DOE/ER/61926--T1

Variational Optimization of Sub-grid Scale
Convection Parameters

Final report

Submitted by
Atmospheric and Environmental Research, Inc.
840 Memorial Drive, Cambridge, MA 02146

In Fulfillment of
Department of Energy CHAMMP/CLIMATE Program
Grant Number DE-FG02-94ER61926
and
Sub-Contract Report in Support of
Department of Energy CHAMMP/CLIMATE Program
Grant Number DE-FG02-91ER61220 awarded to MIT

Marina Živković - Rothman

November 25, 1997

MASTER

fat
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed or represents that its use would not infringe privately-owned rights.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
electronic image products. Images are
produced from the best available original
document.**

Table of Contents

Summary	i
1. Introduction	1
2. Convection Model	2
2.1. Nonlinear model	3
2.1.1 Version 2.0	4
2.1.2 Version 3.0	4
2.1.3 Version 2.3 (hybrid) and parameters	7
2.2 Tangent linear model	9
2.3 Adjoint model	10
3. Adjoint Code Development	10
4. Testing of Adjoint Model Formulation	12
4.1 Correctness	13
4.2 Accuracy	17
5. Sensitivity Analysis	23
5.1 Single column model algorithms	24
5.2 Sensitivity to large-scale parameters	29
5.3 Sensitivity to subgrid-scale parameters	31
6. Conclusion and Future Work	38
Acknowledgment	41
References	42
Appendix A. Nonlinear Code of Model Subroutine ENERGY	45
Appendix B. General Code of Model Routine CONVECT	47
B-1 Generic Code	48
B-2 Nonlinear Code	53
B-3 Adjoint Code	56

Summary

Under the DOE CHAMMP/CLIMATE Program, a convective scheme was developed for use in climate models. The scheme was developed by Prof. K. A. Emanuel, and the purpose of the present study was to develop an adjoint model of its tangent-linear model. The convective scheme was integrated within a single column model which provides radiative-convective equilibrium solutions applicable to climate models.

The main goal of this part of the project was to develop an adjoint of the scheme to facilitate the optimization of its convective parameters. For that purpose, adjoint sensitivities were calculated with the adjoint code. The optimization of the parameters was done by Prof. Emanuel under the contract awarded to MIT, and which is reported separately. Parameter optimization was based on TOGA COARE data which were also used in this study to obtain integrations of the nonlinear and tangent-linear models as well as the integrations of the adjoint model. Some inadequacies of the inner IFA data array were found, and did not permit a single numerical integration during entire 4 months of data. However, reliable monthly radiative-convective equilibrium solutions and associated adjoint sensitivities were obtained and used to bring about the parameter optimization.

1. Introduction

Prof. Emanuel's proposal to DOE was to study the sensitivity of his convection scheme (Emanuel, 1991; Rennó et al., 1994; Emanuel and Živković-Rothmaan, 1997) to the various parameters of the scheme, and to optimize these parameters. Two methods were proposed: a standard sensitivity analysis that modifies each parameter in turn, and the use of the adjoint method (Louis and Živković, 1994). Using the adjoint to compute the gradient of the forecast error with respect to the model parameters allows the use of standard minimum search techniques to optimize model parameters. There are a number of potential pitfalls. First of all, it is not possible to distinguish between the forecast errors due to model parameters and those due to inaccuracies in the data or other errors in the model algorithms. There is thus a danger of overfitting the data (Wergen, 1992). This danger can be minimized by using several different data sets.

The behavior of the atmosphere is quite different when convection occurs than when it is stable. The equations that describe the evolution of the atmosphere are discontinuous at the convection threshold. At or near that point, the derivatives computed by the adjoint may be quite different from those computed by finite difference. This may create difficulties in the optimization procedure by creating local minima or discontinuities in the objective function. The adjoint computations were therefore carefully checked by making finite difference estimates of the derivatives. These results are presented in section 4.

The adjoint method is, in principle, particularly well suited to sensitivity analysis (e.g., Živković *et al.*, 1995) since the derivatives of a functional of the

model output with respect to all the inputs and parameters can be computed with a single integration of the model and one integration of the adjoint model. However, the adjoint formalism assumes that all perturbations remain small during the model integration. This assumption can fail for processes with thresholds such as moist convection. Results of sensitivity analysis are presented in section 5.

Two main tasks were proposed and provided by the PI: i) Development of the adjoint of the convection scheme, and ii) support to Prof. Emanuel in testing the optimization procedure. This report presents results on: development of the convective model version that is suitable for adjoint coding (section 2), development of its adjoint model (section 3), testing of the model convergence to finite difference integrations (section 4), and adjoint and traditional sensitivity analysis (section 5).

2. Convection Model

The Emanuel's convective model used in the present study differs from the widely used entraining-cloud model. It is based on an episodic-mixing approach, in which parcels enter the cloud at the cloud base, move vertically through the cloud in updraft without mixing, then instantaneously mix when they reach the neutral-buoyancy level. Cloud microphysical parameters determine the amount of cloud water converted to precipitation.

During current 3-year project, we worked with 2 versions of the convection model and developed a new model version which is suitable for writing the adjoint model.

2.1. Nonlinear Model

The bulk of work within the first two years was done with version 2.0 of the convection scheme. The scheme was coupled with the radiation model developed by Chou (1994, 1995, 1996) of NASA/Goddard. The radiative-convective equilibrium simulations based on 21-day runs by these two models (Rennó et al., 1994) were used to test modifications of the nonlinear model and development of the adjoint codes. In the third year, version 3.0 of the scheme was introduced. Model version 3.0 was run with a single column model forced by vertical velocities and horizontal temperature and moisture advection provided by Prof. Richard Johnson and Paul Ciecelski of Colorado State University (Lin and Johnson, 1996). These forcings were calculated from the TOGA/COARE inner flux array (IFA) analyses of temperature, water vapor, vertical velocity, and horizontal advection of temperature and water vapor (Velden and Young, 1994; Weller and Anderson, 1996). The radiative heating rates were pre-calculated by the radiation model developed by Chou.

Development of the adjoint code of version 3.0 involved significant change of the algorithm structure and extensive testing of the adjoint codes. As a safeguard should such a task be incomplete during the current funding period, a hybrid version 2.3 was developed. The latter version preserved the algorithm structure of version 2.0 and included most of the code modifications of version 3.0. The hybrid version of the scheme was then used to test the algorithm convergence and initial sensitivity experiments.

2.1.1 Version 2.0

In the first year of this project, the version 2.0 of the nonlinear code (hereafter called NLM) was modified to eliminate potential difficulties in writing the adjoint. Repetitive calculations were eliminated and convective parameters were grouped in a common block suitable for optimization codes. All variables have been sorted according to the need to store or recalculate their values during the climate runs. Since the single column model provides the radiative-convective equilibrium solutions, the optimization runs were planned to be of a long duration (between 1 and 4 months). This type of integration with adjoint codes requires very large direct access files where the history of the NLM integrations are stored. Because of the file size, it was crucial to evaluate which variables should be stored during time integrations and which variables should be re-calculated every time step. This process resulted in a significant code reformulating. Critical IF statements and intrinsic FORTRAN functions such as MIN and MAX were identified and recoded to facilitate derivation of the adjoint. These statements are representative of process discontinuities and need to be exactly preserved when the adjoint equations are solved. An example of the reorganized nonlinear code with new variable declaration for energy calculations is presented in Appendix A. Also, the code was reorganized to simplify its logical structure. The original code was rewritten into 14 modules which are listed in Table 1. The adjoint code was derived for 11 individual modules.

2.1.2 Version 3.0

In the third year of this project, version 3.0 of the scheme was introduced as a part of a single column model that calculates radiative-convective equilibrium from TOGA/COARE data.(For details on model development see

Figure 1. Schematic diagram of version 2.3 of the convection scheme.

time-scale calculations needed to provide stability of the time-dependent solution. The most important changes were: the update of calculations of the terminal velocity, modification of the mass flux calculations and a new version of dry adiabatic adjustment subroutine. The calculations of the terminal velocity was divided between the liquid and solid water condensate. Additional modifications of the mass flux calculations removed explicit dependency of the fractional updraft/downdraft areas below the cloud base.

2.1.3 Version 2.3 (hybrid) and parameters

Since the adjoint code of the version 3.0 would involve significant changes of the algorithm structure and computationally extensive testing of the adjoint model, a hybrid version 2.3 was developed for the purposes of this project. The hybrid version of the code preserved the algorithm structure of version 2.0 and included most of the necessary modifications of version 3.0. Also the old version of the dry adiabatic adjustment subroutine was retained as well as the large scale parameters, α and β . The adjoint model (hereafter called ADJM) and tangent linear model (hereafter called TLM) codes were updated with all other modifications except the new mass flux calculations below the cloud base. Also, calculations of the vertical velocity (within the cloud) and the updraft/downdraft fractional areas were retained for diagnostic purposes. The convection parameters of this scheme, which were optimized in the present study, are listed in Table 2. Figure 1 presents the schematic diagram of the scheme version 2.3.

Rewriting of the nonlinear code and development of the hybrid version involved extensive coding and comparison of the integrations of newly developed codes with integrations obtained by the original codes. Initial

rewriting of the nonlinear codes generally removed about 90% of discrepancies that were not suitable for writing the adjoint. Additional revisions of the nonlinear model were made later when the adjoint codes were developed.

Table 2 . Convection parameters optimized with TOGA/COARE data.

Parameter	Description
ELCRIT	Critical liquid water content for warm cloud autoconversion.
TLCRIT	Critical transition temperature related to the autoconversion threshold value.
SIGD	The fractional area assumed to be covered by the precipitation-driven downdraft.
SIGS	The fraction of the precipitation shaft assumed to fall through unsaturated air outside the cloud.
SIGMD	The fractional areal coverage of spreading cold pools at the surface.
WTRAIN	The terminal velocity of rain [m/sec]. This is assumed to be a constant real fall speed.
WTSNOW	A factor multiplying the terminal fall speed of snow. Unlike rain, snow is taken to fall at the constant pressure velocity.
COEFFR	A coefficient multiplying the rate of evaporation of rain.
COEFFS	A coefficient multiplying the rate of evaporation of snow.
DTMAX	The maximum temperature inversion strength[deg K] permitted at the top of the sub-cloud layer.
SLFRAC	The relative strength of the "toe" in temperature and specific humidity in the surface layer.
ENTP	A factor controlling the rate of entrainment into updrafts.
ALPHA	The rate of approach of the mass fluxes to quasi-equilibrium. Large-scale parameter.
BETA	A large-scale dumping factor.

Finally, the ultimate changes were made during the debugging process of the adjoint codes when the remaining code discrepancies were identified and reprogrammed. Once the code produced a correct adjoint test, it was

implemented back into the nonlinear model and tested against the original radiative-convective equilibrium integrations. Most of the code development effort supported by this project was spent on this process of producing nonlinear and adjoint codes that are consistent with integrations obtained with the original scheme.

2.2 Tangent Linear Model

If the symbolic form of a nonlinear model is

$$\mathcal{M} : \frac{d}{dt}(\mathbf{X}(t)) = \mathcal{F}(\mathbf{X}(t)), \quad (1)$$

then, the tangent linear model, $\mathcal{R}$, is obtained by the linearization of the nonlinear model around a time-evolving solution $\mathbf{X}(t)$,

$$\begin{aligned} \frac{d}{dt}(\mathbf{X}(t) + \delta\mathbf{X}(t)) &= \mathcal{F}(\mathbf{X}(t) + \delta\mathbf{X}(t)) \\ &= \mathcal{F}(\mathbf{X}(t)) + \frac{\partial \mathcal{F}(\mathbf{X}(t))}{\partial \mathbf{X}} \cdot \delta\mathbf{X}(t) + O(\delta\mathbf{X}(t)), \end{aligned} \quad (2)$$

and by retaining only the first order terms in $\delta\mathbf{X}$,

$$\mathcal{R} : \frac{d}{dt}(\delta\mathbf{X}) = \mathcal{F}'_{\mathbf{X}} \cdot \delta\mathbf{X}. \quad (3)$$

The TLM therefore describes to first order the evolution of a perturbation $\delta\mathbf{X}$ around a nonlinear solution $\mathbf{X}(t)$ (Lacarra and Talagrand, 1988). The vector $\mathbf{X}(t)$ will also be referred to as the model *state vector* at some discrete time level, and the matrix $\mathcal{F}(\mathbf{X}(t))$ will be referred to as a *resolvent* of a nonlinear model between two consecutive time levels.

2.3 Adjoint model

Within the scope of the above formalism, the adjoint model (ADJM) of Eq. (3) is defined as a transpose of the TLM, *i.e.*,

$$\mathcal{R}^* : -\frac{d}{dt}(\delta\mathbf{X}') = \mathcal{F}_X'^* \cdot \delta\mathbf{X}', \quad (4)$$

where * denotes a transpose or a complex conjugate transpose (Talagrand and Courtier, 1987). In more general mathematical terms, if $\mathbf{X}$ and $\mathbf{Y}$ are two vectors (for example state vectors) with defined inner product and linear operator, $\mathcal{R}$

$$\langle \mathbf{X}, \mathcal{R} \mathbf{Y} \rangle = \langle \mathcal{R}^* \mathbf{X}, \mathbf{Y} \rangle \quad (5)$$

then $\mathcal{R}^*$ is simply the adjoint of $\mathcal{R}$.

In the case of a model described by Eq. (1), $\mathcal{F}'$ is the Jacobian matrix of the operator $\mathcal{F}(\mathbf{X}(t))$ evaluated for the state vector $\mathbf{X}(t)$ at some discrete time level. If the nonlinear model contains discontinuous functions, then some elements of the Jacobian will not exist for some state vectors or may be very small. This limits the applicability of the adjoint $\mathcal{R}^*$. In the case of convection models, the solution is often adjusted to neutral stability states where Jacobian derivatives become nonexistent. Even correct ADJMs may then have substantially reduced applicability and be accurate only locally (Županski, 1993). For this reason, it is particularly important to evaluate the accuracy of the convection adjoint model before any application.

3. Adjoint Code Development

Writing of the adjoint of the convection scheme involves 3 steps: i) customization of the original code to a modular form suitable for writing the adjoint code, ii) linearization of each module, and iii) testing of its adjoint. All

3 steps can be done independently. During this project we developed and tested two versions of the convection scheme which were run within two different single column models.

For each module listed in Table 1, the input and output variables, parameters and local variables were precisely identified. This information was provided to the software that generated adjoint codes. During the course of this study, an Automatic Adjoint Compiler (AMC) from the Max Planck Institute became available to AER from Ralph Giering. This software was tested against the software developed by AER. It was found to be computationally faster since it produced only the adjoint code based on the original nonlinear code. However, a disadvantage was that it lacked code for the tangent linear model (TLM) which was needed for testing the validity of the adjoint code (see section 4). To facilitate this test, additional TLM code was manually developed for each of the modules.

As the writing of the adjoint codes progressed towards more complex modules, it was found that this approach had become inefficient. The AMC software worked well for simpler routines when it was easy to reconstruct the TLM codes. However, when more complex routines needed to be re-coded, the AMC software became useless since code debugging became impossible. The major part of the scheme is contained in the last five modules from Table 1. The adjoint codes (ADJM) of these modules were developed using the AER software.

With the AER software, the development of the adjoint is more systematic than with the AMC software. For ease of code maintenance, all 3 codes (NLM, TLM and ADJM) are kept together. The general principle is that, in

the primary (generic) code, each line of the model is followed by the corresponding line(s) of the TLM and ADJM. These lines are appropriately tagged with labels at the beginning of the line: <NLM>, <LTM> or <ADJ>. In addition, the lines of code that need to be repeated in the TLM and ADJM to recompute needed intermediate variables are tagged with the label <ALL>, and some statements that do not appear in the NLM but are common to the TLM and ADJM are tagged <PER>. Appendix B-1 contains an example of the model generic code for routine CONVECT developed in the manner just described.

The generic code is then used as an input for a text processing macro which creates corresponding FORTRAN codes. It selects the lines that are appropriately tagged and ignores the others, then removes the tags and replaces them by blanks. In the case of the adjoint it also reverses the order of the computations, except for the lines labeled <ALL>, which are moved to the top of the code structure (for example, DO loop, subroutine, IF ... ENDIF block, etc.). This macro was created in LISP language and is used in the GNU Emacs environment. Appendices B-2 and B-3 exemplify NLM and ADJM codes derived by this program.

4. Testing of Adjoint Model Formulation

It is possible to estimate applicability of an adjoint model by examining the accuracy of a corresponding TLM (Errico *et al.*, 1993). Inaccuracy of a TLM is caused by two sources of error: i) the model discontinuity and ii) the nonlinearity of the model forcing terms. Convection models contain both sources of error by their nature. They represent discontinuous physical processes and contain on/off switches. Also, they are highly nonlinear since

they adjust exponentially varying variables (for example see Vukićević and Errico, 1993).

4.1 Correctness

In the present study, we utilize the "classic" adjoint method to test model correctness which neglects variation of the switch point (Xu, 1996). This means that a correctly formulated TLM (and its ADJM) should contain the same set of discontinuous switches as a nonlinear model. To ensure that this is true, we performed a simple test of the asymptotic behavior of the TLM.

To simplify our presentation, consider model integration at a single time level. A derivative calculated by the TLM, and a derivative calculated by the nonlinear model, should converge asymptotically if the difference ϵ , defined as

$$\epsilon = \mathcal{F}'_X - \lim_{\Delta \mathbf{X} \rightarrow 0} \frac{\mathcal{F}(\mathbf{X} + \Delta \mathbf{X}) - \mathcal{F}(\mathbf{X})}{\Delta \mathbf{X}}. \quad (6)$$

approaches zero as the perturbation of the model state vector, $\delta \mathbf{X}$, decreases to zero. In the case of an error caused by an incorrect switch (representing model discontinuity), the difference ϵ does not vanish even if $\delta \mathbf{X}$ is very small. In the case of an error caused by a model nonlinearity, the value of ϵ decreases with a decrease of perturbation magnitude. Thus, the correctness of the TLM formulation with respect to these two types of errors can be examined by evaluating Eq. (6). The correctness of the ADJM is then simply determined by Eq. (5).

We tested model asymptotic behavior by calculating the first order derivatives for some random perturbation of the model state vector. The derivatives were calculated by the resolvent of the nonlinear model and

compared with corresponding TLM derivatives obtained for the same perturbation. Three types of derivatives were calculated by the finite differentiation of nonlinear iteration: the left-hand derivative denoted by $\mathbf{X} - (\mathbf{X} - \Delta\mathbf{X})$, the right-hand derivative $(\mathbf{X} + \Delta\mathbf{X}) - \mathbf{X}$, and the centered derivative $(\mathbf{X} + \Delta\mathbf{X}) - (\mathbf{X} - \Delta\mathbf{X})$. The TLM derivatives were compared with all three types of finite difference derivatives. We also compared the finite difference derivatives with each other in order to assess accuracy of our calculations.

Fig. 2 *RMS for TLM and centered finite difference derivatives based on the d_2 RMS metric function.*

The comparison was done by calculating various metric functions based on a difference between corresponding derivatives. Several metric functions were calculated:

- i) the *Mean and Root Mean Squared (RMS) fractional metrics* d_1 ,

$$d_1 M = \frac{1}{N} \sum_i \frac{|\mathbf{X}_i - \mathbf{Y}_i|}{\frac{1}{2}(|\mathbf{X}_i| + |\mathbf{Y}_i|)}, \quad d_1 RMS = \sqrt{\frac{1}{N} \sum_i \left[\frac{|\mathbf{X}_i - \mathbf{Y}_i|}{\frac{1}{2}(|\mathbf{X}_i| + |\mathbf{Y}_i|)} \right]^2},$$

ii) the *Mean and Root Mean Squared (RMS) Euclidian metrics* d_2

$$d_2 M = \frac{1}{N} \sum_i |\mathbf{X}_i - \mathbf{Y}_i|, \quad d_2 RMS = \sqrt{\frac{1}{N} \sum_i |\mathbf{X}_i - \mathbf{Y}_i|^2},$$

iii) the *Chebychev metric*

$$d_\infty = \max_{i=1, \dots, N} |\mathbf{X}_i - \mathbf{Y}_i|, \quad \text{etc.}$$

Vectors $\mathbf{X}_i$ and $\mathbf{Y}_i$ contained corresponding derivatives calculated by the two methods. The number of scalar components of the state vector was $N = 552$ for the 50 model levels. The d_1 metrics were used for their high sensitivity to the occurrence of a single significantly large deviation for any scalar component.

Figure 2 illustrates the difference between the two types of calculations based on the $d_2 RMS$ metric. Here, we plotted a difference between TLM and the centered derivative as a function of perturbation magnitude (in percent of the model state vector). Initial profiles of temperature, humidity and pressure were randomly perturbed starting with 10% and ending at 0.0001% perturbation magnitude. Due to a highly nonlinear nature of the model, finite difference calculations did not produce meaningful results for perturbations larger than 1%. The unperturbed NLM run generated convection between levels 2 and 20. However, the NLM run with 10% perturbation magnitude generated convection cloud between levels 2 and 17 in the case when the perturbation was added to the initial model state vector. In the case when perturbation was subtracted, the convection cloud was generated between levels 8 and 13. The TLM by definition did not "see" these nonlinearities, and

always generated convection cloud between levels 2 and 20. For perturbations smaller than 1%, the two models produce comparable and linearly converging results as it demonstrated by the Fig. 2.

Table 3. Values of metric functions d_1M and d_1RMS for TLM and finite difference calculations .

Perturbation magnitude	Derivative type	Finite difference derivative type	d_1M	d_1RMS
1%	TLM	$X - (X - \Delta X)$	0.452	0.942
	TLM	$(X + \Delta X) - X$	0.432	0.911
	TLM	$(X + \Delta X) - (X - \Delta X)$	0.450	0.934
	$X - (X - \Delta X)$	$(X + \Delta X) - X$	0.353	0.815
	$X - (X - \Delta X)$	$(X + \Delta X) - (X - \Delta X)$	0.158	0.411
	$(X + \Delta X) - X$	$(X + \Delta X) - (X - \Delta X)$	0.312	0.748
0.1%	TLM	$X - (X - \Delta X)$	0.413	0.886
	TLM	$(X + \Delta X) - X$	0.409	0.883
	TLM	$(X + \Delta X) - (X - \Delta X)$	0.409	0.880
	$X - (X - \Delta X)$	$(X + \Delta X) - X$	0.141	0.450
	$X - (X - \Delta X)$	$(X + \Delta X) - (X - \Delta X)$	0.113	0.401
	$(X + \Delta X) - X$	$(X + \Delta X) - (X - \Delta X)$	0.058	0.187

To assess the accuracy of the derivative computations, we also tested how the finite difference calculations compared with each other. Table 3 exemplifies this comparison based on d_1 metric functions. Metric values in the last two columns demonstrate that derivatives calculated by both methods are computed with comparable accuracy. We note here that additional errors due to time integration can affect ADJM correctness (Errico et al., 1993). In operational models which include a large number of grid points, the true evolution of the model state is often approximated by interpolated values between the time levels when nonlinear calculations are stored. This type of

error is not a concern in the present study, since the true nonlinear solution is used for each iteration of the ADJM.

4.2 Accuracy

In principal, TLM and ADJM can be considered correct as long as the linearization is performed properly and completely. This procedure provides proper asymptotic behavior of the models for a small perturbation as illustrated by the above results. However, it does not necessarily provide accurate models. The single iteration tests described above do not necessarily include all possible Jacobian derivatives. The accuracy of the model is determined by their general results which describe solution behavior for finite-amplitude perturbations. In other words, expression (6) should be satisfied in more general terms as described by

$$\lim_{\delta X \rightarrow 0} \frac{\mathcal{M}(X + \delta X)_i - \mathcal{M}(X)_i}{(\mathcal{R} \cdot \delta X)_i} = 1 \quad (7)$$

where i denotes the i^{th} component of the state vector $\mathbf{X}$. For that reason any TLM should be tested for its accuracy in addition to its correctness. This is done by time integration of the TLM and nonlinear model for a slightly perturbed initial state (for example see Thépaut and Courtier, 1991).

We started integration of both models with the TOGA/COARE soundings and a single column model running the convection and radiation models. An example of the nonlinear 24-hour integration with TOGA/COARE soundings available every 6 hours is given in Figure 3(a). As diagnosed by the convective pressure velocity ($\omega = \delta p / \delta t$), the convection established after approximately 2 hours, and after initial oscillation, settles in through the rest of integration period. Diagram (b) in the same figure shows temperature tendencies during

(a)

(b)

Fig. 3 (a) The convective pressure velocity during a 24-hour integration of the NLM based on TOGA/COARE data. (b) Temperature tendencies calculated during the first 4 hours of the same model run as in (a). The dashed line pattern denotes negative values.

the first 4 hours of integration of the same NLM run. Negative values are marked by the dashed line pattern. The tendency pattern during this period shows briefly 20 min of positive tendencies in the lower levels, followed by approximately 20 min of negative tendencies, which is then followed by the positive tendency values until the end of the run. We will refer to this integration as a control run.

We used this example to present evolution of the finite-amplitude perturbation simulated by the NLM and the TLM. First we made two additional NLM runs with the same vertical soundings used as an initial state. The model convective parameters as well as the temperature and moisture profiles are then modified by adding a random perturbation to the initial state of the magnitude of observational error. In this case we chose 1% error which is equivalent to the relative temperature error. One NLM run is made with added finite-amplitude and the second NLM run is made with the subtracted finite-amplitude perturbations. The temperature tendency difference between the perturbed and the control NLM run is shown in diagram (a) of Figure 4. The notable feature of this diagram is that the perturbation evolution follows closely the evolution of the NLM solution as presented in diagram 3(b). However, the sign of the tendency perturbation is the opposite to the sign of the tendency itself.

The corresponding TLM run with the same finite-amplitude perturbation of the initial state is shown in diagram (b) of Figure 4. The two diagrams are qualitatively comparable following the general feature of the nonlinear solution (a short period of negative tendency perturbation is followed by the period of the positive tendency perturbation, etc.). However, the differences

(a)

(b)

(c)

Fig. 4 (a) Difference between the temperature tendency evolution during the first 4 hours integration of the NLM runs with perturbed initial state. (b) The evolution of the temperature tendency calculated by the TLM with the same perturbation of the initial state as in (a). (c) The evolution of the temperature tendency calculated by the TLM with 10% perturbation of the model parameters and no perturbation of the initial state.

Fig. 5 Comparison between an incorrect TLM run (a) and correct TLM run (b) using the same perturbation of the initial state as in Fig. 4.

are most notably due to the discontinuities of the NLM. In particular, in the sub-cloud layer the TLM run produces extreme tendency perturbations which are completely absent in diagram (a). The model's apparent level of discontinuity is the first convective level, which in this case is around 900 mb. Because of the convection nature, the calculations are very different below the

cloud base than within the cloud, and the TLM shows strong sensitivity to this discontinuity.

However, if the initial state is not perturbed, but the convective parameters are, the TLM integrations follow quite closely the NLM integrations. Diagram (c) illustrates the tendency perturbation in the latter case, when the TLM is integrated with only convective parameters perturbed by 10% of their original values. Clearly, during this integration the perturbation evolution resembles NLM results in Fig. 4(a).

In applications involving the adjoint model, the magnitude of model sensitivity to threshold discontinuities shown in Fig.4(b) can cause a problem; hence it is important to identify them before any adjoint applications. Model integrations shown in Fig. 4 are typical examples of how the forward model sensitivity described by the TLM integrations can be used to detect problem discontinuities for purposes of an improvement of the NLM formulation and development of an applicable adjoint model. In this particular case, an error was found in the TLM (and consequently in the associated ADJM). A history of perturbations of two diagnostic variables, SIG and W0, was wrongly included in the TLM/ADJM codes. When this error was removed, the TLM integration produced results which were more comparable with the NLM results. Figure 5 illustrates the difference between the incorrect (diagram [a]) and correct (diagram [b]) TLM runs.

Note here that the typical time scales for convection are of the order of 3 hours, and it is reasonable to expect breaking of the linear model integrations beyond 120 minutes.

5. Sensitivity Analysis

An advantage of the adjoint models is that a single model integration in time produces sensitivity of a particular aspect of forecast to all input variables: parameters, initial conditions and boundary conditions. In the experiments presented in the rest of this report, the forecast aspect of interest is a forecast RMS error in relative humidity.

The *adjoint* forecast sensitivity should be distinguished from a *traditional* sensitivity analysis (Hall, 1986). In a traditional sensitivity analysis, a single model parameter is perturbed and sensitivity of all forecast features to changes in that particular parameter are determined. The adjoint sensitivities provide the inverse information. A sensitivity of a particular forecast feature to all input parameters is determined by the integration of the model adjoint. This makes adjoint technique particularly suitable for uncertainty analysis when it is known which forecast features are important for model simulations, but it is not known which aspect of the model causes the uncertainty. An analysis of adjoint sensitivities can therefore provide significant information on model aspects that would have the largest effect on the model simulation (e.g., Živković *et al.*, 1995).

For purposes of this study, traditional sensitivity analysis was performed for large-scale parameters α and β (see Table 2) to evaluate the adequacy of NLM version 2.3 for integrations with TOGA/COARE data. An analysis of the adjoint sensitivities to subgrid-scale convective parameters was performed for purposes of the parameter optimization done by Dr. Emanuel. The results of these analyses are presented in section 5.2 and 5.3, respectively.

5.1. Single column model algorithms

Sensitivity analyses were based on integrations of the convection model version 2.3. Both the NLM and ADJM were incorporated into single column models which read TOGA/COARE data, pre-calculated radiative and advective tendencies, and advanced the predicting variables in time. Figure 5 shows a schematic diagram of the nonlinear version of the single column model. The model state vector consists of input variables: pressure (p), temperature (T), specific humidity (R), saturation specific humidity (RS) at each model level, and the layer atmospheric pressure (PH). The output variables are temperature (FT) and relative humidity (FR) tendencies. During the time integration of NLM, a history file containing an input state vector at each time level was created and used for integration of the ADJM. A schematic diagram of the adjoint version of the single column model is shown in Fig. 6.

In addition to writing an adjoint of a single column model, a new module, VERINC, which calculates the objective function, J , was added to the model. Results presented in this report were obtained for the objective function defined as a forecast error in the column relative humidity, RH , i.e.,

$$J = \iint_{p,t} w \left(RH - RH_{obs} \right)^2 dp \cdot dt. \quad (8)$$

Initial integrations of the single column model for purposes of the parameter optimization were one month long. Four integrations for the months of November, December, January, and February were made. A total RMS error in the relative humidity and temperature was calculated during each integration. These errors are listed in Table 4.

Figure 5. Schematic diagram of the single column model.

The scores vary noticeably from month to month for several reasons: poor data quality, poor simulation and variable length of the model integrations. The variable length of integrations was caused by problematic data on 12/22 and 2/28 which abruptly ended integration of the nonlinear model. As a result, the integration during the month of December is only three weeks long. Consequently, the RMS error for relative humidity is significantly lower for this month than for the other three months. On the other hand, the temperature RMS error during the December integration is still larger than the error during a longer November integration. This is indicative of the error caused by a poor agreement of the simulation with observations probably due to inadequate values of the convective parameters.

Table 4. The temperature and relative humidity RMS error for monthly runs with original data array (first two rows) and filtered data (the last two rows).

RMS	November	December	January	February
Relative Humidity [%]	17.27	15.71	17.81	21.75
Temperature [° C]	2.58	2.72	3.65	3.48
Relative Humidity [%]	14.84	15.71	15.53	23.91
Temperature [° C]	2.28	2.72	3.45	4.19

The error values from the first row of Table 4 are final values of the objective function (8) at the end of each integration. The error growth during each integration is described by the evolution of the objective function. Diagrams in Fig. 7 present evolution curves of the objective function as defined by Eq. (8). Curves in diagram (a) are of variable length indicating the length of each integration. For example, the dashed line pattern represents 21 days long integration during the month of December. The February

Figure 6. Schematic diagram of the adjoint of the single column model.

(a)

(b)

Fig. 7 (a) Evolution of the objective function during four independent monthly integrations. (b) Evolution of the objective function during independent 3-week long integrations with selected data.

integration was 28 days long (solid thick line) and November (dotted line pattern) and January (solid line pattern) integrations were 30 and 31 days long, respectively.

Most importantly, the evolution curves indicate sudden error growth caused by poor agreement of the simulations with data. For example, the February curve in diagram (a) shows a good agreement with the December and January curves for the first seven days of integration. For the remainder of the integration, the slope of the curve is significantly larger than the slope of the other two curves. Similarly, the January curve suddenly changes its slope on 1/27. The November curve, on the other hand, exhibits a similar slope as the December and January curves following the first six days of integration. When the days with questionable data are removed from the integrations, the agreement of the curvature of the slopes becomes more apparent. Diagram (b) illustrates results of the 3-week long simulations made for each month without days with questionable data. In this diagram, the February curve clearly diverges from the other three curves indicating poor simulation which should be considered cautiously for optimization purposes. The RMS error in RH for 3-week long integrations (last two rows in Table 4) also shows less of the variability during the first three months.

5.2. Sensitivity to large-scale parameters

To ensure correctness of the agreement between the integrations with 2.3 version and 3.0 version of the convection model, a traditional sensitivity analysis of the RMS errors was performed for large-scale parameters α and β . Table 5 illustrates RMS errors in relative humidity and temperature for model integrations during the month of November. All integrations were made with parameter $\alpha = 1$. The RMS errors obtained with model version 3.0

were 16.54% for the relative humidity and 2.64 deg for the temperature for $\beta = 0.94$.

Table 5. RMS error in model relative humidity and temperature as a function of the large-scale parameter β . Integrations were made during the month of November. Parameter $\alpha = 1$.

Parameter β	Relative Humidity [%]	Temperature [deg]
0.90	16.84	2.55
0.91	16.88	2.55
0.92	16.90	2.55
0.93	16.90	2.55
0.94	16.84	2.55
0.95	16.81	2.54
0.96	16.83	2.55
0.97	16.85	2.54
0.98	16.82	2.55
0.99	16.83	2.54

Table 6. RMS error in model relative humidity and temperature for two values of parameter α . Parameter $\beta = 0.97$.

Month	Parameter value	Relative Humidity [%]	Temperature [deg]
November	$\alpha = 0.5$	17.29	2.56
	$\alpha = 1.0$	17.29	2.58
December	$\alpha = 0.5$	15.75	2.77
	$\alpha = 1.0$	15.71	2.72
January	$\alpha = 0.5$	17.76	3.66
	$\alpha = 1.0$	17.83	3.65
February	$\alpha = 0.5$	21.67	3.50
	$\alpha = 1.0$	21.75	3.48

The sensitivity of the integrations to the parameter α is illustrated by the results in Table 6. Here two sets of integrations are made for each month: one with $\alpha = 1$. and another with $\alpha = 0.5$. The value of parameter β was 0.97.

Results from both tables show that traditional sensitivity of the model integrations to the large-scale parameters is negligible. The variation of the RMS errors is significantly smaller than the variation of the error from month to month.

5.3. Sensitivity to subgrid-scale parameters

To calculate adjoint sensitivity of the objective function to the model subgrid-scale parameters we integrated the model presented in Fig. 6. The relative adjoint sensitivities obtained during the 3-week-long integrations (Fig. 7[b]) are presented in Table 7. (Note here, that results from the February integrations need to be considered with some reservations with respect to the results of Fig. 7[b].) The convective parameters are ordered according to their values used in the present calculations. Only absolute values of the sensitivities are of interest since their sign should not be considered in the sensitivity analysis of radiative-convective equilibrium simulations (Hall *et al.*, 1982). It is the magnitude of the sensitivities that affects the parameter optimization procedure.

Several observations can be made from Table 7. First, the objective function is the most sensitive to the parameters which have the smallest absolute values. In particular, the parameter ELCRIT has at least 4 orders of magnitude larger influence on the relative humidity RMS than any other model parameter. Secondly, the objective function is differently sensitive to the terminal velocities of snow and rain. Sensitivity to the snow terminal

velocity, WTSNOW, is 3 orders of magnitude larger than sensitivity to the rain terminal velocity, WTRAIN. This result supports the importance of separation of the solid and liquid condensate in convective processes which was made in the model version 3.0. Thirdly, although the *traditional* sensitivity analysis (Section 5.2) shows small sensitivity of integrations to the large-scale parameters, α and β , the objective function is comparably sensitive to these parameters as it is sensitive to microphysical convective parameters such as COEFFR and COEFS.

Table 7. Relative sensitivity of the column relative humidity to convection parameters based on TOGA/COARE data. Units are RH[%] per parameter units.

Parameter value	November	December	January	February
ELCRIT = 0.00125	8.80e+2	5.76e+2	1.62e+3	3.28e+3
SIGD = 0.075	2.67e-3	6.27e-2	4.38e-2	8.67e-2
SIGS = 0.15	1.33e-3	3.13e-2	2.19e-2	4.33e-2
WTSNOW = 0.40	5.75e-4	6.75e-4	3.00e-4	5.50e-3
BETA = 0.97	1.44e-3	5.98e-4	2.99e-3	5.57e-4
COEFS = 0.7	3.86e-4	4.43e-4	3.85e-4	3.14e-3
ALPHA = 1.0	5.64e-4	6.00e-4	1.37e-3	1.46e-3
COEFFR = 1.0	1.71e-4	4.90e-4	3.10e-4	5.53e-4
WTRAIN = 5.5	2.54e-6	8.18e-6	4.98e-6	9.09e-6
TLCRIT = -20.	5.38e-6	5.50e-7	1.80e-6	9.89e-7

Comparative analysis of the relative adjoint sensitivities can be used for further model development and improvement. Physical interpretation of the sensitivities is crucial in this process, and can be used to detect model errors

(a)

(b)

Fig. 8 (a) Evolution of the relative humidity sensitivity to parameter ELCRIT during 4 independent integrations presented by the objective function in Fig. 7(b). (b) Same as (a) but for parameter TLCRIT.

Fig. 9 Same as Fig. 8 but for parameter SIGD.

or inadequacies of physical approximations (e.g. Živković *et al.*, 1995) as well as potential problems with data (Živković and Louis, 1996[a], [b]). Evolution of the sensitivities are also useful in interpretation of the model results. For example, evolution of relative humidity sensitivity to parameters ELCRIT and TLCRIT (see Table 2) in Fig. 8(a) and (b), respectively, clearly shows the discrete nature of the parameterization of autoconversion processes in convective cloud. If an abnormality of the evolution curves exists for these two parameters it will indicate possible inadequacy of parameterization of the autoconversion processes. Similarly, "the kink" common to the February curves for the parameters TLCRIT (Fig.8[b]) and BETA (Fig. 12[b]) indicates a potential problem with the data on 2/8. The BETA parameter is a large-scale parameter which controls "equilibration" of convection with large-scale flow. The parameter TLCRIT is a microphysical parameter, and the fact that it exhibited similar sensitivity growth as BETA indicates a response of the

Fig. 10 Same as Fig. 8 but for the parameters: (a) WTSNOW, and (b) WTRAIN

Fig. 11 Same as Fig. 8 but for the parameters: (a) COEFS, and (b) COEFFR.

Parameter ALPHA

(a)

Parameter BETA

(b)

Fig. 12 Same as Fig. 8 but for the parameters: (a) ALPHA, and (b) BETA .

affected processes to a large-scale forcing. Evolution diagrams for the rest of the parameters from Table 2 are shown in Figures 9 through 12.

6. Conclusion and Future Work

The convective scheme based on cloud microphysical considerations and suitable for use in climate models was developed under the separate grant to MIT. The task of the present project was to provide support in the scheme development, or more specifically to help with optimization of the scheme's convective parameters using real data. As a part of this effort, a tangent linear model (TLM) and its adjoint (ADJM) of the convective scheme were developed. The adjoint model was used to calculate adjoint sensitivities based on TOGA/ COARE data. The sensitivities were then used to facilitate finding of the parameter optimal values in the separate study.

Development of the TLM is the first step in developing the ADJM of any model. In this particular case, given the discrete nature of the parameterization of the convective processes in the models of general circulation, it was crucial to establish convergence and the accuracy level of the TLM as an approximation of evolution of the small perturbation around the nonlinear solution. The accuracy limits determined applicability of the associated ADJM and the validity of adjoint calculations.

Our work began with version 2.0 of the scheme which was developed under the previous DOE contract to MIT. This scheme was a part of a single column model which provided radiative-convective equilibrium solutions based on the GATE data (Emanuel, 1991; Rennó *et al.*, 1994). The extensive coding of the TLM and ADJM was done for this model version, 2.0. In the last year of the project a new 3.0 version of the scheme was introduced (Emanuel and Ž

ivković -Rothman, 1997). The new version differed from the previous version in several aspects: i) a quasi-equilibrium assumption for the rates of air mixing within the cloud was replaced by a local entrainment hypothesis, ii) a cloud base mass flux was relaxed toward a state of boundary layer quasi-equilibrium, iii) conversion of cloud water was loosely based on a stochastic coalescence model, iv) unsaturated downdrafts were allowed to enhance surface fluxes, and v) elevated convection was allowed. Given the time constraints of the project, these changes were incorporated into the TLM and ADJM codes with exception of the second modification. The modified nonlinear model was compared extensively with version 3.0 based on the real data. No substantial differences were found in the RMS errors, and this modified version of the code was numbered 2.3. Hence the adjoint application results were presented in this report for model version 2.3.

As a last part of the project, adjoint sensitivity of the model relative profile (RH) to model convective parameters was calculated by integrating the ADJM with the TOGA/COARE data. In the adjoint calculations, the so-called objective function, which represents model variable of interest, was set to RMS differences in RH between observations and calculations based on the convective scheme. The evolution of the objective function, during nonlinear calculations indicated some problems with the data during the last week of December, the last week of January, and the last three weeks of the entire data record. These data were eliminated from presented calculations, and four independent 3-week long integrations of the ADJ were performed. As shown by Rennó *et al.* (1994), these integrations were long enough to adequately represent radiative-convective equilibrium solutions.

As discussed by Emanuel and Živković-Rothman (1997), there are several possible reasons for inadequate data. The interpolated radiative fluxes might be inadequate, errors in calculated surface fluxes could be present, or adjusted ω profile might be contaminated by the nonuniform systematic errors that cannot be traced by the divergence field. The fact that the cloud mass flux was not relaxed toward a state of boundary-layer quasi-equilibrium in the adjoint calculations might have emphasized the problems with data as well.

Based on independent monthly integrations with the "good" data, relative adjoint sensitivities of the RH with respect to convective parameters were calculated. When compared, these values showed that the relative humidity profile predicted by the convection scheme is most sensitive to the microphysical parameters. In particular, the parameterization of the critical liquid water content for warm cloud autoconversion showed to be the most important. It was followed by the RH sensitivity with respect to updraft/downdraft mixing of the saturated and dry air, as well as to the terminal velocity of the snow. Sensitivity of RH to other convective parameters were comparable to the large-scale parameters that provide equilibration of the convection calculations with the large scale flow.

During the course of this work, several aspects of the adjoint technique were recognized to be beneficial for purposes of the scheme development. In addition to providing adjoint sensitivities, the rigorous testing of the ADJM applicability was very useful in determining critical parts of the nonlinear model (NLM), i.e. scheme, which could foster potential inadequacies in integrations with the real data. As a result, all three models, the NLM, TLM, and ADJM have been completely recoded, providing a consistent model

structure with properly formulated discontinuity points. Also, the interpretation of the TLM solutions provided very good insight into relationships of various hypotheses incorporated in the scheme. As such, the model adjoint is a very good tool for further development of the scheme and its applications with large-scale models.

Acknowledgment

This study was supported by the Department of Energy CHAMMP/CLIMATE Program under grants DE-FG02-94ER61926 and DE-FG02-91ER61220.

References

- Chou, M.-D., 1994: Radiation budgets in the Western Pacific, *J. Clim.*, **7**, 1958-1971.
- Chou, M.-D., W. Ridgway, and M.-H. Yan, 1995: Parameterizations for water vapor IR radiative transfer in the middle atmosphere, *J. Atmos. Sci.*, **52**, 1159-1167.
- Chou, M.-D., and K.-T. Lee 1996: Parameterizations for the absorption of solar radiation by water vapor and ozone, *J. Atmos. Sci.*, **53**, 1203-1208.
- Emanuel, K. A., 1991: A scheme for representing cumulus convection in large-scale models, *J. Atmos. Sci.*, **48**, 2313-2335.
- Emanuel, K. A. and M. Živković , 1997: A convective scheme for use in climate models, Part I: Development and Evaluation. (*to be submitted*)
- Errico, R. M., T. Vukić ević and K. Raeder, 1993: Examination of the accuracy of a tangent linear mode, *Tellus*, **45A**, 462-477.
- Hall, M. C. G., D. G. Cacuci and M. E. Schlesinger, 1982: Sensitivity analysis of a radiative convective model by the adjoint method, *J. Atmos. Sci.*, **39**, 2038-2050.
- Hall, M. C. G., 1986: Application of adjoint sensitivity theory to an atmospheric general convection model, *J. Atmos. Sci.*, **43**, 2644-2651.
- Lin, X., and R. H. Johnson, 1996: Kinematic and thermodynamic characteristics of the flow over the western Pacific warm pool during TOGA COARE, *J. Atmos. Sci.*, **53**, 695-715.

- Lacarra, J.-F. and O. Talagrand, 1988: Short-range evolution of small perturbation in a barotropic mode, *Tellus*, **40A**, 81-95.
- Louis, J.-F. and M. Živković, 1994: Optimizing a Local Weather Forecasting System. Final Report, NSF Grant No. ISI-9100720.
- Rennó, N. O., Emanuel, K. A., and P. H. Stone, 1994: Radiative-convective model with an explicit hydrological cycle, Part I: Formulation and sensitivity to model parameters, *J. Geophys. Res.*, **99**, 14,429-14,441.
- Talagrand, O. and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation - Part I. Theor, *Q. J. R. Meteorol. Soc.*, **113**, 1311-1328.
- Thépaut, J.-N. and P. Courtier, 1991: Four-dimensional variational data assimilation using the adjoint of a multilevel primitive-equation mode, *Q. J. R. Meteorol. Soc.*, **117**, 1225-1254.
- Velden, C. S., and J. A. Young, 1994: Satellite observations during TOGA COARE: large-scale descriptive overview. *Mon. Wea. Rev.*, **122**, 2426-2441.
- Vukićević, T. and R. M. Errico, 1993: Linearization and adjoint of parameterized moist diabatic processes, *Tellus*, **45A**, 493-510.
- Weller, R. A., and S. P. Anderson, 1996: Surface meteorology and air-sea fluxes in the western equatorial Pacific warm pool during TOGA coupled ocean-atmosphere response experiment, *J. Climate*, **9**, 1959-1990.
- Wergen, W., 1992: The effect of model errors in variational assimilation. *Tellus*, **44(A)**, 297-313.

- Xu, Q., 1996: Generalized adjoint for physical processes with parameterized discontinuities. Part I: Basic issues and heuristic examples, *J. Atmos. Sci.*, **53**, 1123-1142.
- Živković, M., J.-F. Louis and J.-L. Moncet 1995: Sensitivity analysis of a radiation parameterization, *J. Geoph. Res.*, **100**, D7, 13,827-13,840.
- Živković, M. and J.-F. Louis, 1996[a]: Subgrid-scale variability and data assimilation over CART observational facilities. Proc. 6th Atmospheric Radiation Measurement (ARM) Science Team Meeting, March 4-7, San Antonio, TX.
- Živković, M. and J.-F. Louis, 1996[b]: Single column variational assimilation experiments with ARM data. Proc. 6th Atmospheric Radiation Measurement (ARM) Science Team Meeting, March 4-7, San Antonio, TX.
- Živković, M. and K. A. Emanuel, 1997: A convective scheme for use in climate models, Part I: An adjoint model. (*in preparation*)
- Županski, D., 1993: The effects of discontinuities in the Betts-Miller cumulus convection scheme on four-dimensional variational data assimilation, *Tellus*, **45A**, 511-524.
- Županski, D. and F. Mesinger, 1995: Four-dimensional variational assimilation of precipitation data, *Mon. Wea. Rev.*, **123**, 1112-1127.

Appendix A. Nonlinear Code of Model Subroutine ENERGY

```
Subroutine ENERGY (T, R, P, PH, NA, ND, NDP1, NL,
  1 GZ, CPN, H, IFLAG)
C=====
C Version: @(#)energy.f 1.5 08/22/96
C=====
C This subroutine calculates geopotential, heat capacity
C and static energy.
C=====
C INPUT VARIABLES: R(NA) - specific humidity
C T(NA) - temperature
C P(NA) - pressure
C PH(NA) - layer pressure
C I/O PARAMETERS: NL - max number of model levels
C checked for moist convection.
C ND - dimension of model input arrays
C NA - dimension of convection arrays
C NDP1 - dimension of convection layers
C IFLAG - diagnostic flag
C OUTPUT VARIABLES: GZ(NA) - geopotential
C CPN(NA) - heat capacity
C H(NA) - static energy
C LOCAL VARIABLES: TVX(NA) dimension added by Marina.
C locals created by Marina: H1(NA)
C=====
```

```
REAL T(NA), R (NA), P(ND), PH(NDP1)
REAL GZ(NA), CPN(NA), H(NA)
REAL TVX(100), H1(100)
integer i
INTEGER IFLAG
COMMON / CONST/
C=====
C COMMON block for model constants.
1 CPD, CPV, CL, RV, RD, RLVO, TZERO, SPECQ,
2 CPVMCL, EPS, GINV, G, ROWL, CPDCPV, EPSILON
C=====
C === CALCULATE ARRAYS OF GEOPOTENTIAL, HEAT CAPACITY AND STATIC ENERGY
C
C Initialize output variables
do 35 i = 1,nl
  GZ(i) = 0.0
  CPN(i) = 0.0
  H(i) = 0.0
35 continue
C Initialize local variables
do 36 i = 1,100
  TVX(i) = 0.0
  H1(i) = 0.0
36 continue
361 CONTINUE
do 37 i = 1,nl
```

energy.f

```
TVX(I) = T(I)*(1.+R(I)/EPS-R(I))
37 continue
do 38 I = 1,NL
38 CONTINUE
do 39 I = 1,NL
  CPN(I) = CPD*(1.-R(I))+CPV*R(I)
  H1(I) = T(I)*CPN(I)
39 CONTINUE
C----- old code -----
C GZ(1) = 0.0
C CPN(1) = CPD*(1.-R(1))+CPV*R(1)
C H(1) = T(1)*CPN(1)
C do 40 I = 2,NL
C TVX = T(I)*(1.+R(I)/EPS-R(I))
C TVY = T(I-1)*(1.+R(I-1)/EPS-R(I-1))
C GZ(I) = GZ(I-1)+0.5*RD*(TVX+TVY)*(P(I-1)-P(I))/PH(I)
C CPN(I) = CPD*(1.-R(I))+CPV*R(I)
C H(I) = T(I)*CPN(I)+GZ(I)
C----- old code -----
C GZ(1) = 0.0
C H(1) = H1(1)
do 40 I = 2,NL
  GZ(I) = GZ(I-1) + 0.5*RD*(TVX(I)+TVY(I-1))
  * (P(I-1) - P(I))/PH(I)
  H(I) = H1(I) + GZ(I)
40 CONTINUE
C Variables no longer needed
50 CONTINUE
C === CHECK WHETHER LOWEST LEVEL TEMPERATURE AND SPECIFIC HUMIDITY
C ARE REASONABLE
C IF(T(1).LT.250.0).OR.(R(1).LE.0.0))THEN
  IFLAG = 0
  END IF
  RETURN
END
```

energy.f

Appendix B. General Code of Model Routine CONVECT

B-1 Generic Code

```
C=====
C      SUBROUTINE CONVECT
C      Hybrid VERSION 2.3
C      May 1, 1994
C=====
C      This is the version adjusted for writing the adjoint.
C=====
C      M. Zivkovic 10-24-94
C=====
C
C      <NLM> Subroutine CONVECT( T1, R1, RSI, P, PH, ND, NDP1, NL, DELT, IFLAG,
C      <NLM> * FT, FR, PRECIP, SIG, W0, MP, RP, NCOUNT, ICB)
C      <LTM> Subroutine LCONVECT( T1, R1, RSI, P, PH, ND, NDP1, NL, DELT, IFLAG,
C      <LTM> * FT, FR, PRECIP, SIG, W0, MP, RP, NCOUNT, ICB,
C      <LTM> * gT1, gR1, gRSI, gP, gPH, gFT, gFR, gPRECIP)
C      <ADU> Subroutine gCONVECT( T1, R1, RSI, P, PH, ND, NDP1, NL, DELT, IFLAG,
C      <ADU> * FT, FR, PRECIP, SIG, W0, MP, RP, NCOUNT, ICB,
C      <ADU> * gT1, gR1, gRSI, gP, gPH, gFT, gFR, gPRECIP)
C=====
C      %W% %G%
C=====
C
C      PARAMETER NA SHOULD IN GENERAL EQUAL ND.
C      NOPT IS AN INTEGER OTHER THAN 0, CONVECT
C      ARRAYS T AND R THAT MAY HAVE BEEN
C      BY DRY ADIABATIC ADJUSTMENT; OTHERWISE
C      RETURNED ARRAYS ARE UNALTERED.
C=====
C      Renamed variables:
C      LVO -----> RLVO
C
C      Added constants:
C      TZERO = 273.15
C      SPECQ = 0.608
C=====
C      M. Zivkovic 9-29-94.
C=====
C      <ALL> PARAMETER (NOPT=1, MAXLEV = 50, NA = MAXLEV)
C=====
C      <ALL> INTEGER NENT(MAXLEV), ICB, INB, JT(MAXLEV), I, J, nk
C      <ALL> INTEGER ICCB(MAXLEV), INNB(MAXLEV), IFLAG, NCOUNT
C=====
C      <ALL> REAL T1(ND), R1(ND), RSI(ND), P(ND), PH(NDP1)
C      <PER> REAL gT1(ND), gR1(ND), gRSI(ND), gP(ND), gPH(NDP1)
C      <ALL> REAL FT(ND), FR(ND), PRECIP
C      <PER> REAL gFT(ND), gFR(ND), gPRECIP
C      <ALL> REAL MP(ND), RP(ND), SIG(ND), W0(ND)
C      <PER> REAL gMP(MAXLEV), gRP(MAXLEV), gSIG(MAXLEV), gW0(MAXLEV)
C      <ALL> REAL SIG1(MAXLEV), W1(MAXLEV), TVP(MAXLEV), SIJ1(MAXLEV, MAXLEV)
C      <PER> REAL gSIG1(MAXLEV), gW1(MAXLEV), gTVP(MAXLEV), gSIJ1(MAXLEV, MAXLEV)
C=====
C      <ALL> REAL PBCRIT, PTCRIT, PPMIN
C      <ALL> REAL SIGD, SIGS, ALPHA, BETA, DTCCRIT
C      <ALL> REAL DELTI
C      <ALL> REAL ELCRIT, TCCRIT, SIGMD
C      <ALL> REAL WTRAIN, WTSNOW, COEFFR, COEFFS
C      <ALL> REAL DTMAX, SLFRAC, ENTP, VSO, DELT
C      <PER> REAL gPCRIT, gPTCRIT, gPPMIN
C      <PER> REAL gSIGD, gSIGS, gALPHA, gBETA, gDTCCRIT
C      <PER> REAL gDELTI
C      <PER> REAL gELCRIT, gTCCRIT, gSIGMD
C      <PER> REAL gWTRAIN, gWTSNOW, gCOEFFR, gCOEFFS
```

con.f

```
<PER> REAL gDTMAX, gSLFRAC, gENTP, gVSO
C=====
C      <ALL> REAL T(MAXLEV), R(MAXLEV), RS(MAXLEV), TH(MAXLEV)
C      <PER> REAL gT(MAXLEV), gR(MAXLEV), gRS(MAXLEV), gTH(MAXLEV)
C      <ALL> REAL GT(MAXLEV), CPN(MAXLEV), H(MAXLEV)
C      <PER> REAL gGT(MAXLEV), gCPN(MAXLEV), gH(MAXLEV)
C      <ALL> REAL SIGP(MAXLEV), EP(MAXLEV)
C      <PER> REAL gSIGP(MAXLEV), gEP(MAXLEV)
C      <ALL> REAL TV(MAXLEV), CLW(MAXLEV)
C      <PER> REAL gTV(MAXLEV), gCLW(MAXLEV)
C      <ALL> REAL TAU(MAXLEV), CAPE(MAXLEV), DTMIN(MAXLEV), HP1(MAXLEV)
C      <PER> REAL gTAU(MAXLEV), gCAPE(MAXLEV), gDTMIN(MAXLEV), gHP1(MAXLEV)
C      <ALL> REAL M(MAXLEV), LV(MAXLEV), LVCP(MAXLEV)
C      <PER> REAL gM(MAXLEV), gLV(MAXLEV), gLVCP(MAXLEV)
C      <ALL> REAL gM(MAXLEV), gLV(MAXLEV), gELI11(MAXLEV, MAXLEV)
C      <PER> REAL gM1(MAXLEV, MAXLEV), gELI111(MAXLEV, MAXLEV)
C      <ALL> REAL gM21(MAXLEV, MAXLEV), gQENT1(MAXLEV, MAXLEV)
C      <PER> REAL gM211(MAXLEV, MAXLEV), gQENT11(MAXLEV, MAXLEV)
C      <ALL> REAL ASIJ(MAXLEV), DELP(MAXLEV, MAXLEV), DELM(MAXLEV, MAXLEV)
C      <PER> REAL gASIJ(MAXLEV)
C=====
C      <ALL> REAL ASUM, BSUM, CSUM, DSUM
C      <PER> REAL gASUM, gBSUM, gCSUM, gDSUM
C=====
C      <ALL> REAL EVAP(MAXLEV)
C      <PER> REAL gEVAP(MAXLEV)
C      <ALL> REAL FT1(MAXLEV), FR1(MAXLEV), TWORK(MAXLEV), RWORK(MAXLEV)
C      <PER> REAL gFT1(MAXLEV), gFR1(MAXLEV), gTWORK(MAXLEV), gRWORK(MAXLEV)
C      <ALL> REAL WT(MAXLEV), RPI(MAXLEV), WATER(MAXLEV)
C      <PER> REAL gWT(MAXLEV), gRPI(MAXLEV), gWATER(MAXLEV)
C=====
C      <ALL> COMMON / CONST /
C
C      COMMON block for convect constants
C      <ALL>1 CPD, CPV, CL, RV, RD, RLVO, TZERO, SPECQ,
C      <ALL>2 CPVNCL, EPS, GINV, G, ROWL, CPDCPV, EPSILON
C=====
C      <ALL> COMMON / INVAR /
C
C      COMMON block for variables specified outside the convect
C      <ALL>1 PBCRIT, PTCRIT, PPMIN,
C      <ALL>2 SIGD, SIGS, ALPHA, BETA, DTCCRIT,
C      <ALL>3 DELTI,
C      <ALL>4 ELCRIT, TCCRIT, SIGMD,
C      <ALL>5 WTRAIN, WTSNOW, COEFFR, COEFFS,
C      <ALL>6 DTMAX, SLFRAC, ENTP, VSO
C=====
C      <PER> COMMON /GINVAR/
C
C      <PER>1 gPCRIT, gPTCRIT, gPPMIN,
C      <PER>2 gSIGD, gSIGS, gALPHA, gBETA, gDTCCRIT,
C      <PER>3 gDELTI,
C      <PER>4 gELCRIT, gTCCRIT, gSIGMD,
C      <PER>5 gWTRAIN, gWTSNOW, gCOEFFR, gCOEFFS,
C      <PER>6 gDTMAX, gSLFRAC, gENTP, gVSO
C=====
C      <ALL> IF(ND .gt. MAXLEV) write(*,*) "Incompatible local array in CONVECT"
C      Initialize all output variables
C      <ALL> CALL RZERO ( FT, FR, PRECIP, MP, RP, ND)
C      <LTM> CALL LRZERO (gFT, gFR, gPRECIP, gMP, gRP, ND)
```

con.f

```
<ADJ> CALL GRZERO (GFT,GFR,GPRECIP,GMP,GRP,ND)
```

```
c ... Initialize arrays and constants
```

```
<ALL> CALL INIT( T1, R1, RS1, P, NA, ND, NL, DELT,
  T, R, RS, TH, FT, FR, PRECIP, IFLAG)
<ALL>* CALL LINIT(T1, R1, RS1, P, NA, ND, NL, DELT,
  T, R, RS, TH, FT, FR, PRECIP,
  GT1,GR1,GRS1,GP,DELTA,GT,GR,GRS,GT,GR,GFR,GPRECIP)
<ADJ> CALL GINIT (T1, R1, RS1, P, NA, ND, NL, DELT,
  T, R, RS, TH, FT, FR, PRECIP,
  GT1,GR1,GRS1,GP,DELTA,GT,GR,GRS,GT,GR,GFR,GPRECIP)
```

```
c Increment the counter
```

```
<ALL> NCOUNT = NCOUNT+1
<ALL> NCOUNT = MIN(NCOUNT,13)
```

```
c Do not bypass this even if the calling
c program has boundary layer scheme!!!
c ... Perform dry adiabatic adjustment.
```

```
<NLM> CALL DRYADJ (T1, R1, RS1, P, ND, NL, NA, NDP1, T, R, RS, TH, PH, NOPT)
```

```
c ... Calculate geopotential, heat capacity
c and static energy
```

```
<ALL> CALL ENERGY (T, R, P, PH, NA, ND, NDP1, NL, GZ, CPN, H, IFLAG)
<ALL> CALL LENERGY (T, R, P, PH, NA, ND, NDP1, NL, GZ, CPN, H,
  IFLAG)
<LTM>* GT,GR,GP,GPH,GGZ, GCPN, GH)
<ADJ> CALL GENERGY (T, R, P, PH, NA, ND, NDP1, NL, GZ, CPN, H,
  GT,GR,GP,GPH,GGZ, GCPN, GH)
```

```
c Start loop through parcel origin levels
```

```
<ALL> DO 800 NK = 1,NL-1
  DO 980 I = 1,NL
    HPI(I) = H(I)
    RP(I) = R(I)
    WT(I) = 0.001
    WENT(I) = 0
    WATER(I) = 0.0
    EVAP(I) = 0.0
    MP(I) = 0.0
    M(I) = 0.0
    LV(I) = RLVO-CPVMCL* (T(I)-TZERO)
    LVCP(I) = LV(I)/CPN(I)
    DO 981 J = 1,NL
      QENT(I,J) = R(J)
      ELIJI(I,J) = 0.0
      MENT1(I,J) = 0.0
      SIJI(I,J) = 0.0
    CONTINUE
  981 CONTINUE
  980 CONTINUE
```

```
c ... Convect only from top of mixed layers
c and only if pressure is greater than PPMIN
<ALL> IF (TH(NK).GT.(TH(NK+1)-0.1).OR.P(NK).LT.PPMIN) GOTO 800
c ... Find the first level above the cloud base, ICB.
```

```
<NLM> CALL CLBASE ( P, T, R, TH, RS, NL, NA, ND, NK, ICCB, IFLAG)
c Use stored values for ICB
```

conf

```
<NLM> ICB = ICCB(NK)
```

```
<ALL> IF (IFLAG.EQ. 3) GO TO 800
```

```
c ... Calculate virtual temperature and parcel virtual
c temperature below and above the cloud base. Also,
c test for instability below ICB.
```

```
<ALL> CALL VIRT ( P, T, R, RS, CLW, GZ, NL, NA, ND, NK,
  EP, TV, TVP, SIGP, ICB, IFLAG)
<ALL>* IF (TH(NK).GT.(TH(NK+1)-0.1).OR.P(NK).LT.PPMIN.or.IFLAG.EQ.0) GO TO 800
```

```
c ... If this point is reached, moist convective adjustment
c is necessary.
```

```
<ALL> IF (IFLAG.NE.4) IFLAG=1
```

```
c ... Find the first level above the parcel's level
c of neutral buoyancy, INB.
```

```
<ALL> CALL CLTOP ( TVP, TV, ICB, NL, NA, NK, INNB)
```

```
<ALL> INB = INNB(NK)
```

```
c ... Calculate time scales, TAU, for air
c in convective clouds to reach level I.
```

```
<ALL> CALL TAUCL( W0, GZ, NL, NA, ND, ICB, INB, TAU)
```

```
c ... Set values for fractional area, SIG, and
c vertical velocity, W0, below and above the cloud.
```

```
<ALL> CALL SIGOUTCL( TV, TVP, TAU, SIG, W0, NA, ND, NL, INB, ICB, W1, SIG1)
<ALL> CALL SIGINCL ( P, PH, TV, TVP, TAU, W0, SIG, NA, ND, NL, NDP1,
  INB, ICB, CAPE, W1, SIG1, M, DTMIN)
```

```
c ... Calculate saturated updrafts and downdrafts.
```

```
<ALL> CALL ENTRAIN ( H, T, R, RS, LV, EP, M, CLW, SIG1,
  NA, ND, NK, ICB, INB, NENT, MENT1, QENT1,
  SIJ1, ELIJI, DELP, DELM, ASIJ, HPI)
```

```
<ALL> DO 982 I = 1,NL
```

```
  DO 983 J = 1,NL
```

```
    IF(QENT1(I,J).NE.0.0) THEN
```

```
      QENT(I,J) = QENT1(I,J)
```

```
    ENDIF
```

```
  983 CONTINUE
```

```
  982 CONTINUE
```

```
c ... Calculate precipitation if EP(INB) .NE. 0.
```

```
<ALL> IF (EP(INB).GE.0.0001) THEN
  CALL PREC ( H, P, PH, T, R, RS, TV, NL, NA, ND, NDP1, INB,
  LV, EP, GZ, M, CLW, SIGP, MENT1, WATER,
  WT, MP, ELIJI, EVAP, PRECIP, RPI, JT)
```

```
<ALL>* ENDIF
```

```
c ... Calculate temperature and moisture tendencies
```

conf


```

c      Variables SIG1, W1, RP1, FT1, FR1 no longer needed
<ADJ>  GSIG1(I) = 0.0
<ADJ>  GW1(I) = 0.0
<ADJ>  GRP1(I) = 0.0
<ADJ>  GFT1(I) = 0.0
<ADJ>  GFR1(I) = 0.0
      100 CONTINUE
<PER> ENDIF
C-----
      RETURN
      END

```

B-2 Nonlinear Code

```

C=====
C----- SUBROUTINE CONVECT
C----- Hybrid VERSION 2.3
C----- May 1, 1994
C-----
C----- This is the version adjusted for writing the adjoint.
C----- M. Zivkovic 10-24-94
C-----
C
C Subroutine CONVECT( T1, R1, RS1, P, PH, ND, NDP1, NL, DELT, IFLAG,
* FT, FR, PRECIP, SIG, WO, MP, RP, NCOUNT, ICE)
C-----
C      %W% %G%
C-----
C
C PARAMETER NA SHOULD IN GENERAL EQUAL ND.
C
C NOT IS AN INTEGER OTHER THAN 0, CONVECT
C ARRAYS T AND R THAT MAY HAVE BEEN
C BY DRY ADIABATIC ADJUSTMENT; OTHERWISE
C RETURNED ARRAYS ARE UNALTERED.
C
C-----
C Renamed variables:
C      LVO -----> RLVO
C
C Added constants:
C      TZERO = 273.15
C      SPECQ = 0.608
C
C-----
C M. Zivkovic 9-29-94.
C-----
C PARAMETER (NOPT=1, MAXLEV = 50, NA = MAXLEV)
C-----
C INTEGER NENT(MAXLEV), ICB, INB, JT(MAXLEV), I, J, nk
C INTEGER ICCB(MAXLEV), INNB(MAXLEV), IFLAG, NCOUNT
C
C REAL T1(ND), R1(ND), RS1(ND), P(ND), PH(NDP1)
C REAL FT(ND), FR(ND), PRECIP
C REAL MP(ND), RP(ND), SIG(ND), WO(ND)
C REAL SIG1(MAXLEV), WI(MAXLEV), TVP(MAXLEV), SIJ1(MAXLEV, MAXLEV)
C
C REAL PBCRIT, PTCRIT, PPMIN
C REAL SIGD, SIGS, ALPHA, BETA, DTCRIT
C REAL DELT1
C REAL ELCRIT, T1CRIT, SIGMD
C REAL WTRAIN, WTSNOW, COEFFR, COEFFS
C REAL DTMAX, SLFRAC, ENTP, VSO, DELT
C
C REAL T(MAXLEV), R(MAXLEV), RS(MAXLEV), TH(MAXLEV)
C REAL GZ(MAXLEV), CPN(MAXLEV), H(MAXLEV)
C REAL SIGP(MAXLEV), EP(MAXLEV)
C REAL TV(MAXLEV), CLW(MAXLEV)
C REAL TAU(MAXLEV), CAPE(MAXLEV), DTMIN(MAXLEV), HP1(MAXLEV)
C REAL M(MAXLEV), LV(MAXLEV), LVCP(MAXLEV)
C REAL QENT(MAXLEV, MAXLEV), ELIJ1(MAXLEV, MAXLEV)
C REAL MENT1(MAXLEV, MAXLEV), QENT1(MAXLEV, MAXLEV)
C REAL ASIJ(MAXLEV), DELP(MAXLEV, MAXLEV), DELM(MAXLEV, MAXLEV)
C
C-----
C REAL ASUM, BSUM, CSUM, DSUM
C-----
C
C EVAP(MAXLEV)
C REAL FT1(MAXLEV), FR1(MAXLEV), TWORK(MAXLEV), RWORX(MAXLEV)

```

```

REAL WT(MAXLEV), RPI(MAXLEV), WATER(MAXLEV)
COMMON / CONST/
-----
COMMON block for convect constants
1 CPD, CPV, CL, RV, RD, RLVO, TZERO, SPECQ,
2 CPVMCL, EPS, GINV, G, ROWL, CPDCPV, EPSILON
COMMON /INVAR/
-----
COMMON block for variables specified outside the convect
1 PBCRIT, PTCRIT, PPMIN,
2 SIGD, SIGS, ALPHA, BETA, DTCRIT,
3 DELT1,
4 ELCRIT, TLCRIT, SIGMD,
5 WTRAIN, WTSNOW, COEFFR, COEFP,
6 DTMAX, SLFRAC, ENTP, VSO
-----
C
IF(ND.GT.MAXLEV) write(*,*) "Incompatible local array in CONVECT"
Initialize all output variables
CALL RZERO ( FT, FR, PRECIP, MP, RP, ND)
C
C ... Initialize arrays and constants
-----
CALL INIT( T1, R1, RS1, P, NA, ND, NL, DELT,
* T, R, RS, TH, FT, FR, PRECIP, IFLAG)
C
C Increment the counter
NCOUNT = NCOUNT+1
NCOUNT = MIN(NCOUNT,13)
C
C Do not bypass this even if the calling
C program has boundary layer scheme!!!
C ... Perform dry adiabatic adjustment.
-----
CALL DRVADJ (T1, R1, RS1, P, NA, ND, NL, NA, NDP1, T, R, RS, TH, PH, NOPT)
C
C ... Calculate geopotential, heat capacity
C and static energy
-----
CALL ENERGY (T, R, P, PH, NA, ND, NDP1, NL, GZ, CPN, H, IFLAG)
C
C Start loop through parcel origin levels
DO 800 NK = 1, NL-1
DO 980 I = 1, NL
HPI(I) = H(I)
RPI(I) = R(I)
WT(I) = 0.001
NENT(I) = 0
WATER(I) = 0.0
EVAP(I) = 0.0
MP(I) = 0.0
M(I) = 0.0
LV(I) = RLVO-CPVMCL* (T(I)-TZERO)
LVCP(I) = LV(I)/CPN(I)
DO 981 J = 1, NL
QENT(I,J) = R(J)
ELIJI(I,J) = 0.0

```

ncon.f

ncon.f

ncon.f

ncon.f

B-3 Adjoint Code

```

C=====
C SUBROUTINE CONVECT
C Hybrid VERSION 2.3
C May 1, 1994
C=====
C This is the version adjusted for writing the adjoint.
C M. Zivkovic 10-24-94
C=====
C
C Subroutine gCONVECT( T1, R1, RS1, P, PH, ND, NDP1, NL, DELT, IFLAG,
C * FT, FR, PRECIP, SIG, W0, MP, RP, NCOUNT, ICB,
C * GT1, GR1, GRS1, GP, GPH, GFT, GFR, GPRECIP)
C
C %* %* %*
C=====
C
C PARAMETER NA SHOULD IN GENERAL EQUAL ND.
C NOPT IS AN INTEGER OTHER THAN 0, CONVECT
C ARRAYS T AND R THAT MAY HAVE BEEN
C BY DRY ADIABATIC ADJUSTMENT; OTHERWISE
C RETURNED ARRAYS ARE UNALTERED.
C
C=====
C Renamed variables:
C LV0 -----> RLVO
C
C Added constants:
C TZERO = 273.15
C SPECQ = 0.608
C
C=====
C M. Zivkovic 9-29-94.
C
C PARAMETER (NOPT=1, MAXLEV = 50, NA = MAXLEV)
C
C INTEGER NENT(MAXLEV), ICB, INB, JT(MAXLEV), I, J, nk
C INTEGER ICCB(MAXLEV), INNB(MAXLEV), IFLAG, NCOUNT
C
C REAL T1(ND), R1(ND), RS1(ND), P(ND), PH(NDP1)
C REAL GT1(ND), GR1(ND), GRS1(ND), GP(ND), GPH(NDP1)
C REAL FT(ND), FR(ND), PRECIP
C REAL GFT(ND), GFR(ND), GPRECIP
C REAL MP(ND), RP(ND), SIG(ND), W0(ND)
C REAL GMP(MAXLEV), GRP(MAXLEV), GSIG(MAXLEV), GW0(MAXLEV)
C REAL SIG1(MAXLEV), W1(MAXLEV), TVP(MAXLEV), SIJ1(MAXLEV, MAXLEV)
C REAL GSIG1(MAXLEV), GW1(MAXLEV), GTVP(MAXLEV), GSIJ1(MAXLEV, MAXLEV)
C
C REAL PBCRIT, PTCRIT, PPMIN
C REAL SIGD, SIGS, ALPHA, BETA, DTCCRIT
C REAL DELTI
C REAL ELCRIT, T1CRIT, SIGMD
C REAL WTRAIN, WTSNOW, COEFFR, COEFFS
C REAL DTMAX, SLFRAC, ENTP, VS0, DELT
C REAL GBCRIT, GPTCRIT, GPPMIN
C REAL GSIGD, GSIGS, GALPHA, GBETA, GDTCCRIT
C REAL GDELT1
C REAL GELCRIT, GT1CRIT, GSIGMD
C REAL GWTRAIN, GWTSNOW, GWCOEFFR, GWCOEFFS
C REAL GDTMAX, GSLFRAC, GENTP, GVS0
C
C REAL T(MAXLEV), R(MAXLEV), RS(MAXLEV), TH(MAXLEV)
C REAL GT(MAXLEV), GR(MAXLEV), GRS(MAXLEV), GTH(MAXLEV)
C REAL GZ(MAXLEV), CPN(MAXLEV), H(MAXLEV)

```

gcon.f

```

C
C REAL GZ(MAXLEV), GCPN(MAXLEV), GH(MAXLEV)
C REAL SIGP(MAXLEV), EP(MAXLEV)
C REAL GSIGP(MAXLEV), GEP(MAXLEV)
C REAL TV(MAXLEV), CLW(MAXLEV)
C REAL GTV(MAXLEV), GCLW(MAXLEV)
C REAL TAU(MAXLEV), CAPE(MAXLEV), DTMIN(MAXLEV), HP1(MAXLEV)
C REAL GTAU(MAXLEV), GCAPE(MAXLEV), GDTMIN(MAXLEV), GH1(MAXLEV)
C REAL M(MAXLEV), LV(MAXLEV), LVCP(MAXLEV)
C REAL GM(MAXLEV), GLV(MAXLEV), GLVCP(MAXLEV)
C REAL QENT(MAXLEV, MAXLEV), ELIJI(MAXLEV, MAXLEV)
C REAL GQENT(MAXLEV, MAXLEV), GELIJI(MAXLEV, MAXLEV)
C REAL MENT1(MAXLEV, MAXLEV), QENTI(MAXLEV, MAXLEV)
C REAL GMENT1(MAXLEV, MAXLEV), GQENTI(MAXLEV, MAXLEV)
C REAL AS1J(MAXLEV), DELP(MAXLEV), DELM(MAXLEV, MAXLEV)
C REAL GAS1J(MAXLEV)
C
C-----
C REAL ASUM, BSUM, CSUM, DSUM
C REAL GASUM, GBSUM, GCSUM, GDSUM
C
C REAL EVAP(MAXLEV)
C REAL GEVAP(MAXLEV)
C REAL FT1(MAXLEV), FR1(MAXLEV), TWORK(MAXLEV), RWORK(MAXLEV)
C REAL GFT1(MAXLEV), GFR1(MAXLEV), GTWORK(MAXLEV), GRWORK(MAXLEV)
C REAL WT(MAXLEV), RP1(MAXLEV), WATER(MAXLEV)
C REAL GWT(MAXLEV), GRP1(MAXLEV), GWATER(MAXLEV)
C
C COMMON / CONST/
C
C COMMON block for convect constants
C 1 CPD, CPV, CL, RV, RD, RLVO, TZERO, SPECQ,
C 2 CPVMCL, EPS, GINV, G, ROWL, CPDPCV, EPSILON
C
C COMMON / INVAR/
C
C COMMON block for variables specified outside the convect
C 1 PBCRIT, PTCRIT, PPMIN
C 2 SIGD, SIGS, ALPHA, BETA, DTCCRIT,
C 3 GDELT1,
C 4 ELCRIT, T1CRIT, SIGMD,
C 5 WTRAIN, WTSNOW, COEFFR, COEFFS,
C 6 DTMAX, SLFRAC, ENTP, VS0
C
C COMMON /GINVAR/
C
C 1 GBCRIT, GPTCRIT, GPPMIN,
C 2 GSIGD, GSIGS, GALPHA, GBETA, GDTCCRIT,
C 3 GDELT1,
C 4 GELCRIT, GT1CRIT, GSIGMD,
C 5 GWTRAIN, GWTSNOW, GWCOEFFR, GWCOEFFS,
C 6 GDTMAX, GSLFRAC, GENTP, GVS0
C
C IF(ND.GT. MAXLEV) write(*,*) "Incompatible local array in CONVECT"
C CALL RZERO ( FT, FR, PRECIP, MP, RP, ND)
C CALL INIT( T1, R1, RS1, P, NA, ND, NL, DELT,
C * T, R, RS, TH, FT, FR, PRECIP, IFLAG)
C NCOUNT = NCOUNT+1
C NCOUNT = MIN(NCOUNT, 13)
C CALL ENERGY (T, R, P, PH, NA, ND, NDP1, NL, GZ, CPN, H, IFLAG)
C DO 800 NK = 1, NL-1
C DO 980 I = 1, NL

```

gcon.f

```

881      HP1(I) = H(I)
      RP(I) = R(I)
      WT(I) = 0.001
      NENT(I) = 0
      WATER(I) = 0.0
      EVAP(I) = 0.0
      MP(I) = 0.0
      M(I) = 0.0
      LV(I) = RLVO-CPVMCL* (T(I)-TZERO)
      LVCPI(I) = LV(I)/CPN(I)
      DO 981 J = 1,NL
        QENT(I,J) = R(J)
        ELIJI(I,J) = 0.0
        MENT1(I,J) = 0.0
        SIJI(I,J) = 0.0
      CONTINUE
      CALL VIRT ( P, T, R, RS, CLW, GZ, NL, NA, ND, NK,
        EP, TV, TVP, SIGP, ICB, IFLAG)
      IF (TH(NK).GT.(TH(NK+1)-0.1).OR.P(NK).LT.PPMIN.OF.IFLAG.EQ.0) GO TO 800
      CALL CLTOP ( TVP, TV, ICB, NL, NA, NK, INNB)
      INB = INNB(NK)
      CALL TAUCL( WO, GZ, NL, NA, ND, ICB, INB, TAU)
      CALL SIGOUTCL( TV, TVP, TAU, SIG, WO, NA, ND, NL, ICB, W1, SIG1)
      CALL SIGINCL ( P, PH, TV, TVP, TAU, WO, SIG, NA, ND, NL, NDPI,
        INB, ICB, CAPE, W1, SIG1, M, DTMIN)
      CALL ENTRAIN ( H, T, R, RS, LV, EP, M, CLW, SIG1,
        NA, ND, NK, ICB, INB, NENT, MENT1, QENT1,
        SIJI, ELIJI, DELP, DELM, ASIJ, HP1)
      DO 982 I = 1,NL
        DO 983 J = 1,NL
          IF (QENT1(I,J) .NE. 0.0) THEN
            QENT(I,J) = QENT1(I,J)
          ENDIF
        CONTINUE
      CONTINUE
      IF (EP(INB) .GE. 0.0001) THEN
        CALL PREC ( H, P, PH, T, R, RS, TV, NL, NA, ND, NDPI, INB,
          LV, EP, GZ, M, CLW, SIGP, MENT1, WATER,
          WT, MP, ELIJI, EVAP, PRECIP, RPI, JT)
      ENDIF
      CALL TEND (H, HP1, T, TH, R, PH, RPI, NK, NA, ND, NDPI, INB, ICB,
        M, EP, GZ, WT, LV, LVCP, CLW, MENT1, QENT, ASUM, BSUM,
        CSUM, DSUM, WATER, MP, ELIJI, EVAP, CPN, FTL, FRI, IFLAG,
        RWORK, TWORK)
      NCOUNT = 2
      GO TO 801
800 CONTINUE
      DO 802 I=1,1
        write(*,*) 'Something is wrong with ADJ generator'
        write(*,*) 'if this line is printed!'
      CONTINUE
801 CONTINUE
      IF (NK .LT. (NL-1)) THEN
        DO 100 I = 1,NL
          GPR1(I) = 0.0
          GFT1(I) = 0.0
          GRP1(I) = 0.0
          GW1(I) = 0.0
        CONTINUE

```

gcon.f

```

      GSIG1(I) = 0.0
      GPR1(I) = GPR1(I) + GPR (I)
      GFT1(I) = GFT1(I) + GFT (I)
      GRP1(I) = 0.0
      GW0(I) = 0.0
      GSIG(I) = 0.0
      CONTINUE
      CALL GTEND (H, HP1, T, TH, R, PH, RPI, NK, NA, ND, NDPI, INB, ICB,
        M, EP, GZ, WT, LV, LVCP, CLW, MENT1, QENT, ASUM, BSUM,
        CSUM, DSUM, WATER, MP, ELIJI, EVAP, CPN, FTL, FRI,
        RWORK, TWORK, GWORK, GTWORK, GH, GHPI, GT, GTH, GR, GPH, GRPI,
        GM, GEP, GGZ, GWT, GLV, GLVCP, GCLW, GMENT1, QOENT, GASUM, GBSUM,
        GCSUM, GDSUM, GWATER, GMP, GELIJI, GEVAP, GCPN, GFT1, GPR1)
      IF (EP(INB) .GE. 0.0001) THEN
        CALL GPREC ( H, P, PH, T, R, RS, TV, NL, NA, ND, NDPI, INB,
          LV, EP, GZ, M, CLW, SIGP, MENT1, WATER,
          WT, MP, ELIJI, EVAP, PRECIP, RPI, JT)
        GH, GP, GPH, GT, GR, GRS, GTV, GLV, GEP, GGZ, GM, GCLW,
        GSIGP, GMENT1, GWATER, GWT, GMP, GELIJI, GEVAP, GPRECIP, GRPI)
      ENDIF
      DO 970 I = 1,NL
        DO 971 J = 1,NL
          IF (QENT1(I,J) .NE. 0.0) THEN
            QENT1(I,J) = QOENT1(I,J) + QOENT(I,J)
            QOENT(I,J) = 0.0
          ENDIF
        CONTINUE
      CONTINUE
      CALL GENTRAIN ( H, T, R, RS, LV, EP, M, CLW, SIG1,
        NA, ND, NK, ICB, INB, NENT, MENT1, QOENT1,
        SIJI, ELIJI, DELP, DELM, ASIJ, HP1,
        GH, GT, GR, GRS, GLV, GEP, GM, GCLW, GSIG1,
        GMENT1, QOENT1, GSIG1, GELIJI, GASIJ, GHPI)
      CALL GSIGINCL ( P, PH, TV, TVP, TAU, WO, SIG, NA, ND, NL, NDPI,
        INB, ICB, CAPE, W1, SIG1, M, DTMIN, GDTMIN,
        GP, GPH, GTV, GTVP, GTAU, GW0, GSIG, GCAPE, GW1, GSIG1, GM)
      CALL GSIGOUTCL( TV, TVP, TAU, SIG, WO, NA, ND, NL, INB, ICB, W1, SIG1,
        GTV, GTVP, GTAU, GSIG, GW0, GW1, GSIG1)
      CALL GTAUCL( WO, GZ, NL, NA, ND, ICB, INB, TAU, GW0, GGZ, GTAU)
      ENDIF
      CALL GVIRT ( P, T, R, RS, CLW, GZ, NL, NA, ND, NK,
        EP, TV, TVP, SIGP, ICB,
        GP, GT, GR, GRS, GCLW, GGZ, GEP, GTV, GTVP, GSIGP)
      DO 984 I = 1,NL
        DO 985 J = 1,NL
          GSIG1(I,J) = 0.0
          GMENT1(I,J) = 0.0
          GELIJI(I,J) = 0.0
          GR(J) = GR(J) + QOENT(I,J)
        CONTINUE
      CONTINUE
      GCPN(I) = GCPN(I) - GLVCP(I)* LV(I)/(CPN(I)*CPN(I))
      GLV(I) = GLV(I) + GLVCP(I)/CPN(I)
      GT(I) = GT(I) - CPVMCL*GLV(I)
      GM(I) = 0.0
      GMP(I) = 0.0
      GEVAP(I) = 0.0
      GWATER(I) = 0.0
      GR(I) = GR(I) + GRP(I)
      GWT(I) = 0.0
      GH(I) = GH(I) + GHPI(I)

```

gcon.f

984 CONTINUE
CALL GENERGY (T, R, P, PH, NA, ND, NDE1, NL, GZ, CPN, H,
* GT, GR, GP, GPH, GGZ, GCPN, GH)
CALL GINIT (TI, RI, RS1, P, NA, ND, NL, DELT,
* T, R, RS, TH, FT, FR, PRECIP,
* GTI, GRI, GRS1, GP, GDELT, GT, GR, GRS, GTH, GFT, GFR, GPRECIP)
CALL GRZERO (GFT, GFR, GPRECIP, GMP, GRP, ND)

RETURN
END