd ${pfcs}
rm * 2> /dev/null
cd ${srcd}
/usr/local/bin/remdmget *
tar -cf - *.f *.h makefile.* | (cd ${pfcs};  tar -xf - )
cd ${pfcs}

# ##############################################################
#    create the nhcd01.h include file (grid dimensions)
# ##############################################################

cat << eof > nhcd01.h
c
c          parameter statement
c
      parameter (M= 55,N= 55 ,KK= 28
     1          ,    m1=   M-1,    m2=   M-2,    m3=   M-3
     2          ,    n1=   N-1,    n2=   N-2,    n3=   N-3
     3          ,    k1=  KK-1,    k2=  KK-2,    k3=  KK-3
```

```
     4           ,  kkp1=  KK+1,  kkp2=  KK+2,  kkp3=  KK+3
     5           ,   mkk=  M*KK,  m1kk= m1*KK,    mn=   M*N
     6           ,   m1n=  m1*N,   mn1=  M*n1,   mk1=  M*k1
     7           ,   mk2=  M*k2,   mk3=  M*k3)
eof

# ############################################################
#   compile and link the analysis and model executables
# ############################################################

make -f makefile.coama
make -f makefile.coamm

# ############################################################
#   create the case#.data input file
# ############################################################

cat << eof > case8.data

 &nonhyd

# ############################################################
#   setup the case specific input parameters
# ############################################################

 CPATHI= '/scr/hodur/coamps/data/case8/output/'
 CPATHO= '/scr/hodur/coamps/data/case8/graphics/'

 AL0   = 600.0,
 DELTA = 6.0,
 DELX  = 750.0,
 DELY  = 750.0,
 DELZA = 600.0,600.0,600.0,600.0,600.0,600.0,600.0,600.0,600.0,600.0,
         500.0,500.0,500.0,500.0,500.0,500.0,400.0,400.0,400.0,400.0,
         400.0,400.0,300.0,300.0,300.0,300.0,300.0,300.0,
 FLAT  = 40.0,
 ROBERT= 0.2,
 SH0   = 0.3,
 SM0   = 0.1,
 UGEO1D= 12.65, 12.72, 12.79, 12.86, 12.93, 13.01, 13.08, 13.14,
         13.18, 12.65, 11.68,  9.81,  8.09,  6.66,  5.61,  5.28,
          4.95,  4.60,  4.25,  3.83,  3.39,  2.66,  1.43,  0.37,
         -0.66, -1.58, -2.50, -3.43,
 UMEAN = 0.0,
 VGEO1D= 16.64, 16.26, 15.89, 15.51, 15.13, 14.75, 14.37, 13.37,
         12.06, 10.02,  7.96,  5.73,  3.82,  2.54,  1.56,  1.16,
          0.64, -0.19, -1.02, -1.61, -2.14, -2.36, -1.93, -1.56,
         -1.20, -0.84, -0.48, -0.13,

 IAHSGM= 2,
 IAMXGL= 2,
 IASHSM= 2,
```

```
 IBDYA = 4,
 ICASE = 8,
 ISFLX = 2,
 JBDYA = 4,
 KATMOS= 28,
 KTAUST= 0,
 MAXTIM= 5400,
 MFIELD= 43200,
 MPRNTA= 108000,
 MSTATA= 436000,
 MTAUA = 3,

 LCOOL = f,
 LDIFF = t,
 LFLUX = f,
 LICE  = t,
 LMOIST= t,
 LRALEE= f,
 LSPLIT= t,
 LTKE  = t,
 LTWOD = f,
 LWRITU= f,
 LWIMP = f,
 LWTOP = f,

# ##################################################################
#   setup the graphics input parameters
# ##################################################################

### 1D grid point plot specifications ###
 C1DFLD= 'pott', 'wwww', 'eeee', 'uuuu',
 I1D   =     10,     20,     30,     40,
 J1D   =     15,     15,     15,     15,
 T1D   = 3600.0,

### XZ cross section plot specifications ###
 XZFLDA= 'wwww',
         'pott',
 XZTSA = 0.0,
         0.0,
 XZTEA = 5400.0,
         5400.0,
 XZTIA = 60.0,
         60.0,
 IXZISA= 1,
         1,
 IXZJSA= 28,
         28,
 IXZIEA= 55,
         55,
 IXZJEA= 28,
         28,
```

```
### XY horizontal level plot specifications ###
 XYFLDA= 'wwww', 'wwww',
         'pott', 'pott',
 XYTSA =    0.0,    0.0,
            0.0,    0.0,
 XYTEA = 5400.0, 5400.0,
         5400.0, 5400.0,
 XYTIA =  300.0,  300.0,
          300.0,  300.0,
 IXYLVA=     17,     28,
             17,     28,

### trajectory plot specifications ###
 TJTSA =    0.0,    0.0,    0.0,
            0.0,    0.0,    0.0,
 TJTEA = 5400.0, 5400.0, 5400.0,
         5400.0, 5400.0, 5400.0,
 TJTIA =    6.0,    6.0,    6.0,
            6.0,    6.0,    6.0,
 TJXA0 =   14.6,   28.0,   41.3,
           14.6,   28.0,   41.3,
 TJYA0 =   14.6,   14.6,   14.6,
           14.6,   14.6,   14.6,
 TJZA0 =   27.5,   27.5,   27.5,
            8.0,    8.0,    8.0,
&end

eof

# ###################################################################
#    Execute the analysis
# ###################################################################

ja -m
( time ./coama < case8.data )
ja -cst

# ###################################################################
#    Execute the forecast model
###################################################################

ja -m
( time ./coamm < case8.data )
ja -cst

# ###################################################################
#    Job is complete
###################################################################
```