AN IMPROVED NUMERICAL MODEL FOR CALCULATIONS OF TRANSPORT AND SIZE DISTRIBUTIONS OF ATMOSPHERIC AEROSOLS AND CLOUD DROPLETS

by

ROALD AKBEROV

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

written under the direction of

Prof. Georgiy Stenchikov

and approved by

_____

_____

_____

New Brunswick, New Jersey

January, 2008

ABSTRACT OF THE THESIS

AN IMPROVED NUMERICAL MODEL FOR CALCULATIONS OF TRANSPORT AND SIZE DISTRIBUTIONS OF ATMOSPHERIC AEROSOLS AND CLOUD DROPLETS

by ROALD AKBEROV

Thesis Director: Prof. Georgiy Stenchikov

Aerosols and cloud droplets in the atmosphere appear in different sizes. They can be transported by wind and experience various microphysical changes due to nucleation, collisional and condensational growth, evaporation and sedimentation. These microphysical processes affect optical properties of clouds and aerosols, and through it, they affect climate. Climate models usually lack a microphysical module, but instead use various parameterizations to represent clouds in the terrestrial atmosphere. Coupling between cloud microphysical models and climate models could improve the quality of numerical predictions of climate. In this study, CARMA (Community Aerosol and Radiation Model for Atmospheres), which is one of the best available microphysical models for calculation of size distributions of atmospheric aerosols and cloud droplets based on environmental conditions, has been redesigned and rewritten for coupling with numerical models of climate. In the revised model, data transfer between subroutines is handled via lists of arguments. The microphysical part of CARMA v. 2.2 was isolated from other processes, such as radiation and transport, and prepared for coupling with existing climate models. Changes were made to the vertical transport subroutines. The PPM (Piecewise Parabolic Method) method of solving the advection equation was

replaced with the REA (Reconstruct-Evolve-Average) method, complemented with the minmod slope limiter. The sedimentation equation is now solved using the upwind method. Sedimentation fluxes, used by the upwind method, are calculated inside the microphysical part of CARMA. The ability of the revised CARMA model to reproduce the observed microstructure of a marine stratocumulus cloud over the North Sea was tested. The improved model closely reproduces most of the observed properties of the cloud and aerosols.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Georgiy Stenchikov for his financial support throughout the course of this study. Thanks to his invitation to work on this project, I was able to extend my former education in mechanical and aerospace engineering toward geophysical sciences; especially toward atmospheric science, oceanography and climatology. I would also extend my thanks to Dr. Tony Broccoli of the Environmental Sciences Department for giving me an inspiration and indispensable guidance in learning synoptic meteorology while I was taking his course on Large Scale Weather Systems. I also acknowledge Dr. Alan Robock of the Environmental Sciences Department, Dr. Dale Haidvogel of the Institute of Marine and Coastal Sciences at Rutgers University and Dr. Larry Horowitz of Princeton University for teaching me invaluable knowledge in the areas of atmospheric science and oceanography and answering numerous questions with immense patience. My thanks also go to Dr. Prosenjit Bagchi and Dr. Francisco Diez of the Mechanical and Aerospace Engineering department for enlightening me on important concepts of multiphase flow and combustion systems, reviewing my thesis and fruitful discussions about the project.

Lastly, I would like to thank Mr. Imtiaz Rangwala, Mr. Steven Lei and all of my friends for supporting me each and every step along the way.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Earth's atmosphere has a layered structure. The atmospheric layer closest to the surface of the Earth is called troposphere. Globally, the troposphere extends vertically to about 10 km from the Earth's surface. An aerosol is a liquid or solid particle suspended in air. Most of the observed aerosols in the troposphere are about 0.1 μm (microns) in radius. Due to the small sizes of aerosols and high densities of air at low altitudes, the tropospheric aerosols can have a lifetime of several weeks. The sources of tropospheric aerosols can be natural or man-made, and most of the tropospheric aerosols originate from land. Aerosols usually grow in size due to Brownian coagulation, and some aerosols which are hygroscopic, i.e. attract water, can also experience a condensational growth due to absorption of water vapor from the air. Aerosols in the troposphere are usually present at number concentrations of several hundred per cubic centimeter, except for aerosols over highly polluted areas, where their concentrations can exceed ten thousand per cubic centimeter.

A cloud is an assembly of cloud droplets or ice particles with the number concentration of several hundred per cubic centimeter. Cloud droplets or ice particles appearing at such high concentrations become visible as a cloud. Warm clouds are clouds which contain cloud droplets only. They are a major focus in this study. Mixed clouds, which can contain both cloud droplets and ice particles, and ice clouds, which can contain ice particles only, will be mentioned only briefly here. An average size of the cloud droplet is about 10 μm in radius. Cloud droplets form on the surfaces of tropospheric

aerosols by the mechanism of heterogeneous homomolecular nucleation. Aerosols which are suitable for the formation of cloud droplets are called condensation nuclei (CN). The relative humidity required for the cloud droplets to form on the surfaces of CN must only slightly exceed 100%. Cloud droplets cannot form by self-nucleation, i.e. without CN, because relative humidity required for homogeneous homomolecular nucleation makes up several hundred percent, which is usually not observed in the atmosphere. Thus, all cloud droplets contain aerosol cores.

Clouds in the troposphere can appear at different altitudes. Depending on the altitude, the clouds can be divided into four types: (1) high level clouds, 5-10 km; (2) middle level clouds, 2-7 km; (3) low level clouds, 0-2 km; (4) clouds with vertical development, which can occupy the entire troposphere, and are called convective or cumulus clouds. Usually, high level clouds are ice clouds, middle level clouds are mixed clouds, and low level clouds are warm clouds. Convective clouds represent a combination of the first three types. A major focus in this study is on low-level clouds, namely marine stratocumulus clouds. These types of clouds are formed as a result of weak vertical convective currents under the top of the marine boundary layer, i.e. at a height of about 1 km above the surface of the Earth. A dry and warm air above this altitude prevents a vertical cloud development as seen in the cumulus clouds. Hence the vertical dimension of the cloud is relatively small, usually on the order of several hundred meters, but horizontal coverage can exceed several hundred kilometers. Marine stratocumulus clouds occupy about one third of the ocean's surface. Due to such a large coverage of the Earth's surface, they play an important role in the energy budget of the Earth. They reduce the penetration of the incoming sun light on to the surface of the ocean. Without these

clouds, the ocean surface would have absorbed most of the incoming solar radiation because of a low albedo of the ocean. These clouds also play an important role in the chemical processes occurring in the atmosphere near the Earth's surface, scavenging air in the boundary layer from air pollutants.

Cloud droplets can grow by (a) condensation of water vapor and (b) gravitational coalescence of cloud droplets - when there is diversity in cloud droplet sizes, i.e. large cloud droplets fall faster and collect smaller droplets while they fall. When they grow to a size of about 1 mm in radius, they can reach the ground without evaporating, and become rain drops.

Here are some important things about stratocumulus clouds. In stratocumulus clouds, cloud droplets are created due to a vertical motion of the parcel of air. A parcel of air expands when it rises, and the moisture contained in the parcel condenses on the surfaces of the CN to form cloud droplets. Mass balance requires other parcels of the air in the neighborhood to sink and compress, so that cloud droplets in them evaporate. Rising and sinking motions of the air within stratocumulus clouds always occur at the same time, and usually have comparable magnitudes.

There exist several models for aerosol and cloud microphysical processes which resolve sizes of cloud droplets and CN based on environmental conditions. CARMA, which stands for the Community Aerosol and Radiation Model for Atmospheres, is one of them. CARMA was developed by NASA Ames Research Center (Toon *et al.*, 1988; Ackerman *et al.*, 1995), and by default, it calculates the cloud microphysical processes within convective clouds. Major steps which led to the development of the CARMA model are described in chapter 2.

CARMA is a stand-alone model for aerosol and cloud microphysical processes. Microphysics in CARMA is coupled to transport and radiation. This work on improving the CARMA model was motivated by a great need to develop a module for climate models which already can calculate a wind field, temperature and species concentrations but lack a tool for converting water vapor and CN to cloud droplets. The first step in the improvement of CARMA was organizing input and output to each of CARMA's subroutines. Several corrections were made to the vertical transport subroutines which are explained in chapters 3 and 4. Radiation was removed from the CARMA model. Horizontal transport subroutines and ice-handling subroutines which were present in the original CARMA v. 2.2, but were deactivated in the default simulation, were completely removed from CARMA. The microphysical part of CARMA v. 2.2 was isolated from other processes and prepared for being coupled with climate models.

To our knowledge, no coupling between cloud microphysical models and climate models has yet been accomplished. Recently Timmreck *et al.* (2001) reported of a coupling between a climate model called ECHAM4 and an aerosol microphysical model for sulfuric acid aerosols appearing in the stratosphere after strong volcanic eruptions.

The developed version of CARMA is tested by comparing model results on marine stratocumulus clouds over the North Sea with observational data from Nicholls (1984). The model results agree with the observed data.

Organization of the thesis

In Chapter 2, the literature review on microphysical models for clouds and aerosols in the atmosphere is given with the emphasis on the history of development of CARMA. In Chapter 3, the physical model for the process as well as the governing equations is considered. The numerical algorithms are presented in Chapter 4. Modeling and results for the microphysics of stratocumulus clouds are given in Chapter 5. Finally, the conclusions of the research are presented in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

Development of numerical models of clouds started about forty years ago. One of the first models of clouds was a model by Srivastava (1967). In this model, water was present in three forms: a gaseous form - water vapor, and two liquid forms - cloud water and rain water. The model did not allow for ice formation. The phase transition between water vapor and cloud water occurred spontaneously when the water vapor's mass exceeded a specified equilibrium value, or equivalently if relative humidity exceeded 100%. The cloud water did not account for cloud droplet sizes. It was just the total mass of cloud water present in a unit mass of air that the model took into a consideration. However, the latent heat released during the cloud water formation was included in the model. Every time when the relative humidity dropped below 100%, the model allowed cloud water to evaporate and released water vapor. The accumulation of cloud water led to subsequent formation of rain water based on certain threshold criteria. Unlike cloud water, the model allowed the rain water to fall with a certain speed, which led to the formation of rain. The model neglected microphysics. Furthermore, the model assumed a horizontal uniformity of the cloud, so the model was one-dimensional. The model was simple but produced results for the precipitation rate that were close to observations. Further developments of the cloud models accounted for the sizes of cloud droplets and aerosols and all microphysical processes.

CARMA is one of the models which explicitly resolve sizes of cloud droplets and aerosols. The development of the CARMA model went in parallel with the development

of other cloud and aerosol microphysical models. Since we are using the CARMA model in this study, it is useful to understand those basic steps which have led to the development of the CARMA model. Everything started with the work of Hamill *et al.* (1977). In 1961, a layer of aerosols made up of H_2O and H_2SO_4 was found in the stratosphere between about 15 and 30 km above the surface of the Earth (Junge *et al.*, 1961). In order to describe the life cycle of the observed aerosols, Hamill *et al.* (1977) developed a theory and supported it with calculations accounting for each individual microphysical process affecting the stratospheric aerosol layer. The model resolved the sizes of the aerosols using bins and included the relevant and most important microphysical processes such as nucleation, condensational growth/evaporation, Brownian coagulation and sedimentation. In this model, stratospheric chemistry was ignored. The mutual influence of processes was not accounted for in the model. The model considered neither a horizontal transport nor vertical transport of aerosols; all the processes were calculated for a single grid box corresponding to height of 20 km. The lack of the transport of aerosols and gases-precursors within the stratospheric aerosol layer was primarily due to the fact that the stratosphere is mostly quiescent and winds are quite weak at that level in the atmosphere. Most of the important ideas and algorithms concerning the microphysics of aerosols were adopted from Fuchs (1964).

The next step in the development of the CARMA model was by the work in Turco *et al.* (1979). Turco combined all the microphysical processes into one model, allowing for the mutual influence of the processes on each other. The model included chemical aspects such as sources of H_2SO_4. Among the improvements to the model was a special treatment of solid cores released during evaporation of the aerosols and their subsequent recycling

in the stratosphere. Again, this model contained the Brownian coagulation only, among all possible mechanisms for coagulation. In addition, Turco also added a one-dimensional transport of aerosols and gases in the vertical direction throughout the entire troposphere and stratosphere (between 0 and 58 km).

About a decade later, the model, described in Turco *et al.* (1979), was reformulated and generalized by Toon *et al.* (1988). The first and biggest improvement came to the model from including the horizontal transport of aerosols and gases. Chemistry was not included in this model. The mechanisms of nucleation, condensational growth and evaporation were generalized to include as many condensing or evaporating gases onto or from the surfaces of cloud droplets or aerosols as needed. In fact, stratospheric sulfuric acid aerosols form by either homogeneous heteromolecular or heterogeneous heteromolecular nucleation that involves two gases: water vapor and sulfuric acid vapor. Their subsequent growth is performed by condensation of these two vapors onto the surfaces of the aerosols. In the troposphere, cloud droplets are formed from water vapor only and their growth is realized by absorbing the water vapor alone. The generalized algorithm of Toon had the ability to account for both scenarios. Due to these modifications, the model became indifferent to whether the tropospheric cloud droplets or stratospheric aerosols were to be modeled; hence the model became suitable for modeling tropospheric clouds and aerosols as well. This work heavily relied on the information in Pruppacher and Klett (1978).

The next set of contributions to the model was made by Ackerman *et al.* (1995). This model was adjusted to calculate the vertical transport of cloud droplets, aerosols on which cloud droplets form, i.e. CN, water vapor and potential temperature concentration and

cloud microphysical processes in the boundary layer of the Earth. In the new model formulation, the microphysics of clouds and aerosols was coupled to the atmospheric radiation and turbulence closure equations. The new model included the gravitational coalescence in addition to the pre-existing Brownian coagulation mechanism. However, the horizontal variability of clouds was neglected; thus, the model was one-dimensional. The model results were validated by comparing the model output with the measurements from Nicholls (1984) for the structure of the stratocumulus cloud over the North Sea. This model, with some modifications, was laid into the foundation of the CARMA model. The modifications included the removal of the turbulence closure equations, and the re-adjusting of the model to calculate the microphysical structure of cumulus clouds. Once created in 1995, the CARMA model has since undergone a series of modifications towards the model that is currently available to the public today, i.e. CARMA v. 2.2. All versions of the CARMA model were designed to perform as a stand-alone model.

The version of the CARMA model developed in this study is designed to become a part of the existing climate models. The numerical weather/climate prediction models often lack an explicit treatment of size distributions of cloud droplets and aerosols. Instead, models use numerous parameterizations for various cloud types and aerosol types (e.g., Schmidt *et al.*, 2006). A coupling between cloud microphysical models and numerical weather/climate prediction models has only recently become possible with the aid of large supercomputers capable of handling lengthy calculations. For example, Timmreck *et al.* (2001) reported of a coupling between two models: a model for microphysics of stratospheric aerosols and the ECHAM4 climate model. To our knowledge, a full coupling between cloud microphysical models and climate models has

never been accomplished. One of the most difficult cloud types to parameterize in the climate models is marine stratocumulus clouds (Schmidt *et al.*, 2006). The developed version of the CARMA model was modified to calculate the size distributions of cloud droplets and aerosols of a marine stratocumulus cloud. The results of the model calculations were validated by comparing the model results with observations of Nicholls (1984). The validation was performed by a method described in Zhang (2005), who used a slightly earlier version of the CARMA model, i.e. CARMA v. 2.0.

CHAPTER 3

THEORETICAL BACKGROUND AND GOVERNING EQUATIONS

Model equations

The model equations are written in the form that involves the transport of a tracer on the left side of the equations and the source and sink terms on the right side of the equations. The effect of microphysics appears through the right side of the equations. The model deals with two types of tracers: particles, which include cloud droplets and CN, and water vapor. The transport of potential temperature concentration is governed by the same equation. The complete three-dimensional equation for the transport of a tracer in the conservative form can be expressed as follows (Toon *et al.*, 1988):

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(uc) + \frac{\partial}{\partial y}(vc) + \frac{\partial}{\partial z}(wc)$$

$$- \frac{\partial}{\partial x}\left[k_x \rho \frac{\partial (c/\rho)}{\partial x}\right] - \frac{\partial}{\partial y}\left[k_y \rho \frac{\partial (c/\rho)}{\partial y}\right] - \frac{\partial}{\partial z}\left[k_z \rho \frac{\partial (c/\rho)}{\partial z}\right] = \tag{3.1}$$

$$(sources - sinks) \cdot Vm \cdot Hm_1 \cdot Hm_2$$

where c represents the tracer concentration such as the particle concentration, water vapor concentration, and potential temperature concentration.

On the left side of equation (3.1), the second, third and fourth terms are advection terms for the x, y and z directions respectively; the fifth, sixth and seventh terms are diffusion terms for the x, y and z directions, respectively. In the model, z is the vertical coordinate, x and y are horizontal coordinates. The right side represents sources and sinks for the tracer concentrations. Vm, Hm_1 and Hm_2 are metric factors which

depend on the chosen coordinate system. For the Cartesian coordinate system, the values of Vm, Hm_1 and Hm_2 are all equal to unity. For the other coordinate systems, Vm, Hm_1 and Hm_2 can be found in Toon *et al.* (1988).

u, v and w appearing in the advection terms are advection velocities in the x, y and z directions respectively (wind speed components), which are not calculated by the model. Their values must be set up for the simulations.

k_x, k_y and k_z are turbulent diffusion coefficients in the x, y and z directions respectively. The model does not calculate their values from a turbulence model. Their values must be also set up for the simulations.

The three-dimensional transport equation (3.1) can be split into three one-dimensional equations as described in Yanenko (1971).

The advantage of this approach is that the model can be run as one-, two- or three-dimensional model by simply turning off the unneeded equations.

Since we would like to run the model only in one dimension, i.e. the vertical direction, assuming the horizontal uniformity of the cloud, equations for the horizontal transport, that originate from splitting in time equation (3.1), do not need to be solved by the model, and therefore they do not appear here. The tracer transport equation for the vertical direction then can be written as

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}(wc) - \frac{\partial}{\partial z}\left[k_z \rho \frac{\partial (c/\rho)}{\partial z}\right] = (sources - sinks) \cdot Vm \cdot Hm_1 \cdot Hm_2 \qquad (3.2a)$$

This equation can be split in time in the following manner:

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}(wc) - \frac{\partial}{\partial z}\left[k_z \rho \frac{\partial (c/\rho)}{\partial z}\right] = 0 \qquad (3.2b)$$

$$\frac{\partial c}{\partial t} = (sources - sinks) \cdot Vm \cdot Hm_1 \cdot Hm_2 \qquad (3.2c)$$

which allows for separation of the transport from microphysics.

For our purposes, only the Cartesian coordinate system will be of interest, hence Vm, Hm_1 and Hm_2 can be taken out of the above equations. In the equations below, given for each of the individual tracers, the unsplit form, given by equation (3.2a), will be used in place of the split form: equation (3.2b) and equation (3.2c).

The vertical transport equation can, therefore, be written as :

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}(wc) - \frac{\partial}{\partial z}\left[k_z \rho \frac{\partial (c/\rho)}{\partial z}\right] = sources - sinks \qquad (3.3)$$

The model must be complemented with the continuity equation for the air density ρ :

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(u\rho) + \frac{\partial}{\partial y}(v\rho) + \frac{\partial}{\partial z}(w\rho) = 0 \qquad (3.4)$$

When running the model as a two- or three-dimensional model, the air density is found from equation (3.4). However, since only the transport in one dimension is of interest, the air density is not allowed to change in the simulations, i.e. equation (3.4) is not used for updating the air density ρ (it is assumed that vertical convergence is balanced by the horizontal divergence), but the rate of change that would have resulted from solving equation (3.4) must be added into the remaining transport equations. This results in an additional term on the left side of equation (3.3), as will be seen below. For convenience, the subscript z of the vertical turbulent diffusion coefficient will be omitted in the equations below.

Each of the transport equations below can be derived from equation (3.3) by including a minor modification to the equation in the form of the fourth term on the left side of the equation.

Particle transport equation:

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial z}\left[\left(w - v_f\right)C\right] - \frac{\partial}{\partial z}\left[k_p \rho \frac{\partial (C/\rho)}{\partial z}\right] - \frac{C}{\rho}\frac{\partial}{\partial z}(w\rho) =$$

$$S_n - R_n C + \frac{\partial}{\partial r}\left[g_r C\right] \tag{3.5}$$

$$+ \int_{r_{\min}}^{r} K^C\left(r, r^3 - r'^3\right)C(r')C\left(r^3 - r'^3\right)dr - C\int_{r_{\min}}^{r_{\max}} K^C(r, r')C(r')dr'$$

Here the advection velocity in the second term of equation (3.5) differs from w by v_f, which is the terminal fall speed of particles, also called the sedimentation velocity. In equation (3.5), C is the particle number concentration, ρ is the air density, w is the vertical velocity, z is the vertical coordinate, t is the time, $k_p = k_m/\sigma_p$ is the eddy diffusivity for particles, k_m is the turbulent viscosity coefficient, σ_p is the turbulent Prandtl number for particles (ratio of the turbulent viscosity to the eddy diffusivity for particles), $r_{\min}$ and $r_{\max}$ are the smallest and largest radii of particles allowed in the model for each of the two types considered in the model, respectively, r is an independent variable, determines the radius space for the growth of particles, r' is a dummy argument for the radius of a particle, S_n is a source of particles, R_n is a removal rate of particles, g_r is the rate of condensational growth or evaporation, K^C is the kernel of the integro-differential equation for coagulation. In the model, coagulation is allowed only between cloud droplets. Coagulation may occur due to the following processes: Brownian motion,

convective Brownian diffusion enhancement and gravitational coalescence. Equation (3.5) is solved for each particle size bin of CN and for each particle size bin of cloud droplets. Since cloud droplets contain cores, the transport of the cores must also be taken into account. Hence, an equation similar to (3.5) is solved for the transport of cloud droplet cores, where C will be the mass concentration of cloud droplet cores. The equation for the cores is given in Turco *et al.* (1979).

The sedimentation velocity, v_f, is computed based on three flow regimes of the flow around particles, which are determined based on the particle Reynolds number, $\mathrm{Re}_p = v_f 2r\rho/\mu$. First, a highly viscous flow is assumed around particles, so nonlinear inertial terms are neglected in the flow equations. An analytical solution exists for this flow regime (Stokes regime) for the drag exerted on the sphere and is given by the Stokes formula. In the absence of other forces, the drag force will be equal to the gravity force, and from that, an equation for the terminal fall velocity of particles of radius r can be obtained, where a correction for the gas non-continuum effect, β, is introduced to account for a highly rarified air at high altitudes:

$$v_f = \frac{2}{9}\frac{gr^2\rho_p}{\mu}\beta \tag{3.6}$$

where g is the acceleration due to gravity ($g = 9.806\,m/s^2$), μ is the dynamic viscosity of the air, ρ_p is the density of the material of the particle ($1780\,kg/m^3$ for CN made of ammonium bisulfate and $1000\,kg/m^3$ for cloud droplets). The air density is significantly lower than the particle density; thus it does not appear in equation (3.6), β is the slip factor to account for the gas non-continuum effect and it was determined by Millikan (1923):

$$\beta = 1 + 1.246Kn + 0.42Kn\,e^{\frac{-0.87}{Kn}} \tag{3.7}$$

where $Kn = \lambda / r$ is the Knudsen number, $\lambda = 2\mu / \rho V_g$ is the mean free path of air molecules, $V_g = \sqrt{8/\pi R_{air} T}$ is the mean thermal velocity of air molecules, R_{air} is the gas constant of air ($287\ J/kg\ K$), T is the absolute temperature. The last term in equation (3.7) accounts for a smooth transition from the $Kn << 1$ case (hydrodynamic particle motion) to the $Kn >> 1$ case (molecular particle motion). In the former case, $\beta \to 1$ and in the latter case, β reduces to the form given by Cunningham (1910). The Stokes formula for the drag is valid till about $\mathrm{Re}_p = 1$. Thus, for $\mathrm{Re}_p < 1$, v_f computed from equation (3.6) will be used in the model.

If the obtained Re_p is in the range $1 \le \mathrm{Re}_p \le 1000$, then v_f is calculated via first determining the drag coefficient C_d through the curve $C_d = f\left(\mathrm{Re}_p\right)$ for spheres (Figure 10-6 from Pruppacher and Klett, 1978). Re_p and C_d thus computed will be used to calculate: $X = \ln\left(C_d\,\mathrm{Re}_p^2\right)$ and $Y = \ln\left(\mathrm{Re}_p\right)$. The curve-fitting to the data in Figure 10-35 from Pruppacher and Klett (1978) gives: $Y = 0.83X - 0.013X^2$. The obtained Y will be used to calculate Re_p from: $\mathrm{Re}_p = \exp(Y)\beta$. Then the particle terminal fall velocity will be computed from:

$$v_f = \frac{\mu}{2r\rho_p}\mathrm{Re}_p \tag{3.8}$$

If the obtained Re_p is in the range $\mathrm{Re}_p > 1000$, v_f is computed from (Pruppacher and Klett, 1978):

$$v_f = \sqrt{\frac{8}{3}\frac{gr\rho_p}{C_d\rho}}\beta \tag{3.9}$$

Here $C_d = 0.45$ is an asymptote to the curve $C_d = f\left(\mathrm{Re}_p\right)$ for the case of $\mathrm{Re}_p > 1000$.

The "drag crisis" that occurs at $\mathrm{Re}_p > 200{,}000$ (as discussed on p. 341 in Pruppacher and Klett, 1978), where C_d drops dramatically to 0.2, is ignored in the model. Hydrodynamic deformation of liquid cloud droplets as well as any breakup due to Rayleigh-Taylor instability is also not taken into account in the model.

The coagulation kernel, K^C, is a sum of three kernels:

$$K^C = K^B + K^{DE} + K^{GC} \tag{3.10}$$

where K^B is the Brownian coagulation kernel, K^{DE} is the coagulation kernel due to convective Brownian diffusion enhancement, and K^{GC} is the gravitational collection kernel.

K^B is calculated using a formula given by Fuchs (1964). For the Brownian coagulation of particles of sizes r_i and r_j, $K^B_{i,j}$ will be given as

$$K^B_{i,j} = \frac{2\pi\left(r_i + r_j\right)\left(D_{p_i} + D_{p_j}\right)}{\dfrac{r_i + r_j}{r_i + r_j + \left(\delta_i^2 + \delta_j^2\right)^{1/2}} + \dfrac{4\left(D_{p_i} + D_{p_j}\right)}{\left(V_{p_i}^2 + V_{p_j}^2\right)^{1/2}\left(r_i + r_j\right)}} \tag{3.11}$$

where δ_i is a correction factor (Fuchs, 1964):

$$\delta_i = \frac{1}{6r_i l_{b_i}}\left[\left(2r_i + l_{b_i}\right)^3 - \left(4r_i^2 + l_{b_i}^2\right)^{\frac{3}{2}}\right] - 2r_i \tag{3.12}$$

where $l_{b_i} = 8D_{p_i}/\pi V_{p_i}$, $V_{p_i} = \sqrt{8k_B T/\pi m_i}$ is the average kinetic velocity of a particle of radius r_i and mass m_i, k_B is the Boltzmann constant ($k_B = 1.38054 \cdot 10^{-23}\ J/K$), D_{p_i} is the particle diffusion coefficient calculated from the Einstein relation $D_{p_i} = k_B T B_i$, where $B_i = \beta_i / 6\pi r_i l_{b_i}$ is the mobility of particles (ratio of the particle velocity to the force that causes the steady motion).

Convective Brownian diffusion enhancement is related to the mixing of air in the wakes of falling large particles, which was found to be significant for enhancement of Brownian coagulation (Pruppacher and Klett, 1978). A linear expression between K^B and K^{DE} according to Pruppacher and Klett (1978) is

$$K_{i,j}^{DE} = \begin{cases} \left(0.45\,\mathrm{Re}_{p_j}^{1/3}\, Sc_i^{1/3}\right) K_{i,j}^B & if\ \mathrm{Re}_{p_j} < 1 \\ \left(0.45\,\mathrm{Re}_{p_j}^{1/2}\, Sc_i^{1/3}\right) K_{i,j}^B & if\ \mathrm{Re}_{p_j} \geq 1 \end{cases} \qquad for\ r_j \geq r_i \tag{3.13}$$

where Sc_i is the particle Schmidt number given by:

$$Sc_i = \mu / \rho D_{p_i} \tag{3.14}$$

The gravitational collection kernel K^{GC} is expressed as (Pruppacher and Klett, 1978)

$$K_{i,j}^{GC} = E_{coll_{i,j}}\, E_{coal_{i,j}}\, \pi \left(r_i + r_j\right)^2 \left| v_{f_i} - v_{f_j} \right| \tag{3.15}$$

where $E_{coll_{i,j}}$ is the collision efficiency determining the probability of a collision of particles of radii r_i and r_j, interpolated from Hall's values (Hall, 1980), $E_{coal_{i,j}}$ is the coalescence efficiency equal to the ratio of the number of coalescences, i.e. sticking of particles together, to the number of collisions of particles of radii r_i and r_j, taken from

Beard and Ochs (1984). g_r is the condensational growth/evaporation rate. g_r is written in the form proposed by Barkstrom (1978) but without the radiation term:

$$g_r \equiv \frac{dr}{dt} = \frac{g_0 n_{vap}\left(S + 1 - A_k A_s\right)}{1 + g_0 g_1 n_{vap}} \tag{3.16}$$

where g_0 and g_1 are given by the following expressions:

$$g_0 = \frac{D_v F_v M_w}{A_v \rho_w r} \qquad\qquad g_1 = \frac{L_v^2 M_w \rho_w r}{RT^2 K_t F_t} \tag{3.17}$$

in which D_v and K_t are diffusivity of water vapor in air and thermal conductivity of dry air, respectively, M_w is the molecular weight of water ($M_w = 18\,g/mol$), ρ_w is the density of liquid water ($\rho_w = 1000\,kg/m^3$), Av is Avogadro's number ($Av = 6.02252 \cdot 10^{23}\,mol^{-1}$), R is the universal gas constant ($R = 8.3143\,J/mol \cdot K$), L_v is the latent heat of water evaporation, F_t and F_v are ventilation corrections accounting for the effects of cloud droplet sedimentation on the heat flux and water vapor molecules flux from or to the cloud droplets, respectively, calculated via functions recommended by Pruppacher and Klett (1978):

$$F_v = \begin{cases} 1 + 0.108\left(Sc^{1/3}\,\mathrm{Re}_p^{1/2}\right)^2 & 0 \le Sc^{1/3}\,\mathrm{Re}_p^{1/2} \le 1.4 \\ 0.78 + 0.308\left(Sc^{1/3}\,\mathrm{Re}_p^{1/2}\right) & Sc^{1/3}\,\mathrm{Re}_p^{1/2} > 1.4 \end{cases} \tag{3.18a}$$

$$F_t = \begin{cases} 1 + 0.108\left(\mathrm{Pr}^{1/3}\,\mathrm{Re}_p^{1/2}\right)^2 & 0 \le \mathrm{Pr}^{1/3}\,\mathrm{Re}_p^{1/2} \le 1.4 \\ 0.78 + 0.308\left(\mathrm{Pr}^{1/3}\,\mathrm{Re}_p^{1/2}\right) & \mathrm{Pr}^{1/3}\,\mathrm{Re}_p^{1/2} > 1.4 \end{cases} \tag{3.18b}$$

where $\mathrm{Pr} = \mu C_p / K_t$ is the Prandtl number for the air, C_p is the specific heat at constant pressure for the air ($1004\,J/kg\,K$). n_{vap} is saturation vapor pressure, expressed as a number density of water vapor molecules, $S = \left(n_\infty / n_{vap}\right) - 1$ is the supersaturation of water

vapor with respect to liquid water, where n_∞ is the ambient vapor pressure: $n_\infty = GA_v / M_w$. The gas kinetic corrections to the diffusion coefficient of water vapor, D_v, and the thermal conductivity, K_t, are described by Toon *et al.* (1989b). In these expressions the condensation coefficient (called the mass accommodation coefficient by *Toon et al.*, 1979) and the thermal accommodation coefficients are taken to be unity.

The factors A_k and A_s in (3.16) account for, respectively, the increase in vapor pressure exerted by the cloud droplet due to curvature (the Kelvin effect) and its reduction due to the solute effect. They are given by

$$A_k = \exp\left(\frac{2M_w\sigma_{w/a}}{r\rho_w RT}\right) \qquad A_s = \exp\left(-\frac{\nu_d \Phi_s m_s M_w}{m_w M_s}\right) \tag{3.19}$$

The remaining factors in A_k and A_s are determined by the composition of the cloud droplet core; thus, ν_d is the dissociativity of the dissolved material ($\nu_d = 2$ for ammonium bisulfate, sulfuric acid and most other common materials for CN), Φ_s is the practical osmotic coefficient whose value can be taken as 1 for dilute solutions, m_s is the mass of the core, $\sigma_{w/a}$ is the surface tension of water-air interface, M_s is the molecular weight of the material of the core ($M_s = 115\,g/mol$ for ammonium bisulfate, $M_s = 98\,g/mol$ for sulfuric acid). In equation (3.16) the radiation term is omitted for the purpose of decoupling microphysics from radiation. Based on sensitivity tests of Zhang (2005), including the radiative term into equation (3.16) gives only an insignificant correction to the overall results of the cloud simulations.

The meaning of the terms in the particle transport equation is following. The second term on the left side of equation (3.5) represents the divergence of the vertical flux due to advection and sedimentation. The third term accounts for the divergence of the vertical flux due to turbulent diffusion, and it slightly differs from a conventional diffusion term by the density of the air, ρ. The reason for the difference is following. The rate of vertical diffusion of particles and gases by atmospheric mixing is usually given by a phenomenological relation (Turco *et al.*, 1979):

$$F_{diff} = -k_z \rho \frac{\partial}{\partial z}\left(\frac{C}{\rho}\right) \tag{3.20}$$

where F_{diff} is the diffusive flux, k_z is the eddy diffusivity, C is the concentration of the diffusing species. The divergence of F_{diff} gives the third term in equation (3.5).

The last term on the left side of equation (3.5) represents the horizontal divergence that compensates for any change in air density due to vertical convergence (Ackerman at al, 1995). In 1-d models, it is assumed that any vertical divergence is balanced by horizontal convergence, hence air density is not allowed to change in time, but a calculated rate of change that would have resulted from advection is added into all transport equations through this corrective term as was explained earlier. The term may be rewritten into the form $\frac{C}{\rho}\frac{\partial \rho}{\partial t}$ via using the conventional continuity equation for the air density: $\partial\rho/\partial t + \partial(w\rho)/\partial z = 0$.

On the right side of (3.5), the first term is a source term that accounts for the creation of entirely new CN. The second term accounts for conversions between CN and cloud droplets. The third term on the right side of (3.5) represents the flux divergence in radius

space due to condensational growth and evaporation of water. The first integral represents the creation of particles of radius r due to collisions between smaller particles, and the second integral represents the loss of particles of radius r due to collisions with other particles. For ease of programming, the two integrals include radii of particles rather than their volumes, and that differs from Fuchs's original form (Fuchs, 1964). A discrete form of the two integrals is following:

$$\int_{r_{\min}}^{r} K^C\left(r, r^3 - r'^3\right) C(r') C\left(r^3 - r'^3\right) dr \quad = \quad \sum_{j=1}^{i} K_{k,j}^C \, C_k C_j \tag{3.21a}$$

$$C \int_{r_{\min}}^{r_{\max}} K^C(r, r') C(r') dr' \quad = \quad C_i \sum_{j=0}^{\infty} K_{i,j}^C \, C_j \tag{3.21b}$$

Transport of water vapor:

$$\frac{\partial G}{\partial t} \quad + \quad \frac{\partial}{\partial z}[wG] \quad - \quad \frac{\partial}{\partial z}\left[k_G \, \rho \frac{\partial (G/\rho)}{\partial z}\right] \quad - \quad \frac{G}{\rho}\frac{\partial}{\partial z}(w\rho) \quad = \tag{3.22}$$

$$-\frac{4\pi}{3}\rho_w \int_{r_{\min}}^{r_{\max}} r'^3 \frac{d}{dr'}[g_r(r') C(r')] dr'$$

where G is the water vapor mass concentration, $k_G = k_m / \sigma_G$ is the eddy diffusivity for water vapor, σ_G is the turbulent Prandtl number for water vapor. The right side of equation (3.22) represents the water vapor exchange with cloud droplets (water in CN is ignored in the model, so no exchange occurs between CN and water vapor).

Energy equation:

$$\frac{\partial(\rho\theta)}{\partial t} \quad + \quad \frac{\partial}{\partial z}(w\rho\theta) \quad - \quad \frac{\partial}{\partial z}\left(k_h \, \rho \frac{\partial \theta}{\partial z}\right) \quad - \quad \theta \frac{\partial}{\partial z}(w\rho) = \tag{3.23}$$

$$\frac{\theta}{C_p T_0}\left[-L_v\left(\frac{\partial G}{\partial t}\right)_{phase}\right]$$

where θ is the potential temperature, determined via $\theta = \left(p_{ref} / p\right) T^{R_{air}/C_p}$, $k_h = k_m / \sigma_h$ is the eddy diffusivity for heat, σ_h is the turbulent Prandtl number for heat, p_{ref} is the standard reference pressure ($10^5\ Pa$), T_0 is the triple-point temperature ($273.16 K$). The effect of microphysics appears through the right side of equation (3.23), i.e. the phase change of G. Along with the phase change term, the right side of equation (3.23) must include a radiation term as well. However, Zhang (2005) showed that the inclusion of the radiation term gives only an insignificant correction to the overall results of the cloud simulations for the marine stratocumulus clouds of interest. Hence, the radiation term is omitted here. It also allows for decoupling radiation from microphysics. Once the potential temperature concentration, $(\rho\theta)$, is found from equation (3.23), the temperature can be updated via $T = \frac{(\rho\theta)}{\rho}\left(p / p_{ref}\right)^{R_{air}/C_p}$.

Temperature-dependent properties

Temperature-dependent properties below are calculated only once before the time loop based on an initial vertical temperature profile.

Dynamic viscosity is estimated from Sutherland's equation (Smithsonian Meteorological Tables, 1951):

$$\mu = \mu_{ref} \frac{T_{ref} + C_{suth}}{T + C_{suth}} \left(\frac{T}{T_{ref}} \right)^{3/2} \tag{3.24}$$

where C_{suth} is Sutherland's constant for the gas. For the air $C_{suth} = 120K$. $\mu_{ref} = 1.8325 \times 10^{-4}\ g/cm \cdot s$ is the reference viscosity at reference temperature $T_{ref} = 296.16K$. μ is in $g/cm \cdot s$.

Thermal conductivity of dry air is estimated from an expression given in Pruppacher and Klett (1978):

$$K_t = 418\left[5.69 + 0.017\left(T - T_0\right)\right] \tag{3.25}$$

where K_t is in $erg/cm \cdot s \cdot K$.

Similarly, the diffusivity of water vapor in air from Pruppacher and Klett (1978) is

$$D_v = 0.211\left(\frac{T}{T_0}\right)^{1.94}\left(\frac{p_{ref}}{p}\right) \tag{3.26}$$

with D_v in $cm^2\, s^{-1}$.

And surface tension of water-air interface from Pruppacher and Klett (1978) is

$$\sigma_{w/a} = 76.1 - 0.155\left(T - T_0\right) \tag{3.27}$$

with $\sigma_{w/a}$ in dyn/cm or erg/cm^2.

Latent heat of evaporation is estimated from the expression given in Stull (1988):

$$L_v = \left[2.5 - 0.00239\left(T - T_0\right)\right] \cdot 10^{10} \tag{3.29}$$

with L_v in cm^2/s^2

Air density is estimated from the equation of state:

$$\rho = p \Big/ R_{air} T \tag{3.30}$$

with ρ in g/cm^3.

Static pressure of dry air is calculated from the barometric law:

$$p = p_{ref}\, e^{-\frac{z}{H}} \tag{3.31}$$

where $H = R_{air}T/g$ is the scale height.

Boundary conditions for particle transport equation

The flux of particles across the lower boundary of the model is set to be zero for the particle transport, and for sedimentation the flux is given by

$$F_{sed_{1/2}} = -C(z_1)v_{f_{1/2}} \tag{3.32}$$

where $F_{sed_{1/2}}$ is the sedimentation flux across the lower boundary of the model, $C(z_1)$ is the particle concentration at the center of the bottom vertical layer, $v_{f_{1/2}}$ is the sedimentation velocity at the bottom of the domain.

The flux of particles across the upper boundary of the model is set to be zero for both particle transport and sedimentation.

Boundary conditions for water vapor transport equation

The flux of the water vapor across both the lower boundary and the upper boundary of the domain is taken as zero in the model.

Boundary conditions for energy equation

Potential temperature concentration, $\rho\theta$, is kept constant at the centers of the bottom and top vertical layers of the domain at the values calculated from the initial temperature profile. Hence, temperature also remains constant at the centers of the bottom and top vertical layers.

Changes made to the original CARMA v. 2.2

The improvement of the CARMA model is related to introducing several changes to the original CARMA v. 2.2. These changes include: removing subroutines related to radiation, horizontal transport and formation of ice, changing input/output to each of the subroutines via introducing lists of arguments, introducing a subroutine carma_box (a microphysical driver), from which all microphysical subroutines are called, separating vertical advection, vertical diffusion and sedimentation, changing the algorithm for solving the vertical advection equation and sedimentation equation, correcting a treatment of boundary conditions for the diffusion equation and introducing subroutines for outputting results in the GrADS (Grid Analysis and Display System is software for visualizing geophysical data) format. A brief explanation of each of the changes is provided below.

The original CARMA v. 2.2 has three components: transport, microphysics and radiation. These components are strongly coupled. Effect of radiation on microphysics appears in two places: equation (3.16) for the rate of condensational growth of cloud droplets and equation (3.23) - energy equation. Sensitivity tests of Zhang (2005) signify that the radiation terms are insignificant for the overall results of the cloud simulations. Hence microphysics can be decoupled from radiation, and radiation subroutines can be removed. We choose to keep variables, corresponding to radiation, in the lists of arguments of subroutines solving equations (3.16) and (3.23). This will allow for coupling microphysics with radiation modules of climate models later.

In the original CARMA v. 2.2 horizontal transport subroutines are present, but not called. Since our final goal is to create a microphysical module for climate models, the

vertical transport subroutines are sufficient to create a background where the microphysical changes of cloud droplets and atmospheric aerosols can be tracked. A convenient way to deal with clouds is to neglect a horizontal variability of cloud properties (plane-parallel assumption) and consider only transport in the vertical direction. Thus, horizontal transport subroutines can be removed from our model.

The original CARMA v. 2.2 contains subroutines which are related to a freezing of cloud droplets, formation of ice particles on the surfaces of freezing nuclei, growth and melting of ice particles, etc. These subroutines are not called in the default simulation. One of major uncertainties in cloud parameterizations within climate models is related to marine stratocumulus clouds (Schmidt *et al.*, 2006), which usually do not contain ice. Our main objective is to provide a microphysical module for climate models which will allow for an explicit calculation of the microphysics of these types of clouds. Hence, ice-related subroutines can be removed from CARMA.

In the original CARMA v. 2.2, subroutines interchange data through global variables, which hides information about input parameters for each of the subroutines and outputs. It is the most difficult part of this work, taking into a consideration that the number of input and output parameters for most subroutines of CARMA exceeds 100, and the number of subroutines in the improved version of CARMA is close to 80 (~140 in the original CARMA v. 2.2). A schematic of the original CARMA v. 2.2 is provided in Appendix II, and a schematic of the improved version of CARMA is given in Appendix III. A list of all subroutines of the improved version of CARMA along with their functions is given in Appendix IV.

In the original CARMA v. 2.2, coagulation subroutines were called once per time step, in which loops over spatial grid points were present, whereas subroutines related to nucleation, condensational growth and evaporation were called once per spatial grid point. A microphysical module, carma_box, which is intended to be developed in this study, must be called once per spatial grid point. Introducing a subroutine called carma_box to drive all microphysics related processes is necessary, because it will allow for extracting microphysics from CARMA and couple it with climate models later.

In the original CARMA v. 2.2, the diffusion-advection equation was solved in vertical transport subroutines. Sedimentation was accounted for by introducing a sedimentation velocity of particles into the advection speed. The diffusion-advection equation in CARMA v. 2.2 is solved using an explicit scheme if a certain stability criterion is satisfied for the time step size; otherwise it is solved implicitly. Separating advection from diffusion gives benefits of choosing an algorithm for time integration of the advection equation separately from an algorithm for the diffusion equation. Separating sedimentation from vertical advection is a necessity, because sedimentation is a microphysical process; hence it must be calculated inside the microphysical module carma_box. In the improved version of CARMA, the vertical diffusion equation is solved implicitly, but the vertical advection equation and the sedimentation equation are solved explicitly. Numerical algorithms of solution of the advection equation, diffusion equation and sedimentation equation are given in Chapter 4.

We found that in subroutines for solving the vertical advection equation, $\partial C/\partial t + \partial/\partial z\left[wC\right] = 0$, in the original CARMA v. 2.2 by the PPM method, boundary conditions were mistreated. Instead of rewriting subroutines that would use the PPM

method from scratch, we replace them with subroutines which use a simpler method called REA, which is complemented with the minmod slope limiter. The tests of the algorithm are provided in Chapter 4.

In the improved version of CARMA, for solving the sedimentation equation we choose to use the upwind method in place of the REA with the minmod slope limiter. The upwind method requires a concentration value only from one vertical layer located above; hence it is good for using it with sedimentation fluxes calculated inside the carma_box.

In the original CARMA v. 2.2, in subroutines for solving the diffusion equation, $\partial C/\partial t + \partial/\partial z\left[k_p \rho \partial(C/\rho)/\partial z\right] = 0$, we found some errors with imposing boundary conditions. The subroutines did not allow for large non-zero vertical diffusion coefficients on the domain boundaries. The determined bugs were fixed in vertical diffusion subroutines.

We also created subroutines for writing the model results in the GrADS format, which is popular software for visualizations of geophysical data.

Several other changes were made to the CARMA model. Some subroutine names were changed from the original names. The original makefile for compiling CARMA v. 2.2 was replaced with a makefile of a simpler structure.

CHAPTER 4

NUMERICAL ALGORITHMS

The complete one-dimensional transport equation (3.3) can be split in time for obtaining a numerical solution to the equation via solving one by one equations for advection, diffusion and microphysical processes on one time step. Below are descriptions of the numerical techniques used for the discretization of the equations.

Algorithm for solving the advection equation

The piecewise parabolic method (PPM) of Colella and Woodward (1984) for solution of the advection equation implemented in CARMA v. 2.2, which is the third-order accurate method in space, was replaced by the second-order accurate method called REA (Reconstruct-Evolve-Average), complemented with the minmod slope limiter (LeVeque, 2002). Below is the step-by-step arrival at the final discrete analogue expressions for the advection equation obtained using this second-order accurate method in space, which follows the following scheme: Upwind=>REA=>REA with the minmod slope limiter. The discrete analogue of the sedimentation equation, which will be arrived at along the way, will use the first-order accurate in space upwind method for the advantages explained below.

The advection equation

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}(wc) = 0 \tag{4.1}$$

can be conveniently rewritten into the form:

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}\left(F_{adv}\right) = 0 \tag{4.2a}$$

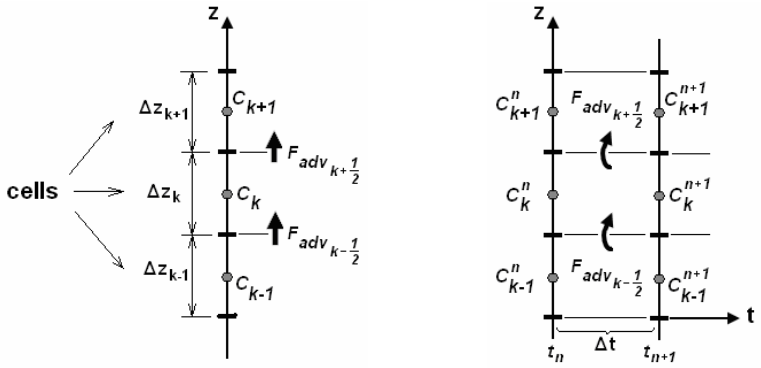
$$F_{adv} = wc \tag{4.2b}$$

where F_{adv} is the advection flux.

Equations (4.2a) and (4.2b) can be discretized in space via the finite-volume method. The physical domain of interest is divided into grid cells called finite volumes as shown in Figure 4-1(a). The temporal discretization of equation (4.2a) is performed via using the first-order Euler forward finite-difference scheme (explicit scheme). Combined together, the discrete analog of equation (4.2a) can be expressed as follows (LeVeque, 2002):

$$c_k^{n+1} = c_k^n - \frac{\Delta t}{\Delta z_k}\left(F^n_{adv_{k+\frac{1}{2}}} - F^n_{adv_{k-\frac{1}{2}}} \right) \tag{4.3}$$

Figure 4-1(b) demonstrates the discretization of the time-space domain, where t_n and t_{n+1} are two consecutive time layers, Δt is the time step size, c_{k-1}^n, c_k^n and c_{k+1}^n are values of c from the previous time layer n in the centers of cells $k-1$, k and $k+1$ respectively, subscript $n+1$ refers to the values of c at the new time layer $n+1$.



(a) (b)

Figure 4-1 (a) One-dimensional finite-volume grid with the specified grid cell sizes and inter-cell advection fluxes to cell k; (b) Discretization of the time-space domain.

The discretization of the advection flux, F_{adv} , varies from one method to another . For the upwind method, which is the first-order accurate method in space, the advection flux on the boundaries of cell k can be expressed as follows:

$$F^n_{adv_{k-\frac{1}{2}}} = \begin{cases} w^n_{k-\frac{1}{2}} c^n_{k-1}, & \text{if } w^n_{k-\frac{1}{2}} > 0 \\ w^n_{k-\frac{1}{2}} c^n_{k}, & \text{if } w^n_{k-\frac{1}{2}} < 0 \\ 0, & \text{if } w^n_{k-\frac{1}{2}} = 0 \end{cases} \tag{4.4a}$$

$$F^n_{adv_{k+\frac{1}{2}}} = \begin{cases} w^n_{k+\frac{1}{2}} c^n_{k}, & \text{if } w^n_{k+\frac{1}{2}} > 0 \\ w^n_{k+\frac{1}{2}} c^n_{k+1}, & \text{if } w^n_{k+\frac{1}{2}} < 0 \\ 0, & \text{if } w^n_{k+\frac{1}{2}} = 0 \end{cases} \tag{4.4b}$$

where $F^n_{adv_{k-\frac{1}{2}}}$ and $F^n_{adv_{k+\frac{1}{2}}}$ are advection fluxes to cell k from the neighboring cells $k-1$ and $k+1$ respectively. $w_{k-\frac{1}{2}}$ and $w_{k+\frac{1}{2}}$ are vertical velocities at the lower and upper boundaries of cell k. As can be seen from equations (4.4a) and (4.4b), c and w are determined at different locations, i.e. a staggered grid is used for c and w. While c is determined at the vertical layer centers, w is determined at the layer boundaries.

The advantage of the upwind method is simplicity. Information only from one additional grid cell located upstream is needed to find the tracer concentration on a new time layer in a given grid box. Despite a tremendous amount of numerical diffusion introduced into the solution by the upwind method when the Courant number ($w\Delta t/\Delta z$) is less than one, it is perfect for solving the sedimentation equation for the particle concentration C :

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial z}\left(-v_f\, C\right) = 0 \tag{4.5}$$

which, similarly to the advection equation above, can be conveniently replaced by two equations:

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial z}\left(F_{sed}\right) = 0 \tag{4.6a}$$

$$F_{sed} = -v_f\, C \tag{4.6b}$$

where F_{sed} is the sedimentation flux.

Similarly to the advection fluxes, the sedimentation fluxes on the boundaries of cell k can be expressed as

$$F^n_{sed_{k-\frac{1}{2}}} = -v_{f_{k-\frac{1}{2}}} C^n_k \tag{4.7a}$$

$$F^n_{sed_{k+\frac{1}{2}}} = -v_{f_{k+\frac{1}{2}}} C^n_{k+1} \tag{4.7b}$$

Once the sedimentation fluxes on the previous time step are calculated from equations (4.7a) and (4.7b) based on the concentration of particles C in two vertical layers: a given vertical layer and one vertical layer above, the particle concentration can be updated via:

$$C^{n+1}_k = C^n_k - \frac{\Delta t}{\Delta z_k}\left(F^n_{sed_{k+\frac{1}{2}}} - F^n_{sed_{k-\frac{1}{2}}}\right) \tag{4.8}$$

The calculation of the sedimentation fluxes via using equations (4.7a) and (4.7b) can be placed into the microphysical part of the model, since sedimentation is a microphysical process.

In place of the first-order upwind method, the solution to the advection equation (4.2a) and (4.2b) can be obtained via assuming that the tracer concentration c is not constant in the vertical layer, but instead changes linearly with height. The slope of the

linear variation of c with height can be introduced to produce the second-order method in space. For constructing a slope, two values of c are necessary. The first value can be the value of c at the center of the given vertical layer, and the second value of c can be the value of c at the center of either the layer below or the layer above the given vertical layer. The two slopes corresponding to the two choices can be written as follows:

$$\sigma_k^n = \frac{c_k^n - c_{k-1}^n}{\frac{1}{2}\left(\Delta z_{k-1} + \Delta z_k\right)} \tag{4.9a}$$

$$\sigma_k^n = \frac{c_{k+1}^n - c_k^n}{\frac{1}{2}\left(\Delta z_k + \Delta z_{k+1}\right)} \tag{4.9b}$$

Slope (4.9a) is used when $w > 0$, slope (4.9b) is used when $w < 0$, and when $w = 0$ the choice between (4.9a) and (4.9b) is unimportant. Taking into account the slopes, the advection fluxes for cell k can be written as follows (LeVeque, 2002):

$$F_{adv_{k-\frac{1}{2}}}^n = \begin{cases} w_{k-\frac{1}{2}}^n c_{k-1}^n + \frac{1}{2} w_{k-\frac{1}{2}}^n \left(\Delta z_k - w_{k-\frac{1}{2}}^n \Delta t\right)\sigma_{k-1}^n, & \text{if } w_{k-\frac{1}{2}}^n > 0 \\ w_{k-\frac{1}{2}}^n c_k^n - \frac{1}{2} w_{k-\frac{1}{2}}^n \left(\Delta z_k + w_{k-\frac{1}{2}}^n \Delta t\right)\sigma_k^n, & \text{if } w_{k-\frac{1}{2}}^n < 0 \\ 0, & \text{if } w_{k-\frac{1}{2}}^n = 0 \end{cases} \tag{4.10a}$$

$$F_{adv_{k+\frac{1}{2}}}^n = \begin{cases} w_{k+\frac{1}{2}}^n c_k^n + \frac{1}{2} w_{k+\frac{1}{2}}^n \left(\Delta z_k - w_{k+\frac{1}{2}}^n \Delta t\right)\sigma_k^n, & \text{if } w_{k+\frac{1}{2}}^n > 0 \\ w_{k+\frac{1}{2}}^n c_{k+1}^n - \frac{1}{2} w_{k+\frac{1}{2}}^n \left(\Delta z_k + w_{k+\frac{1}{2}}^n \Delta t\right)\sigma_{k+1}^n, & \text{if } w_{k+\frac{1}{2}}^n < 0 \\ 0, & \text{if } w_{k+\frac{1}{2}}^n = 0 \end{cases} \tag{4.10b}$$

where σ_{k-1}^n, σ_k^n and σ_{k+1}^n are slopes for cells $k-1$, k and $k+1$ respectively.

The advection fluxes determined via equations (4.10a) and (4.10b) make up the REA method (LeVeque, 2002).

In cases when one of the slopes (4.9) becomes exceedingly large, in order to produce physically realistic results, the slope in cell k can be chosen from the two options, (4.9a) or (4.9b), through the use of the minmod function (LeVeque, 2002):

$$\sigma_k^n = \text{minmod}\left(\frac{c_k^n - c_{k-1}^n}{\frac{1}{2}\left(\Delta z_{k-1} + \Delta z_k\right)} \quad , \quad \frac{c_{k+1}^n - c_k^n}{\frac{1}{2}\left(\Delta z_k + \Delta z_{k+1}\right)} \right) \tag{4.11}$$

where the minmod function is given by (LeVeque, 2002)

$$\text{minmod}\left(a,b\right) = \begin{cases} a, & \text{if} \quad |a| < |b| \quad \text{and} \quad ab > 0 \\ b, & \text{if} \quad |b| < |a| \quad \text{and} \quad ab > 0 \\ 0, & \text{if} \quad ab \le 0 \end{cases} \tag{4.12}$$

The minmod slope limiter chooses a slope between the two options, (4.9a) and (4.9b), based on which one of the two slopes is smaller. If the two slopes are of an opposite sign, then the slope in cell k is taken to be zero. In the latter case, a local maximum or minimum of function c is present in the cell. When this occurs, the method reduces to the first order accurate in space upwind method in that particular cell. When the slope is defined through equations (4.11) and (4.12), one calls the method as the REA method with the minmod slope limiter.

Since the advection equation is solved explicitly, the time step size Δt must obey the CFL condition for the numerical solution to be stable. The CFL condition requires that the cell Courant number be less than or equal to unity, which leads to

$$\Delta t \le \frac{\Delta z_k}{\max\left(\left|w^n_{k-\frac{1}{2}}\right|, \left|w^n_{k+\frac{1}{2}}\right|\right)} \tag{4.13}$$

If the chosen time step size does not obey this condition, the time step size will be divided into sub-steps, so that condition (4.13) holds. After the solution of the advection equation (as well as the sedimentation equation) in a loop over sub-steps is obtained, the normal time step size will be restored for using it in the model. To avoid divisions by zero, $|w|$ in equation (4.13) is limited from below by $10^{-10}\, cm/\sec$. The discrete analogue of the advection equation given by equation (4.3) leads to a system of linear algebraic equations formed on every time step. Because the coefficient matrix of the system is diagonal, no matrix solvers are required for obtaining the solution.

Algorithm for solving the diffusion equation

The diffusion equation

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}\left[-K_z \rho \frac{\partial (c/\rho)}{\partial z}\right] = 0 \tag{4.14}$$

can be conveniently rewritten into the form:

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial z}\left(F_{dif}\right) = 0 \tag{4.15a}$$

$$F_{dif} = -K_z \rho \frac{\partial (c/\rho)}{\partial z} \tag{4.15b}$$

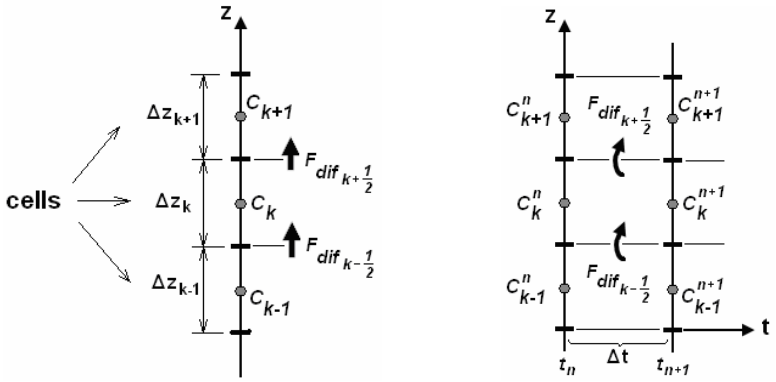
where F_{dif} is the diffusion flux.

Equations (4.15a) and (4.15b) can be discretized in space via the finite-volume method. The physical domain of interest is divided into grid cells called finite volumes as

shown in Figure 4-2(a). The temporal discretization of equation (4.15a) is performed via using the first-order Euler backward finite-difference scheme (implicit scheme). Combined together, the discrete analog of equation (4.15a) can be expressed as follows (LeVeque, 2002):

$$c_k^{n+1} + \frac{\Delta t}{\Delta z_k}\left(F_{dif_{k+\frac{1}{2}}}^{n+1} - F_{dif_{k-\frac{1}{2}}}^{n+1} \right) = c_k^n \tag{4.16}$$

Figure 4-2(b) demonstrates the discretization of the time-space domain, where t_n and t_{n+1} are two consecutive time layers, Δt is the time step size, c_{k-1}^n, c_k^n and c_{k+1}^n are values of c from the previous time layer n in the centers of cells $k-1$, k and $k+1$ respectively, subscript $n+1$ refers to the values of c at the new time layer $n+1$.



(a) (b)

Figure 4-2 (a) One-dimensional finite-volume grid with the specified grid cell sizes and inter-cell diffusion fluxes to cell k; (b) Discretization of the time-space domain.

The discretization of the diffusion flux, F_{dif}, varies from one method to another. For the central difference scheme, which is the second-order accurate method in space, the diffusion flux on the boundaries of cell k can be expressed as follows:

$$F^n_{dif_{k-\frac{1}{2}}} = K^n_{z_{k-\frac{1}{2}}} \rho_{k-\frac{1}{2}} \frac{\frac{c^{n+1}_k}{\rho_k} - \frac{c^{n+1}_{k-1}}{\rho_{k-1}}}{\Delta z_{k-\frac{1}{2}}} \tag{4.17a}$$

$$F^n_{dif_{k+\frac{1}{2}}} = K^n_{z_{k+\frac{1}{2}}} \rho_{k+\frac{1}{2}} \frac{\frac{c^{n+1}_{k+1}}{\rho_{k+1}} - \frac{c^{n+1}_k}{\rho_k}}{\Delta z_{k+\frac{1}{2}}} \tag{4.17b}$$

where $F^n_{dif_{k-\frac{1}{2}}}$ and $F^n_{dif_{k+\frac{1}{2}}}$ are diffusion fluxes to cell k from the neighboring cells $k-1$ and $k+1$ respectively, $K_{z_{k-\frac{1}{2}}}$ and $K_{z_{k+\frac{1}{2}}}$ are vertical turbulent diffusion coefficients at the lower and upper boundaries of cell k, $\rho_{k-\frac{1}{2}}$ and $\rho_{k+\frac{1}{2}}$ are the air density values at the lower and upper boundaries of cell k, $\Delta z_{k-\frac{1}{2}} = 0.5\left(\Delta z_{k-1} + \Delta z_k\right)$, $\Delta z_{k+\frac{1}{2}} = 0.5\left(\Delta z_k + \Delta z_{k+1}\right)$. As can be seen from equations (4.17a) and (4.17b), c and K_z are determined at different locations, i.e. a staggered grid is used for c and K_z. While c is determined at the vertical layer centers, K_z is determined at the layer boundaries.

Since the diffusion equation is solved implicitly, the time step size Δt is not restricted by any stability condition, but may be restricted by the accuracy considerations. Unlike the advection equation, the discrete analogue of the diffusion equation given by equation (4.16) leads to a system of linear algebraic equations formed on every time step. Because the coefficient matrix of the system is tri-diagonal, fast matrix solvers can be used for solving the system such as the matrix solvers based on the LU decomposition method.

Algorithm for solving the coagulation equation

Numerical algorithm for solving the coagulation equation is given in Toon *et al.* (1988). The integro-differential equation for coagulation, deduced from equation (3.5), can be expressed as

$$\frac{\partial C}{\partial t} = \int_{r_{\min}}^{r} K^C\left(r, r^3 - r'^3\right) C(r') C\left(r^3 - r'^3\right) dr - C \int_{r_{\min}}^{r_{\max}} K^C(r, r') C(r') dr' \qquad (4.18)$$

Here we present only a brief description of the algorithm for solving the coagulation equation. All the details can be found in Toon *et al.* (1988). The coagulation integrals in equation (4.18) can be represented as sums over various particle radius bins. Toon *et al.* (1988) derive expressions for particle production and loss rates based on simple considerations. For example, when a particle of size i collides with a particle of a larger size j, the resulting particle will be greater than size i. Hence, a particle is lost from size bin i. The rate of disappearance is $K^C_{i,j} C_i C_j$. An intermediate particle, that results from a collision, will lie between size bins j and $j+1$. The algorithm conserves mass in such a way that only a fraction of the particle mass will transfer to bin $j+1$, and the remaining part of the particle mass stays in size bin j. During a collision, the mass of cloud droplet cores is transferred to various size bins for cloud droplets at the same rate as the cloud droplets number. The boundary conditions in coagulation are based on conservation of mass. Rather than allowing particles to coagulate out of the largest size bin, $r_{\max}$, and therefore to be lost, particles are allowed to collect in the largest size bin. In CARMA coagulation is allowed only between cloud droplets.

Algorithm for solving the condensational growth/evaporation equation

The condensational growth/evaporation equation, deduced from equation (3.5), can be expressed as

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial r}[g_r C] \tag{4.19}$$

This is the advection equation in the radius space. Numerical algorithm for solving equation (4.19) is based on the PPM method of Colella and Woodward (1984), which is the third order accurate method in space. The boundary conditions imposed on the solution imply that no growth may occur into the smallest bin, $r_{\min}$, for cloud droplets and no growth may occur out of the largest bin, $r_{\max}$. No evaporation occurs into the largest bin for cloud droplets, but cloud droplets are allowed to evaporate out of the smallest bin via introducing a fictitious $0th$ bin. Since the radius grid is nonuniform, equation (4.19) cannot be solved exactly, and solution is associated with a numerical diffusion, especially for the largest bins, in which the bin width Δr is greatest; hence the Courant number, $g_r \Delta t / \Delta r$, is very small.

Tests of solution of the advection equation

The advection equation in the original CARMA v. 2.2 is solved using the PPM method of Colella and Woodward (1984). In the improved version of CARMA, the PPM method was replaced with the REA method, complemented with the minmod slope limiter (LeVeque, 2002). We run a standard test of the adopted algorithm to see its performance. A computer code has been developed to solve the advection equation through using this algorithm and the tested code later was plugged into the improved

version of CARMA in place of existing subroutines containing the PPM method. For the test of the developed code, the height of the numerical domain is chosen to be 1 m. The model is initialized with a square wave of amplitude $c = 1m^{-3}$ in the middle of the domain, at $\frac{1}{3}m < z < \frac{2}{3}m$. Above and below the initial wave location, $c = 0$. The wave is advected upwards with a constant vertical velocity of $1m/s$. The numerical domain has periodic boundaries, so the wave, disappearing through the upper boundary, reappears through the lower boundary of the domain. Since the explicit method is chosen to advance the solution of the advection equation in time, the time step must be limited by the CFL condition: $\Delta t \le w\Delta t/\Delta z$, i.e. Courant number must be less than or equal to unity. In case of the Courant number equal to unity, the advection equation is solved exactly, i.e. the wave moves along the domain with no change in shape or amplitude. We intentionally choose the Courant number to be less than unity, and see how the adopted algorithm works on this simple test problem. The solution obtained by the PPM method can be compared with the exact solution, i.e. solution that would take place, if the Courant number was chosen to be equal to unity. For that, we allow the wave to make one or more complete revolutions or cycles and return to its initial condition after the cycles, and compare the shape and amplitude of the obtained wave with the initial square-shaped wave. For the sake of showing its advantages over the upwind method, which is used for solving the sedimentation equation, and the pure REA method with no slope limiters, we compare our results for the REA method, complemented with the minmod slope limiter, with the exact solution (initial wave location), with the upwind method and with the pure REA method. We run the program till the wave makes 1 revolution and 5

revolutions, and compare the obtained results. We choose to use three grids for the test: a uniform grid with 10 vertical layers, a uniform grid with 100 vertical layers and a nonuniform grid with 100 vertical layers, in which there are 80 uniformly-spaced vertical layers in the initial wave location, $\frac{1}{3}m < z < \frac{2}{3}m$, 10 uniformly-spaced vertical layers below and above the initial wave location, so that the vertical layer thicknesses in the lower and upper parts of the domain are equal to each other.

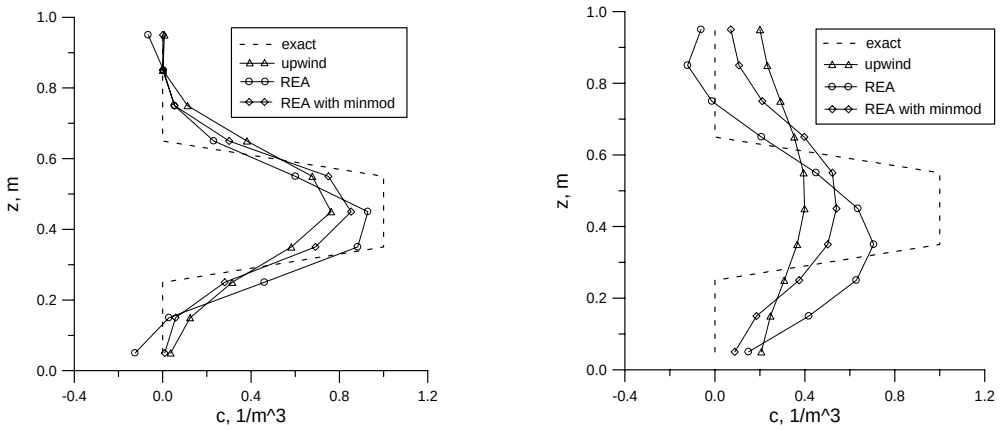
Figure 4-3 shows results of the test performed on a uniform grid containing 10 uniformly-spaced vertical layers. The time step is equal to $0.08333\,sec$, which gives the Courant number equal to 0.8333. Figure 4-3(a) shows results after 1 revolution, and Figure 4-3(b) – results after 5 revolutions. In both cases, the upwind method shows a great decline in the shape and amplitude of the wave, because it is numerically very diffusive. However, the values of the concentration remain positive everywhere in the domain. On the other hand, the REA method produces less numerical diffusion compared to the upwind method, i.e. the amplitude of the wave is closer to unity, but the solution has a phase lag; it lags somewhat behind the exact solution. Moreover, the REA method produces negative concentrations above and below the wave location. The solution obtained via using the REA method, complemented with the minmod slope limiter, is less diffusive than the upwind method, because in the regions with no sharp gradients in c, the method coincides with the REA method. At the same time, the method does not produce a phase lag; the method reduces to the upwind method, when the gradients are large. Also the REA method, complemented with the minmod slope limiter, does not produce negative concentrations similarly to the upwind method.

Figure 4-4 shows results of the test performed on a uniform grid containing 100 uniformly-spaced vertical layers. The time step is equal to $0.008\,sec$, which gives the Courant number equal to 0.8. Figure 4-4(a) shows results after one revolution, and Figure 4-3(b) – results after 5 revolutions. This figure shows better the advantages offered by the REA method, complemented with the minmod slope limiter, compared to Figure 4-3. Even after 5 revolutions, both the shape and the amplitude of the wave obtained via using this method closely approximate the exact solution, whereas the upwind method gives a 10% reduction in the amplitude of the wave and a triangle-like shape, and the REA method gives negative concentrations and wiggles due to a greater wave number allowed by the used fine mesh.

Figure 4-5 shows results of the test performed on a non-uniform grid, containing the number of uniformly-spaced vertical layers equal to 80 in the central third of the domain and 10 uniformly distributed grid cells above and below the central part of the domain. The time step is equal to $0.004\,sec$, which gives the maximum Courant number equal to 0.96 and minimum Courant number equal to 0.12. Figure 4-5(a) shows results after 1 revolution, and Figure 4-5(b) – results after 5 revolutions. The advantages of the REA method, complemented with the minmod slope limiter, over the upwind method and the pure REA method shown by this figure are similar to those demonstrated by Figure 4-3 and Figure 4-4. This figure demonstrates the functionality of the adopted method on a non-uniform grid. Also, it can be noted from Figure 4-5(a) that there is a strong inflexion on the downwind side of the wave where a transition from a coarse mesh to a fine mesh occurs, produced by the pure REA method; the inflexion does not appear when the minmod slope limiter is added to the algorithm.

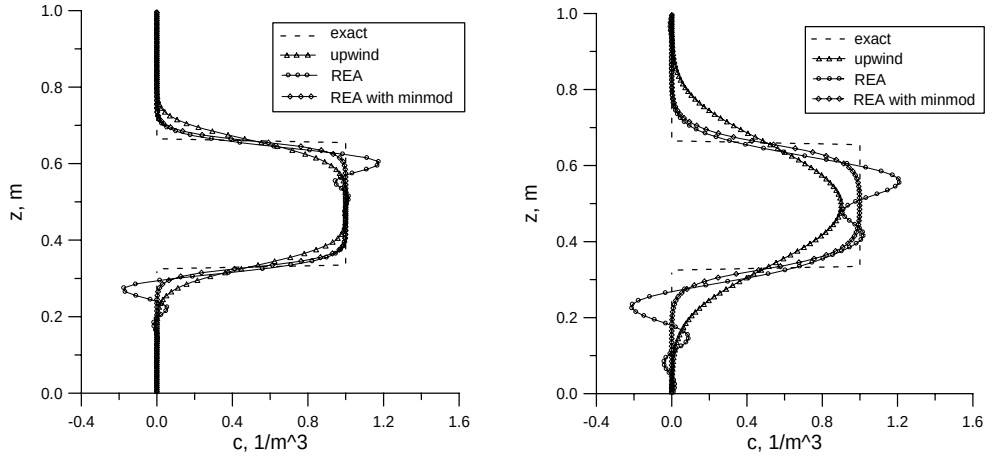
These tests give a confidence in the adopted method for solving the advection equation, i.e. the REA method, complemented with the minmod slope limiter. The chosen method provides a good accuracy of the solution, even in the regions of sharp discontinuities, and does not produce negative concentrations. Therefore the method can be used in the improved version of CARMA.

Tests of solving the diffusion equation are not required. In the improved version of CARMA we adopt the original algorithm from CARMA v. 2.2. The only change we made was correcting treatment of boundary conditions; some errors were found in imposing boundary conditions. Numerical tests of the solution of microphysical equations are given in Toon *et al.* (1988), and therefore they will not be given here.



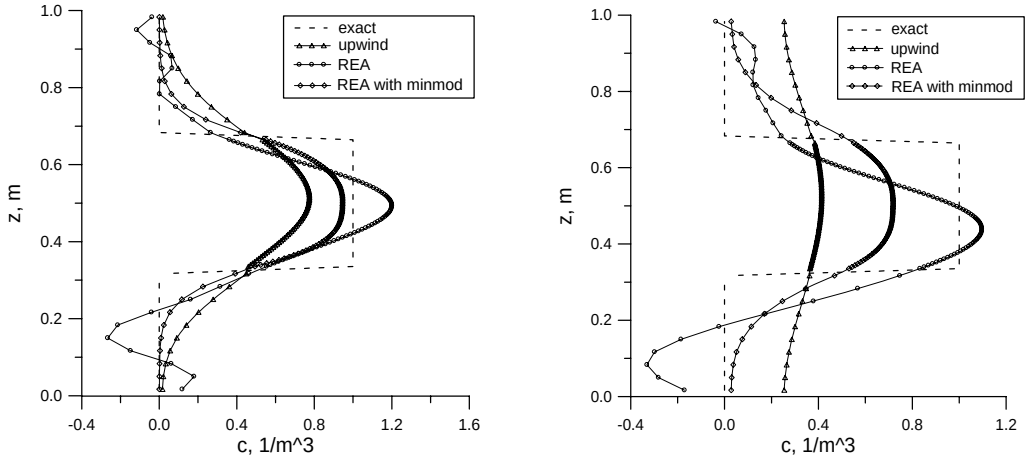
(a) 1 revolution (b) 5 revolutions

Figure 4-3. Tests of solving the advection equation by three methods: upwind, REA and REA with minmod slope limiter. Grid contains 10 uniformly-spaced vertical layers. Courant number is 0.8333. (a) results after 1 revolution; (b) results after 5 revolutions.



a) 1 revolution (b) 5 revolutions

Figure 4-4. Tests of solving the advection equation by three methods: upwind, REA and REA with minmod slope limiter. Grid contains 100 uniformly-spaced vertical layers. Courant number is 0.8. (a) results after 1 revolution; (b) results after 5 revolutions.



(a) 1 revolution (b) 5 revolutions

Figure 4-5. Tests of solving the advection equation by three methods: upwind, REA and REA with minmod slope limiter. Grid contains 100 nonuniformly-spaced vertical layers. 80 uniformly-spaced layers are located in the initial wave location, 10 uniformly-spaced layers are located above and below the wave location. Minimum Courant number is 0.12 and maximum Courant number is 0.96. (a) results after 1 revolution; (b) results after 5 revolutions.

CHAPTER 5

MODELING AND DISCUSSION OF RESULTS

Geographic region of N84 measurements

Models require a validation through comparing outputs produced by the model with available experimental data. Since low-level clouds are of interest, the results from an observational study of the structure and dynamics of a marine stratocumulus cloud over the North Sea, conducted by Nicholls between 11 am and 3 pm (local time) on July 22, 1982, suit our purposes very well. The details of the observational study can be found in Nicholls (1984), and hereafter Nicholls's results will be referred to as N84 data. Figure 5-1 shows a map of the region with a location of the area, where measurements were conducted, denoted by the L-shaped label in the center of the figure. The figure also demonstrates the horizontal wind field over the North Sea along with the $15°C$ and $20°C$ isotherms (dashed lines). The stratocumulus cloud in the figure is shown by the extensive shading.

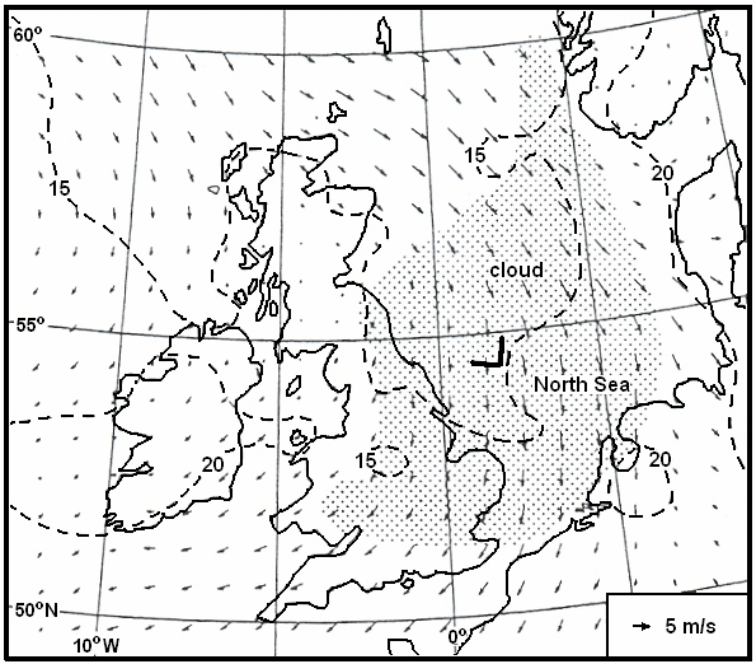


Figure 5-1. Map of a region over the North Sea, where observational data on a microstructure of a marine stratocumulus cloud from Nicholls (1984) were obtained. A location of the region, where measurements were conducted, is denoted by the L-shaped label. The figure demonstrates the horizontal wind field over the North Sea obtained on July 22, 1982. The 15°C and 20°C isotherms are denoted by the dashed lines. The marine stratocumulus cloud is demonstrated by the shading in the figure. 1° of latitude ≅ 111 km, 1° of longitude ≅ 64 km at latitude 55° (Replotted from Nicholls, 1984).

The N84 data also indicate that at the time of measurements no upper clouds, lying over the shown marine stratocumulus cloud, were observed. The measurements by aircrafts were performed along a region of the map denoted by the L-shaped label at several altitudes located above, below and inside the cloud. The nominal length of each of the legs in the L-shaped pattern was about 60 km. In the L-shaped region of interest, the top and base of the cloud were fluctuating in time with the time-mean height of the cloud top of 830 m and the time-mean height of the cloud base of 380 m during the time of the measurements.

Validation approach

A marine stratocumulus cloud studied in N84 was present before, during and after the flights. Unfortunately no data on the studied stratocumulus cloud and environmental conditions are available for the time preceding 11 am of July 22, 1982. Hence, reproducing the observed data using a time-dependent model can become difficult without having good initial conditions for the model. N84 data represent one snapshot in time, averaged between 11 am and 3 pm. Several approaches are available in the literature on initial conditions and the length of calculations for the cloud models. For example, Duynkerke and Driedonks (1987) initialize their model with atmosphere, that contains an observed in N84 cloud microstructure, and run the model till the moment when results start diverging significantly from N84 data (≈2 hours of simulation time), and then compare results with N84 measurements.

Bougeault (1985) initializes his model with cloud-free environmental conditions. The model is run for 7 days of simulation time till a steady-state is reached, and results for a cloud microstructure on noon of the fourth day of simulation time are compared with observations. Ackerman *et al.* (1995) initialize their model with cloud-free environmental conditions at midnight and run the model for 12 hours of simulation time, and then compare their results with N84 measurements. All three approaches are used in the models which contain radiation. This allows for keeping track of the solar zenith angle and simulating diurnal cycles. In addition, the models can reproduce turbulence intensities in the boundary layer based on the solar zenith angle because they contain the turbulence closure, such as the $k-\varepsilon$ model of turbulence (Launder and Spalding, 1972), to determine turbulent diffusion coefficients. CARMA v. 2.2 does not contain the

turbulence closure equations; the turbulent diffusion coefficients are parameterized in CARMA. Furthermore, in the improved version of CARMA, radiation is removed from model equations. Thus, none of the three approaches, described above, appears to be suitable for our purposes.

Zhang (2005), who uses CARMA v. 2.0, initializes his model with cloud-free environmental conditions and runs his model for 1 hour of simulation time. During 1 hour, vertical velocity oscillates three times in the pre-defined region of the cloud, i.e. parcels of air go up and down three times. Cloud droplets are formed during an upward motion (updraft), and evaporate during a downward motion (downdraft). Results are then time-averaged over all three updrafts and downdrafts and compared with N84 measurements. In the actual stratocumulus clouds, upward and downward motions of air occur at the same time, which cannot be reproduced using a one-dimensional model. Averaging over all updrafts and downdrafts roughly reproduces an actual situation occurring in stratocumulus clouds. Zhang (2005) also removes radiation from model equations solved by CARMA. Thus, the approach of Zhang (2005) can be adopted here. Details of the approach are given below.

Vertical velocity, w

According to Zhang (2005), the vertical velocity, w, in the cloud region can be defined through the following function of z and t:

$$w(z,t) = \cos\left(\pi \frac{z - z_c}{h_c}\right)\left[w_0 + w_1 \sin\left(\frac{2\pi t}{t_c}\right)\right] \qquad \text{for} \quad z_c - \frac{h_c}{2} < z < z_c + \frac{h_c}{2} \qquad (5.1)$$

where $z_c = 605\,m$ is the height of the cloud center, $h_c = 450\,m$ is the cloud's geometric thickness, $w_0 = 0.3\,m/s$ is the value of the vertical velocity at cloud's center at the beginning of the oscillation cycle, $w_1 = 1\,m/s$ is the amplitude of an oscillation in w at cloud's center, $t_c = 20\,min$ is the period of one oscillation. A graphical representation of w defined through equation (5.1) is given in Figure 5-2(a).

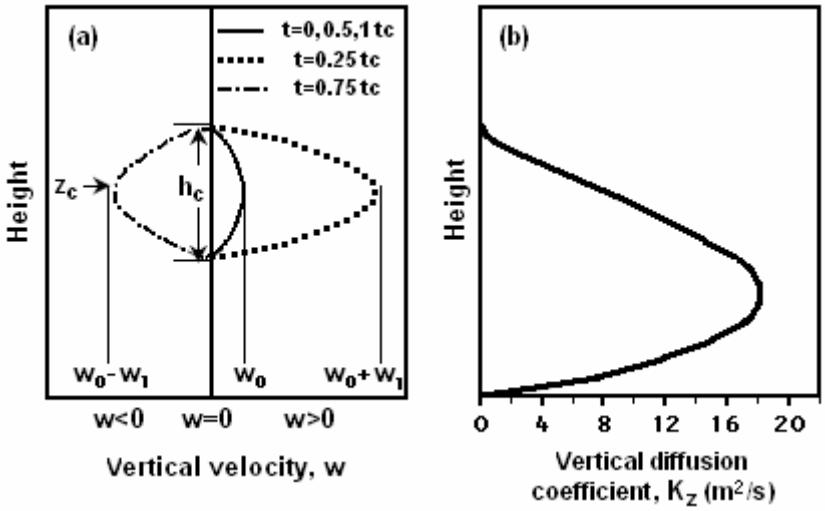


Figure 5-2. Vertical profiles of (a) vertical velocity during an oscillation period and (b)vertical diffusion coefficient (Replotted from Zhang, 2005)

Figure 5-2(a) demonstrates that the maximum of vertical velocity always occurs at cloud's center; w gradually decreases toward the base and top of the cloud. Above and below the cloud, w is kept as zero at all times.

Figure 5-2(a) also shows the values of w in the center of the cloud at various stages of the oscillation cycle. Each cycle starts with a small positive value w_0. At the end of the first quarter w grows to the maximum value $w_0 + w_1$, and after that w decreases to

the values w_0 and $w_0 - w_1$ at the end of the second quarter and third quarter respectively. During the fourth and last quarter w recovers its initial value w_0, and the cycle repeats.

Observational data of Yum and Hudson (2001) indicate that in low-level clouds updrafts are slightly greater than downdrafts, so on average, the air is rising in the cloud, but the mean vertical velocity usually does not exceed $0.5\,m/s$. Hence, $w_0 = 0.3\,m/s$ appears to be a suitable choice. The same data of Yum and Hudson (2001) indicate that the maximum in w in low-level clouds is usually on the order of $2\,m/s$ for both updrafts and downdrafts. Thus, $w_1 = 1\,m/s$ leads to the maximum updraft velocity $1.3\,m/s$ and the maximum downdraft velocity $0.7\,m/s$, which are close to the values measured by Yum and Hudson (2001). Based on the values of w_0, w_1 and h_c, Zhang (2005) estimates that $t_c = 20\,min$.

Vertical diffusion coefficient, k_z

According to the observations of Brost and Wyngaard (1978), the vertical turbulent diffusion coefficient, k_z, in the stratocumulus-topped marine boundary layer can be defined via the following empirical formula:

$$k_z = \kappa u_* z\left(1 - \frac{z}{h}\right)^{1.5}\left(1 - B_0\,\frac{d\theta}{dz}\right) \tag{5.2}$$

where $\kappa = 0.4$ is the von Karman constant, $u_* = 0.35\,m/s$ is the friction velocity, h is the boundary layer thickness, which is assumed to be the same as the height of the cloud top, B_0 is an empirical constant equal to $40\,m/K$. Zhang (2005) takes an upper limit for the product $B_0\left(d\theta/dz\right)$ equal to 0.5. A graphical representation of function $k_z(z)$ is given

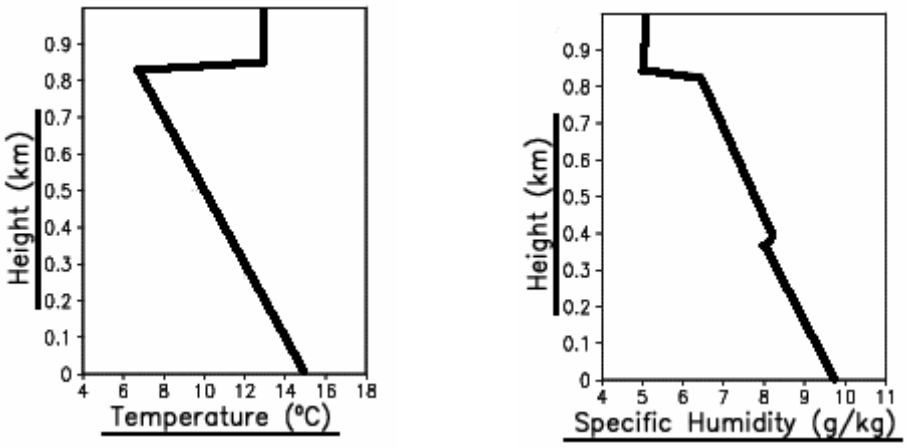
in Figure 5-2(b). k_z is equal to zero on the sea surface and on the top of the marine boundary layer (on the top of the cloud); the maximum in k_z is reached at a height of about $h/3$. Above the cloud top k_z is set to zero. Since both w and k_z are set to zero above the cloud top, no advection or diffusion occurs at those levels. The vertical turbulent diffusion coefficient for particles and water vapor can be taken as 1.2 times greater than that for momentum (Brost and Wyngaard, 1978). In the boundary layer, the molecular diffusion coefficient is by several orders of magnitude smaller than the turbulent diffusion coefficient; hence its contribution can be neglected in equation (5.2).

Temperature and humidity

According to Figure 5-1(b), the L-shaped region of measurements is located close to the $15°C$ isotherm. Hence, the sea surface temperature (SST) can be taken equal to $288K$, which is also, according to Tucker and Barry (1984), a climatological value for the month of July for the region of measurements. The initial vertical temperature profile follows a dry adiabat (decrease by 1°C per 100 meters of height) up to the cloud top height of 830 m, where the temperature jumps to the observed in N84 value of $286K$ (temperature inversion), and does not change with height above, as shown in Figure 5-3(a). All temperature-dependent properties, such as dynamic viscosity, thermal conductivity and density of the air, are calculated once before the time loop based on the adopted values of temperature.

Zhang (2005) takes the value of relative humidity within the pre-defined region of the cloud equal to 98.5%, as in Ackerman *et al.* (1995), for the initial vertical profile. Below the cloud region, the relative humidity is assumed to be lower and equal to 95%. N84

measurements indicate that above the cloud, the air is drier and specific humidity does not change with height taking the value of $5\,g/kg$ (relative humidity≈50%). The initial vertical profile of specific humidity, based on the adopted values of relative humidity, is shown in Figure 5-3(b).



(a) (b)

Figure 5-3. Initial vertical profiles of (a) temperature; (b) specific humidity.

Initial CN concentration

Some observations conducted over the sea demonstrate that the CN concentration decreases with height, other observations demonstrate the opposite (Pruppacher and Klett, 1978). Hence, the vertical profile of CN concentration is usually taken to be uniform over the sea (Jaenicke, 1993). The size distribution of CN can be expressed via a log-normal probability density function of the form:

$$\frac{dC_{CN}(r)}{d\ln r} = \frac{C_{CN_T}}{\sqrt{2\pi}\,\ln\sigma_g}\exp\left[-\left(\ln\frac{r}{r_g}\right)^2 \Big/ 2\left(\ln\sigma_g\right)^2\right] \tag{5.3}$$

where r_g is the geometric mean (mode) radius, σ_g is the geometric standard deviation, C_{CN_T} is the total number concentration of CN. Ackerman *et al.* (1995) initialize their model with $r_g = 0.05\mu m$, $\sigma_g = 2.5$ and $C_{CN_T} = 1000\, cm^{-3}$, and the material of CN is taken as ammonium bisulfate (NH_4HSO_4) in accordance with measurements of Covert (1988) in the remote North Pacific. Zhang (2005) adopts another size distribution of CN in the form of a sum of three log-normal size distributions, each of which has the form of equation (5.3). Three values of r_g, σ_g and C_{CN_T} for each of the three modes of the tri-modal size distribution of CN over the sea can be taken from Jaenicke (1993). After summing over the three modes, $C_{CN_T} = 203\, cm^{-3}$. This is much smaller than the total number concentration of CN measured by Covert (1988) in the remote North Pacific. Only a certain part of CN will be activated to cloud droplets. Since we adopt Zhang's treatment of temperature, relative humidity, vertical diffusion coefficient and vertical velocity, we take the total number concentration of CN equal to $C_{CN_T} = 200\, cm^{-3}$ as in Zhang (2005), but for convenience, use only a unimodal size distribution function of CN, as in Ackerman *et al.* (1995), given by equation (5.3) with $r_g = 0.05\mu m$, $\sigma_g = 2.5$. Figure 5-4 shows $C_{CN}(r)$ based on the adopted values. The material of CN is taken as ammonium bisulfate (NH_4HSO_4).

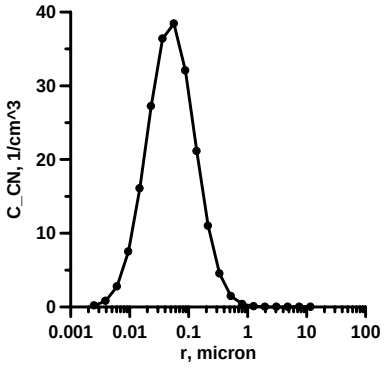


Figure 5-4. Initial log-normal size distribution of CN. Parameters of the lognormal size distribution are $r_g = 0.05\mu m$, $\sigma_g = 2.5$, $C_{CN_T} = 200\, cm^{-3}$.

Numerical details

The vertical column of interest, located between 0 m and 1000 m, is subdivided into 50 uniformly-spaced vertical layers, each of which is 20 m thick. 20 size bins are used for both CN and cloud droplets. The radii of both CN and cloud droplets increase exponentially from one bin to another with the ratio of volumes between adjacent bins equal to 3.8 for CN and 4.8 for cloud droplets, resulting in a radius grid for CN spanning from $r_{\min} = 0.0025\mu m$ to $r_{\max} = 11.75\mu m$ and for cloud droplets – from $r_{\min} = 0.05\mu m$ to $r_{\max} = 1032\mu m$. The time step size used for coagulation, sedimentation and vertical transport, which includes advection and diffusion, is 10 seconds, which is the default value in CARMA v. 2.2. Since the advection equation and sedimentation equation are solved explicitly, the model splits the time step into substeps in order comply with the CFL condition. The time step size, used for nucleation, condensational growth and evaporation, is also split into substeps. The minimum number of substeps is 2 and the maximum number is 1000. During a nucleation, CARMA chooses the maximum number of substeps, but during a condensational growth or evaporation, the number of time substeps drops to 2.

Analyzed cloud properties

Total number concentration of cloud droplets C_{drops_T} in cm^{-3}:

$$C_{drops_T} = \sum_{i=1}^{NBIN} C_{drops_i} \tag{5.5}$$

Mean radius r_{mean} in cm:

$$r_{mean} = \frac{\sum_{i=1}^{NBIN} r_i \, C_{drops_i}}{\sum_{i=1}^{NBIN} C_{drops_i}} \tag{5.6}$$

Liquid water content q_l in g/kg:

$$q_l = \frac{4}{3}\pi \frac{\rho_w}{\rho} 1000 \sum_{i=1}^{NBIN} r_i^3 \, C_{drops_i} \tag{5.7}$$

Specific humidity q_v in g/kg:

$$q_v = \varepsilon \, (s+1) 1000 \frac{e_s}{p} \tag{5.7}$$

Temperature T in K or $°C$

where C_{drops_i} is the number concentration of cloud droplets in bin i in cm^{-3}, r_i is the radius of cloud droplets in bin i in cm, $NBIN$ is the number of size bins, $\rho_w = 1000\, kg/m^3$ is the density of liquid water, $s = RH/100 - 1$ is the supersaturation of water vapor with respect to liquid water, $RH = e/e_s \cdot 100\%$ is the relative humidity in %, e is the ambient water vapor pressure in $dyne/cm^2$, e_s is the saturation water vapor pressure over liquid water in $dyne/cm^2$, $\varepsilon = \frac{R_{air}}{R_{H_2O}} = \frac{M_w}{M_{air}} = \frac{18}{29} = 0.622$, R_{H_2O} is the gas

constant of water vapor ($462\,J/kg\,K$), 1000 is for converting kg/kg to g/kg. Specific humidity and water vapor mixing ratio are defined by the same expression, and therefore the terms can be used interchangeably.

Even though N84 reported the radius of the mean cloud droplet volume, $r_v = \left[3\rho q_l / 4\pi \rho_w C_{drops}\right]^{1/3}$, as a measure of the mean cloud droplet size, Zhang (2005) converted data on r_v from N84 to the values of r_{mean} for convenience purposes and compared his model results on r_{mean} with r_{mean} from N84 measurements. We choose to use Zhang's (2005) approach.

Model results vs. N84 data

For comparing results from our model with N84 data, we, first, scan figures from Zhang (2005), and then extract N84 data from the figures into ASCII files. The created ASCII files are then rewritten into binary files in the GrADS format. Since results from the improved version of CARMA also come in the GrADS format (see appendix VII), our model results can be directly compared to the scanned data (N84 measurements) from Zhang's (2005) figures.

In Zhang (2005), the model is run for one hour of simulation time, i.e. three cycles of an oscillating vertical velocity, and then the obtained results are averaged over all three cycles and compared with N84 data. The idea behind this approach is that this time-averaging roughly corresponds to a cloud microstructure averaged over the area of the cloud; the number of parcels of air performing an upward motion and the number of

parcels of air performing a downward motion in stratocumulus clouds are roughly equal according to Pruppacher and Klett (1978).

There is a difference between how we compare our model data vs. N84 measurements and how Zhang compares his data vs. the same N84 data set. In both cases, the model is free of cloud droplets at the beginning of the run (cloud-free environmental conditions), but the relative humidity in the pre-defined cloud region and concentrations of CN at the beginning of the run are relatively high; thus, the beginning of the first cycle of oscillating vertical velocity experiences an activation of CN to cloud droplets more frequently than the beginning of the second or third cycles, as can be witnessed from the evolution of supersaturation at the center of the cloud's thickness in Figure 5-5(a); this signifies that cloud droplets, formed during the first cycle, tend to be smaller than cloud droplets formed during subsequent cycles. If one takes into account the fact that a time-average motion of air through the cloud is upward based on the adopted function $w(z,t)$, then it can be understood that after the first cycle cloud droplets may not evaporate completely; hence the second cycle starts with a certain cloud microstructure. As can be noted, for example, from the temporal evolution of supersaturation at the center of the cloud's thickness in Figure 5-5(a), the third cycle starts at environmental conditions close to those at the beginning of the second cycle. Based on the arguments, it may be concluded that a somewhat more realistic representation of the stratocumulus cloud can be obtained, if simulation results are averaged in time only over the second and third cycles, i.e. if the first cycle is excluded from time-averaging. This is our approach that we adopt here.

Comparison of our model results with N84 measurements is given in Figures 5-5 – 5-13. Dashed lines in the figures represent initializations, circles – N84 data, solid lines are results from the improved version of CARMA, averaged over the second and third cycles of an oscillating vertical velocity.

Figure 5-6 shows comparison between the measured N84 vertical profile of temperature and the vertical profile obtained via our model. The initial vertical profile of temperature is also included. Taking into account uncertainties in measurements of temperature, $\pm 1°C$, given in Nicholls and Leighton (1986) regarding N84 data, the improved version of the CARMA model closely reproduces the N84 data; the difference between the model and N84 data lies within the experimental uncertainties. As can be noted from the figure, there is a temperature increase relative to the initial temperature profile by about $2°C$ in the upper part of the cloud due to the latent heat release during nucleation and condensation. Above the cloud top, the temperature remains at its initial value of $13°C$ because no advection or diffusion occurs at those levels. Below the cloud base, only the turbulent diffusion is important, which is by itself not significant enough to yield noticeable changes in temperature.

Figure 5-7 shows comparison between the measured in N84 vertical profile of specific humidity and the vertical profile obtained via our model. The initial vertical profile of the specific humidity is also included in the figure. No experimental uncertainties of measurement of specific humidity were reported. In addition to the experimental uncertainty of measurements of temperature, the second uncertainty, that was reported in Nicholls and Leighton (1986) regarding N84 data, was the uncertainty in measurement of total water content ($q_T = q_v + q_l$) of $\pm 0.5\, g/kg$. The model reproduces

N84 data closely in the pre-defined cloud region. Below the cloud, the model overestimates the specific humidity by about $0.5\,g/kg$, which is identical to results in Zhang (2005) and very close to results in Ackerman *et al.* (1995) who overestimate q_v by $1\,g/kg$ under the cloud base. Since no advection or diffusion of water vapor above the cloud top is present in our model, the value of q_v does not change during the simulation time at those levels.

Figure 5-8 shows comparison between the measured in N84 vertical profile of the total number concentration of cloud droplets, C_{drops_T}, and the vertical profile obtained via our model. The model reproduces closely N84 data. The total number of cloud droplets remains almost constant in the cloud region as was observed in N84.

Figure 5-9 shows comparison between the values of r_{mean}, taken from Zhang (2005) who converted the r_v values from N84 to the values of r_{mean}, and the values of r_{mean} produced by our model. The model reproduces closely N84 data. The mean radius slightly increases with height through the cloud as was observed in N84.

Figure 5-10 shows comparison between the measured in N84 vertical profile of the liquid water content, q_l, and the vertical profile obtained via our model. The model reproduces closely N84 data. The biggest difference between the model data and N84 data is in the upper part of the cloud ($0.3\,g/kg$). Liquid water content increases with height as was observed in N84. As can be noted, the increase in the liquid water content with height is gained through the increase in the mean radius rather than through the increase in the total cloud droplet concentration, as was also discussed in N84, which

indicates that the growth of cloud droplets through condensation of moisture dominates over the gravitational coalescence mechanism.

Figures 5-11, 5-12 and 5-13 show comparison between the measured in N84 size distributions of cloud droplets at three heights: 730 m, 480 m and 300 m, respectively, and size distributions of cloud droplets at those heights obtained via our model. The model reproduces closely N84 data for the range of cloud droplet radii, for which the data are available. Near the cloud top, at 730 m, the peak at $r = 10\mu m$ in the cloud droplet number is not as pronounced in the model results as it is in the measurements (similarly to Zhang, 2005, and Ackerman *et al.*, 1995). In the lower region of the cloud, at 480 m, the model underpredicts the number of small cloud droplets with $r < 10\mu m$. In the tails of the distributions ($r > 100\mu m$) at all heights, our model reproduces the cloud droplet concentrations more closely than the model of Zhang (2005) and the model Ackerman *et al.* (1995), whose models somewhat underpredict the cloud droplet numbers. Near the top of the cloud, at 730 m, where supersaturation is highest, the model shows another peak in total cloud droplet concentration at $0.14\mu m$, which corresponds to the nucleation mode.

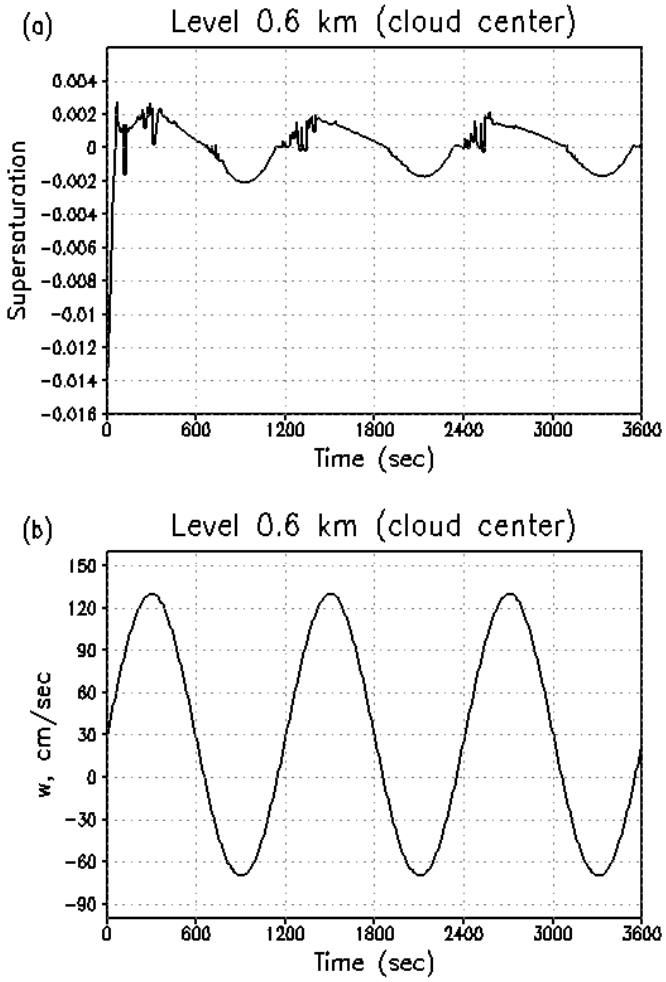


Figure 5-5. Temporal evolutions of (a) supersaturation and (b) vertical velocity in the center of the cloud. Initial size distribution of CN is lognormal with geometric mean radius 0.05 μm, σ_g=2.5, and the total number of CN per cm^3 is 200. The initial size distribution of CN is assumed to be uniform in the vertical direction. Visualization is accomplished using GrADS.

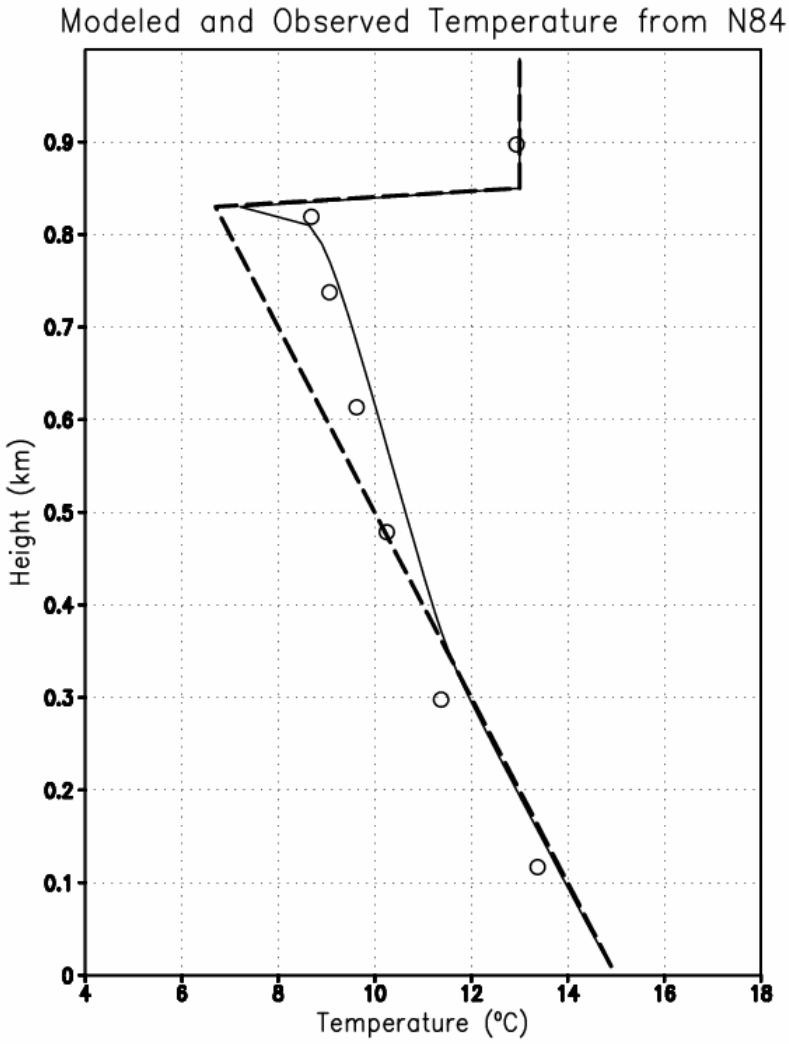


Figure 5-6. Vertical profile of temperature. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity, dashed lines are model initializations.

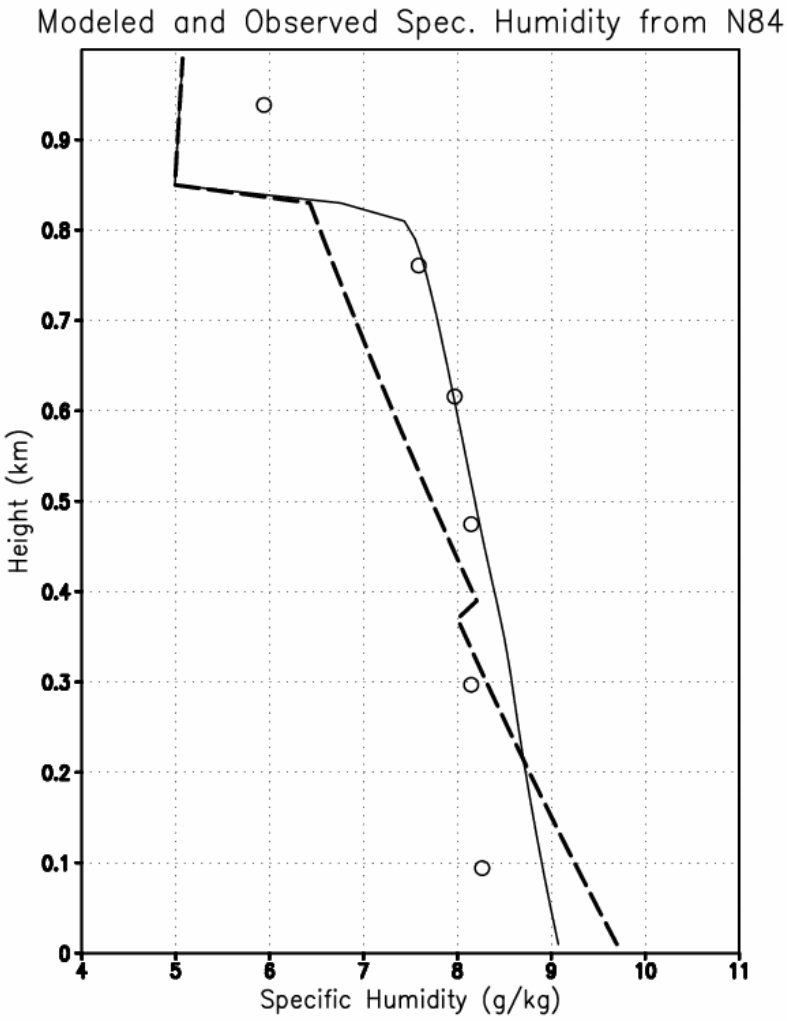


Figure 5-7. Vertical profile of specific humidity. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity, dashed lines are model initializations.

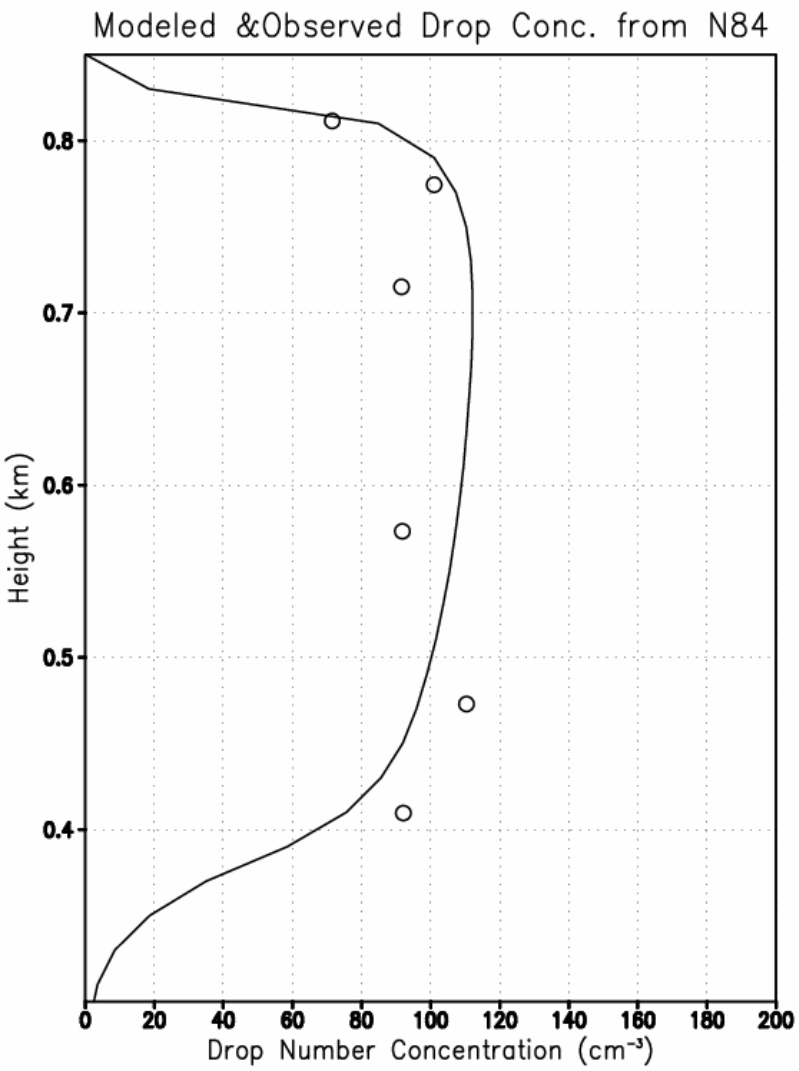


Figure 5-8. Vertical profile of cloud droplet number concentration. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

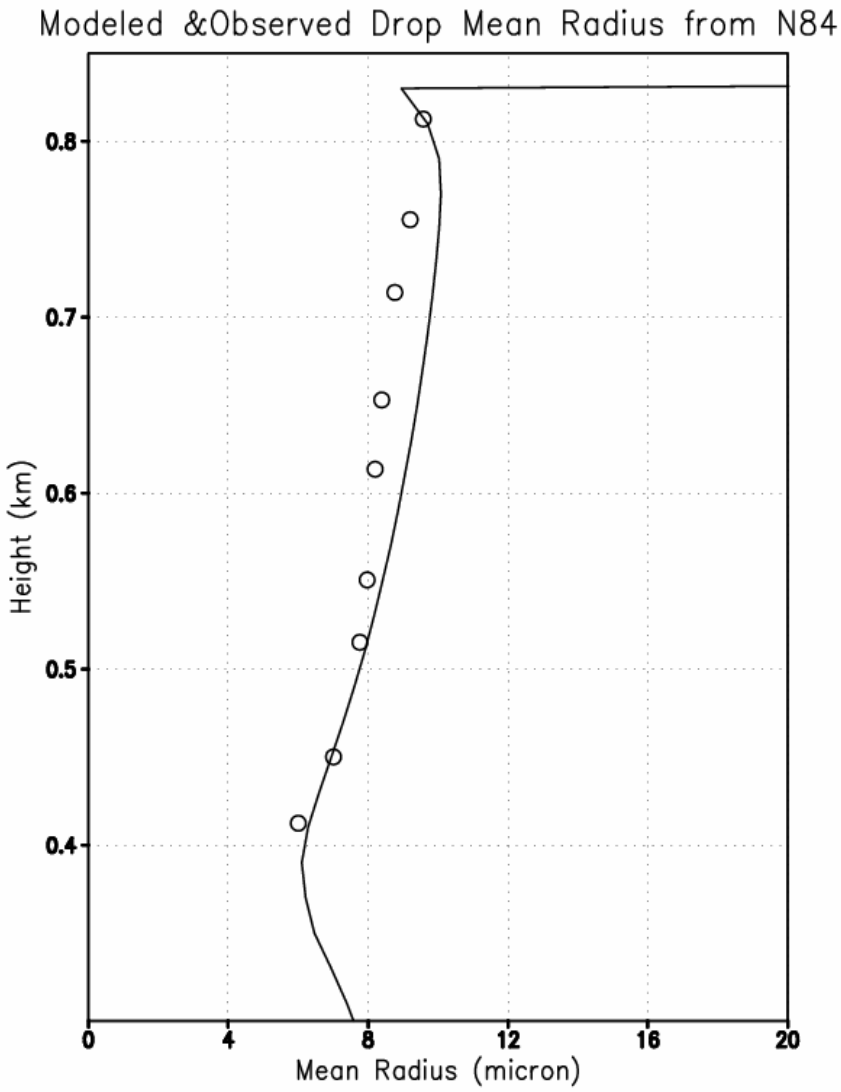


Figure 5-9. Vertical profile of mean radius of cloud droplets. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

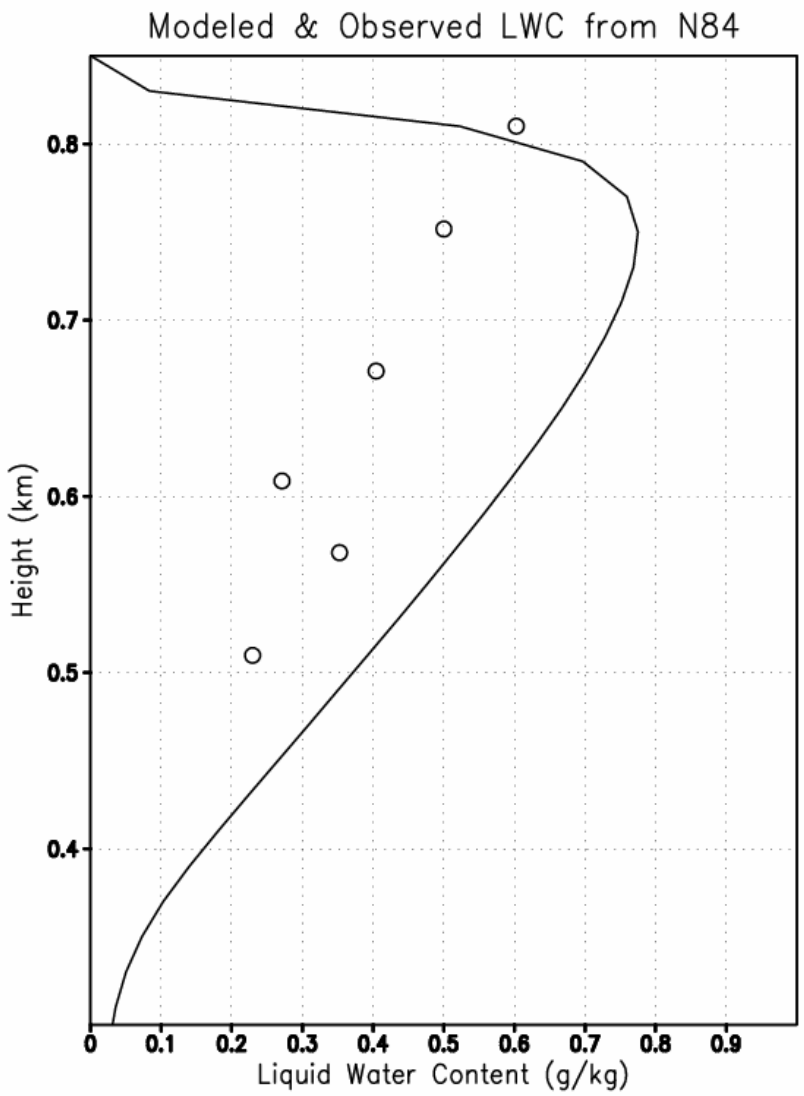


Figure 5-10. Vertical profile of liquid water content. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

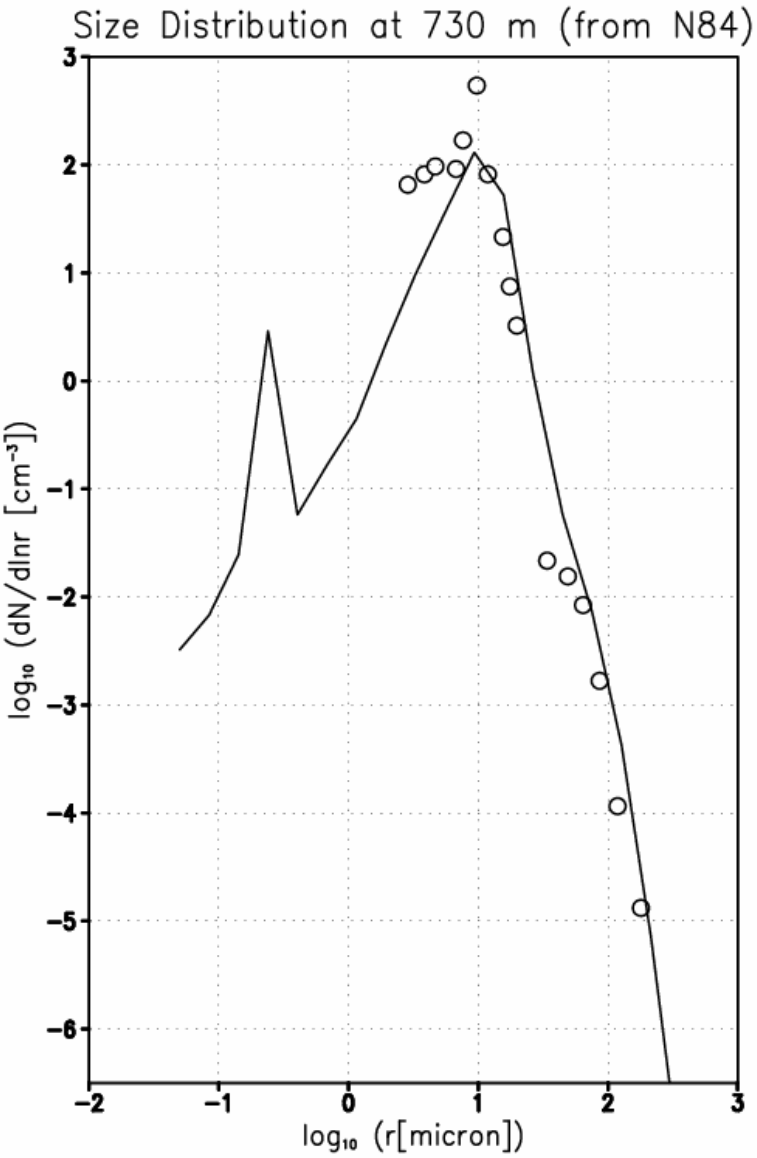


Figure 5-11. Size distribution of cloud droplets at height 730 m. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

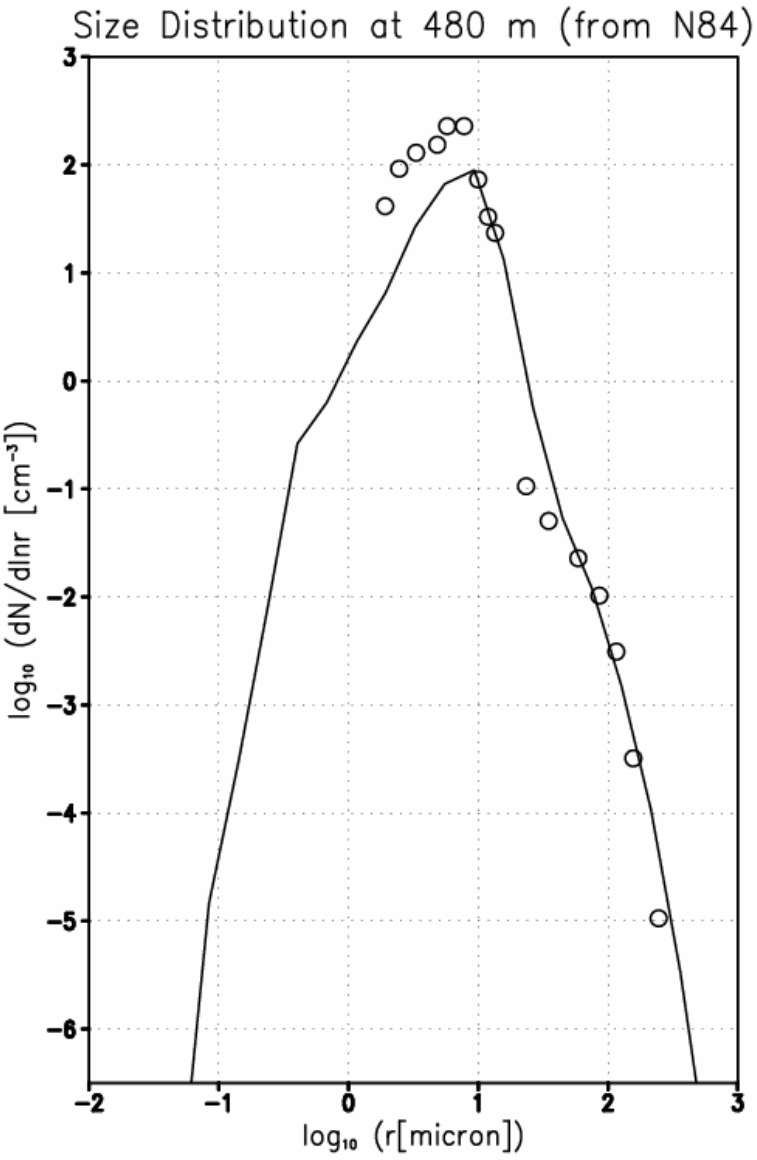


Figure 5-12. Size distribution of cloud droplets at height 480 m. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

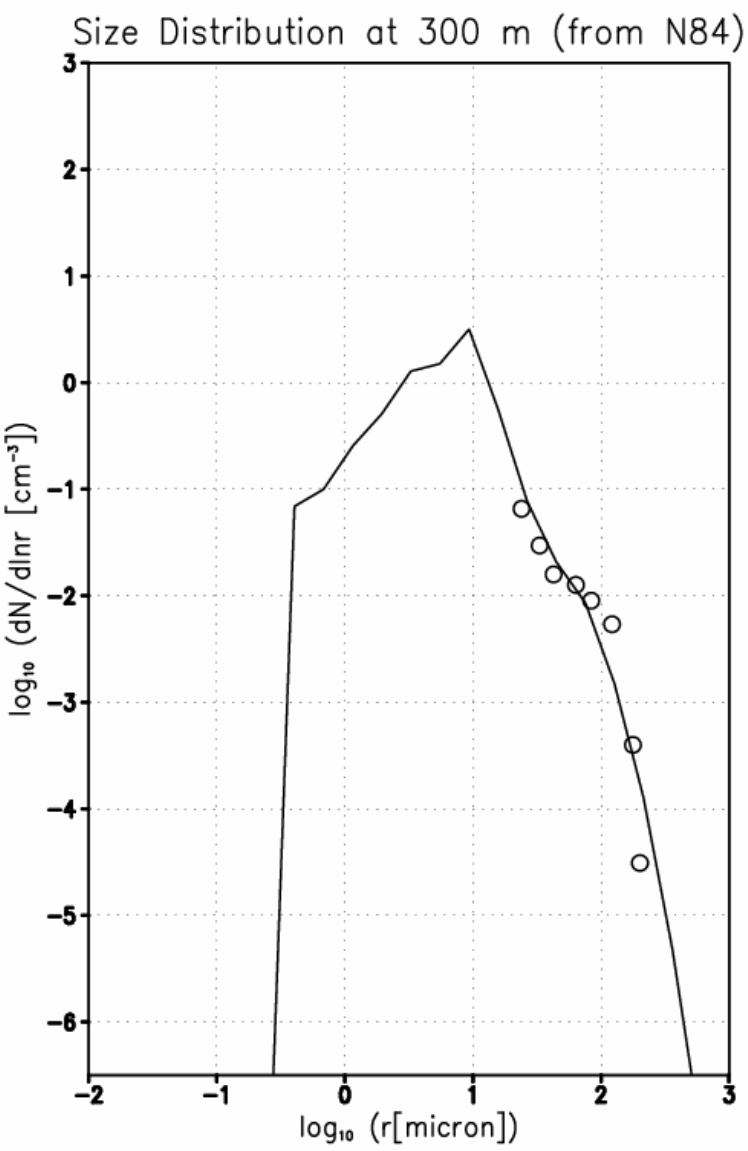


Figure 5-13. Size distribution of cloud droplets at height 300 m. Circles - N84 measurements, solid lines - model data averaged over the second and third cycles of oscillating vertical velocity.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The improved version of the CARMA model for calculating the size distributions of atmospheric aerosols and cloud droplets was developed from the original CARMA v. 2.2. The microphysical part of the CARMA model was isolated from transport and radiation subroutines and prepared for coupling with climate models. The vertical transport subroutines of original CARMA v. 2.2 were changed. The PPM method of solving the advection equation was replaced with the REA method, complemented with the minmod slope limiter. Sedimentation was isolated from vertical advection, and the sedimentation equation is solved using the upwind method. The ability of the improved CARMA model to reproduce the observed microstructure of a marine stratocumulus cloud over the North Sea was tested. The improved model closely reproduces most of the observed cloud and aerosols effects.

Recommendations for future work

The most direct application of the developed model is to couple it with the existing climate models, where the aerosol and cloud representations are simplified through a use of parameterizations. Using the microphysical module inside the climate models could possibly make the numerical predictions of climate performed by climate models somewhat more accurate.

The testing of the model on reproducing numerically Nicholls's observational data on stratocumulus clouds over the North Sea demonstrates robustness of the model. Hence, the model can be used for these types of clouds. In perspective, the model can be used to complement experimental data on clouds where the size distributions on clouds are not readily available, and most of the time, only integral properties are measured, such as the total water contents, the mean radius of particles, etc. The model can be applied for generating data on size distributions of cloud droplets in clouds observed over the Californian coast of the US through the use of the observational data from the ARM (Atmospheric Radiation Measurement) mobile facility of the US Department of Energy (Miller and Slingo, 2007).

APPENDIX I

NOMENCLATURE

A_k	curvature factor in the expression for condensational growth rate
A_s	solute factor in the expression for condensational growth rate
Av	Avogadro's number ($Av = 6.02252 \cdot 10^{23} mol^{-1}$)
B	mobility of particles in the Einstein relation
B_0	empirical constant in the expression for vertical diffusion coefficient
C	number concentration of CN or cloud droplets
C_{CN}	number concentration of CN
$C_{CN\,T}$	total number concentration of CN
C_{drops}	number concentration of cloud droplets
C_{drops_T}	total number concentration of cloud droplets
C_d	drag coefficient
C_p	specific heat at constant pressure for the air
C_{suth}	Sutherland's constant for the gas
c	concentration of an arbitrary tracer
D_p	particle diffusion coefficient
D_v	diffusivity of water vapor in air
E_{coal}	coalescence efficiency of particles
E_{coll}	collision efficiency of particles
e	ambient water vapor pressure
e_s	saturation water vapor pressure over liquid water
F_{adv}	advection flux
F_{dif}	diffusion flux
F_{sed}	sedimentation flux
F_t	ventilation correction to account for effect of sedimentation on heat flux
F_v	ventilation correction to account for effect of sedimentation on vapor flux
G	mass concentration of water vapor
g	acceleration due to gravity
g_r	rate of condensational growth or evaporation of cloud droplets
H	scale height
Hm_1	metric factor in the transport equation for the x-direction
Hm_2	metric factor in the transport equation for the y-direction
h	thickness of stratocumulus-topped boundary layer

h_c	cloud's geometric thickness
K^B	Brownian coagulation kernel
K^C	kernel of the integro-differential equation for coagulation
K^{DE}	coagulation kernel due to convective Brownian diffusion enhancement
K^{GC}	gravitational collection kernel
Kn	Knudsen number
K_t	thermal conductivity of dry air
k_B	Boltzmann constant ($k_B = 1.38054 \cdot 10^{-23}\ J/K$)
k_G	eddy diffusivity for water vapor
k_h	eddy diffusivity for heat
k_m	eddy diffusivity for momentum (turbulent viscosity coefficient)
k_p	eddy diffusivity for particles
k_x	turbulent diffusion coefficient in the x-direction
k_y	turbulent diffusion coefficient in the y-direction
k_z	turbulent diffusion coefficient in the z-direction
L_v	latent heat of water evaporation
M_s	molecular weight of the material of the core
M_w	molecular weight of water
m_s	mass of the core of a cloud droplet
n_{vap}	saturation vapor pressure (number density of water vapor molecules)
n_∞	number density of water vapor molecules in the ambient air
Pr	Prandtl number for the air
p	dry air pressure
p_{ref}	reference pressure
q_l	liquid water content
q_v	specific humidity or water vapor mixing ratio
r	independent variable for the radius space
r_g	geometric mean radius
r_{mean}	arithmetic mean radius
r_v	radius of the mean cloud droplet volume
r'	dummy argument for the radius of particles
R	universal gas constant ($R = 8.3143\ J/mol \cdot K$)
R_{air}	gas constant of air ($R_{air} = 287\ J/kg\ K$)
R_{H_2O}	gas constant of water vapor ($R_{H_2O} = 462\ J/kg\ K$)
R_n	removal rate of particles in the particle transport equation
Re_p	particle Reynolds number
RH	relative humidity

$r_{\min}$	smallest radius of particles
$r_{\max}$	largest radius of particles
S	supersaturation of water vapor with respect to liquid water
Sc	particle Schmidt number
S_n	source of particles in the particle transport equation
t	time
t_c	period of one oscillation of vertical velocity
T	absolute temperature
T_{ref}	reference temperature
T_0	triple-point temperature
u	x-component of velocity (zonal wind speed, east-west)
u_*	friction velocity
V_g	mean thermal velocity of air molecules
Vm	metric factor in the transport equation for the z-direction
V_p	average kinetic velocity of a particle
v	y-component of velocity (meridianal wind speed, north-south)
v_f	particle terminal fall velocity (sedimentation velocity)
w	z-component of velocity (vertical velocity of the wind)
w_0	vertical velocity value at $t = 0$ in cloud's center
w_1	amplitude of an oscillation of vertical velocity
x	horizontal coordinate in the east-west direction
y	horizontal coordinate in the north-south direction
z	vertical coordinate
z_c	height of the cloud center
β	slip factor in expression for sedimentation velocity
δ	correction factor in expression for Brownian coagulation kernel
κ	von Karman constant ($\kappa = 0.4$)
λ	mean free path of air molecules
μ	dynamic viscosity of air
μ_{ref}	reference viscosity at reference temperature
ν_d	dissociativity of the dissolved material of the cloud droplet's core
ρ	air density
ρ_p	density of the material of the particle
ρ_w	density of liquid water
σ_G	turbulent Prandtl number for water vapor
σ_g	geometric standard deviation
σ_h	turbulent Prandtl number for heat
σ_p	turbulent Prandtl number for particles

$\sigma_{w/a}$	surface tension of water-air interface
Δt	time step size
Δz	vertical layer thickness (mesh size)
θ	potential temperature
Φ_s	practical osmotic coefficient

APPENDIX II

SCHEMATIC OF THE ORIGINAL CARMA V. 2.2

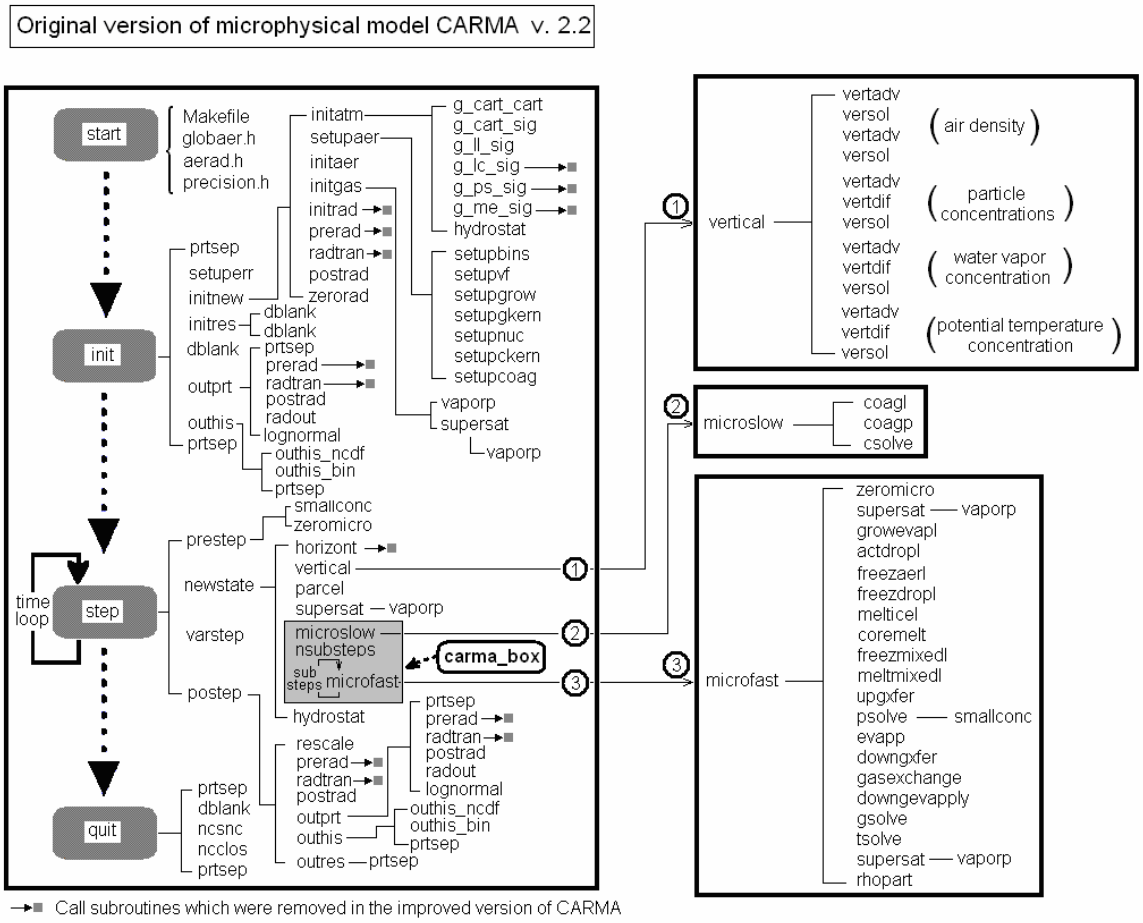


Figure II-1. Schematic of the original CARMA v. 2.2

APPENDIX III

SCHEMATIC OF THE IMPROVED VERSION OF CARMA

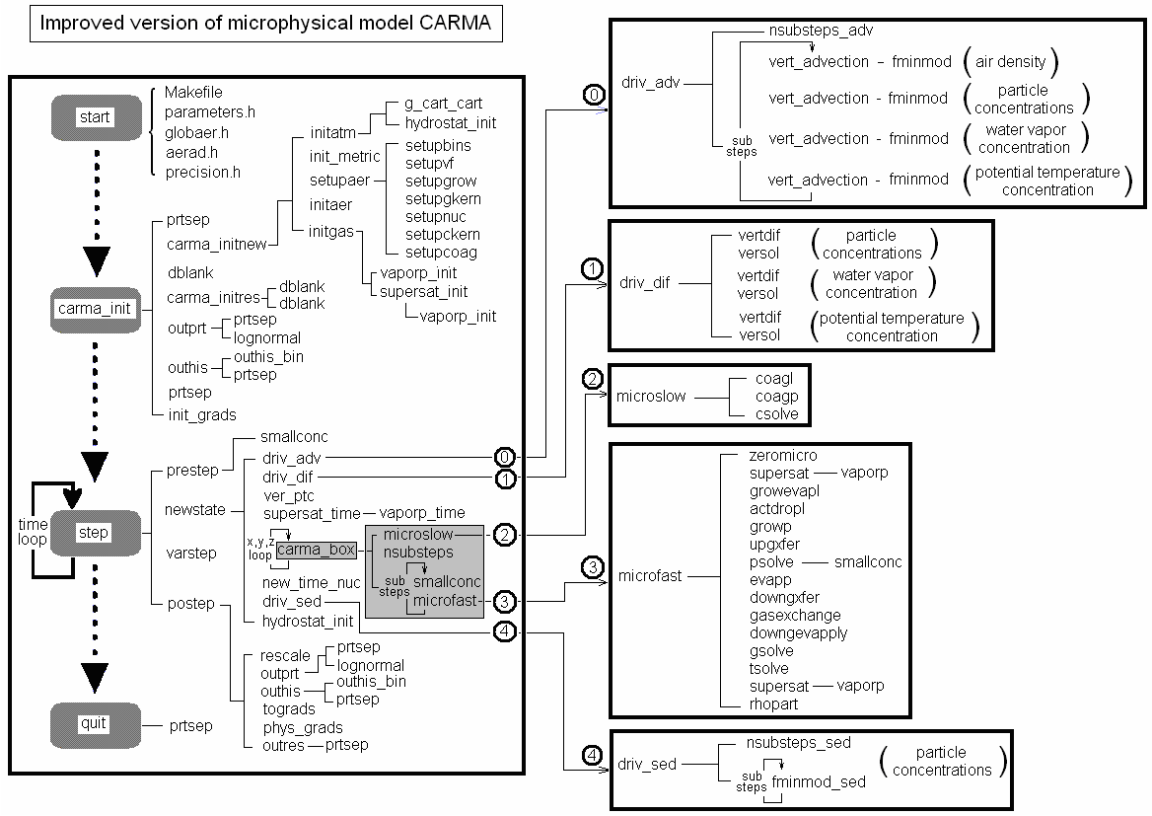


Figure III-1. Schematic of the improved version of CARMA

Note: Block 1 from the original CARMA v. 2.2 was subdivided into three blocks: 0, 1, and 4

APPENDIX IV

LIST OF SUBROUTINES IN THE IMPROVED VERSION OF CARMA

Table IV-1. Subroutines in the improved version of CARMA, their functions and subroutines called in

\	Subroutine	Function	Subroutines, called in ( )-routine included in the same file
aer	main.f	Main program	Call carma_init, step, quit
aer	carma_init.f	Performs all initializations	Call prtsep, carma_initnew, dblank, carma_initres, outprt, outhis, prtsep, init_grads
aer	step.f	Performs all calculations necessary for one time step	Call prestep, newstate, varstep, postep
aer	quit.f	Closes all opened files	Call prtsep
aer	prtsep.f	Outputs separator line	Call --
aer	carma_initnew.f	Initializes the model for a new model simulation (cold start)	Call initatm, init_metric, setupaer, initaer, initgas
aer	dblank.f	Finds non-blank char	Call --
aer	carma_initres.f	Initializes the model from a restart	Call dblank
aer	initatm.f	Initializes atmospheric structure, grid-related	Call (g_cart_cart), hydrostat_init
aer	init_metric.f	Initializes metric scale factors: xmet, ymet, zmet for each type of supported grid coord system	Call --
aer	setupaer.f	Defines mapping arrays and time-independent parameters for aerosol & cloud microphysics, setup aerosol types, composition, mass density, size grid	Call setupbins, setupvf, setupgrow, setupgkern, setupnuc, setupckern, setupcoag
aer	setupbins.f	Sets up particle size bins	Call --
aer	setupvf.f	Evaluates particle fall velocity	Call --
aer	setupgrow.f	Defines time-independent parameters to calculate evaporation /condensational growth	Call --
aer	setupgkern.f	Defines radius-dependent but time-independent parameters to calculate evaporation /condensational growth	Call --
aer	setupnuc.f	Calculates critical supersaturation for CN activation to cloud droplets	Call --
aer	setupckern.f	Evaluates coagulation kernels	Call --

aer	setupcoag.f	Sets up mapping arrays and pre-computes factors for coagulation	Call --
aer	initaer.f	Initializes particle concentration-number concentration, core mass, size distribution	Call --
aer	initgas.f	Initializes atmospheric profile of water vapor	Call vaporp_init, supersat_init
aer	hydrostat_init.f	Updates pressure by hydrostatic integration	Call --
aer	hydrostat.f	Updates pressure by hydrostatic integration	Call --
aer	init_grads	Outputs bin-independent initial conditions into GrADS files	Call --
aer	prestep.f	Handles all preliminary setup at the beginning of every timestep, saves model state.	Call smallconc
aer	smallconc.f	Limits all particle concentrations in a grid box to SMAL_PC	Call --
aer	varstep.f	Adjusts time-step, based on concentrations changed	Call --
aer	postep.f	Handles all post-timestep processing at the end of every timestep.	Call rescale, outprt, outhis, tograds, phys_grads, outres
aer	newstate.f	Updates state variable with new values at the current simulation time	Call drive_adv, driv_dif, ver_ptc, supersat_time, carma_box, new_time_nuc, driv_sed, hydrostat_init
aer	driv_adv.f	Drives vertical advection (air density, particles, water vapor, potential temperature concentration)	Call (nsubsteps_adv), vert_advection
aer	vert_advection.f	Calculates vertical advection using second order REA with minmod slope limiter	Call (fminmod)
aer	driv_dif.f	Drives the vertical diffusion calculation	Call vertdif, versol
aer	vertdif.f	Calculates vertical transport rate (by diffusion only)	Call --
aer	versol.f	Solves the vertical diffusion equation (particles, water vapor, potential temperature concentration)	Call --
aer	driv_sed.f	Drives sedimentation (particles)	Call (nsubsteps_sed), (fminmod_sed)
aer	ver_ptc.f	Adds constant source term (for the first 5 timesteps, for fog simulation) to the vertical transport equation for potential temperature concentration (by default, the constant term is zero)	Call --
aer	supersat.f	Evaluates supersaturations for all gases, called at one spatial point	Call vaporp

aer	supersat_init.f	Evaluates supersaturations for all gases, called at one spatial point	Call vaporp_init
aer	supersat_time.f	Evaluates supersaturations for all gases, called at one spatial point	Call vaporp_time
aer	vaporp.f	Calculates vapor pressure for all gases over entire spatial grid	Call --
aer	vaporp_init.f	Calculates vapor pressure for all gases over entire spatial grid	Call --
aer	vaporp_time.f	Calculates vapor pressure for all gases at one spatial grid point	Call --
aer	carma_box.f	Drives the aerosol microphysics calculations	Call microslow, nsubsteps, smallconc, microfast
aer	microslow.f	Drives potentially slower microphysics calculations-to allow for time splitting of coagulation at a different timestep size from other microphysical calculations	Call coagl, coagp, csolve
aer	coagl.f	Calculates coagulation loss rates	Call --
aer	coagp.f	Calculates coagulation production terms	Call --
aer	csolve.f	Calculates new particle concentration from coagulation microphysical processes	Call --
aer	nsubsteps.f	Calculates the number of sub-timesteps for the current model spatial point, find mass bin close to mode mass. The maximum time-step is the growth rate of the bin divided by the mass bin width /2	Call --
aer	microfast.f	Drives fast microphysics calculation (CN activation to cloud droplets, condensational growth, evaporation)	Call zeromicro, supersat, growevapl, actdropl, growp, upgxfer, psolve, evapp, downgxfer,gasexchange, downgevapply, gsolve, tsolve, supersat, rhopart
aer	zeromicro.f	Zeros the fast microphysics sinks and sources, at one spatial point per call	Call --
aer	growevapl.f	Evaluates particle loss rate due to condensational growth and evaporation	Call --
aer	actdropl.f	Evaluates particle loss rate due to nucleation (cloud droplet activation only)	Call --
aer	growp.f	Calculates particle source terms due to growth	Call --
aer	upgxfer.f downgxfer.f	Calculates particle source/sink terms due to element transfer processes	Call --

aer	psolve.f	Calculates new particle concentration due to faster non-coagulation microphysical processes (csolve for coagulation)	Call smallconc
aer	evapp.f	Calculates particle source term due to evaporation	Call --
aer	gasexchange.f	Calculates total production of gases due to nucleation, growth, and evaporation	Call --
aer	downgevapply.f	Applies evaporation and nucleation production term to particle concentration	Call --
aer	gsolve.f	Calculates new water vapor concentration	Call --
aer	tsolve.f	Calculates new temperature due to microphysics	Call --
aer	rhopart.f	Calculates new average particle densities	Call --
aer	new_time_nuc.f	Manages index of smallest bin nucleated from group inucmin and nucleation update time time_nuc	Call --
aer	rescale.f	Rescales vertical winds and diffusion coefficients	Call --
aer	outprt.f	Outputs info about the current timestep	Call prtsep, lognormal
aer	lognormal.f	Fits log-normal particle number distribution	Call --
aer	outhis.f	Controls history file output	Call outhis_bin, prtsep
aer	outhis_bin.f	Outputs model state to history file	Call --
aer	outres.f	Outputs info for restart file	Call prtsep
aer	tograds.f	Outputs bin-dependent information about the current timestep into GrADS files	Call --
aer	phys_grads.f	Outputs bin-independent information about the current timestep into GrADS files	Call --

APPENDIX V

VARIABLE NAMES IN THE MICROPHYSICAL BLOCK CARMA_BOX

In the improved version of CARMA, the microphysical driver "carma_box" interacts with the remaining time-dependent non-microphysical subroutines of CARMA (with the subroutine "newstate" that calls "carma_box") through the list of arguments. For convenience, original variables of CARMA v. 2.2 were replaced with local variables, declared in subroutine "newstate"; the local variables are used in the list of arguments of "carma_box". The names of the local variables are derived from original names by adding a digit to the end of the original name which indicates the number of dimensions of the variable (array) left after removing the spatial dimensions ("carma_box" is called once per a single grid box).

Table V-1. Variable names in the microphysical block carma_box and their correspondence to variable names in the original CARMA v. 2.2

Variable in carma_box	N of dims	Variable in original CARMA	N of dims
REAL			
(A)			
akelvin1	1	akelvin	2
akelvini1	1	akelvini	2
(C)			
ckernel4	4	ckernel	5
cmf2	2	cmf	2
coaglg2	2	coaglg	2
coagpe2	2	coagpe	2
conmax0	0	conmax	0
(D)			
diffmass4	4	diffmass	4
dm2	2	dm	2
dtime0	0	dtime	0
dtime_save0	0	dtime_save	0
d_gc1	1	d_gc	2

d_pc2	2	d_pc	3
d_ptc0	0	d_ptc	1
(E)			
evaplg2	2	evaplg	2
evappe2	2	evappe	2
(F)			
flux_sed_2	2	flux_sed	5
(G)			
gasprod1	1	gasprod	1
gcl_1	1	gcl	2
gc_1	1	gc3	2
gro1_2	2	gro1	3
gro2_1	1	gro2	2
growlg2	2	growlg	2
growpe2	2	growpe	2
gro_2	2	gro	3
gwtmol1	1	gwtmol	1
(P)			
pcl_2	2	pcl	3
pc_2	2	pc3	3
pconmax1	1	pconmax	2
period_nuc0	0	period_nuc	0
pkernel6	6	pkernel	7
ptcl_0	0	ptcl	1
ptc_0	0	ptc3	1
pvapi_1	1	pvapi3	2
pvapl_1	1	pvapl3	2
p_0	0	p3	1
(Q)			
qrad_2	2	qrad3	3
(R)			
radheat_0	0	radheat3	1
rhoa_0	0	rhoa3	1
rhoelem1	1	rhoelem	1
rhop_2	2	rhop	3
rlhe1	1	rlhe	2
rlheat0	0	rlheat	0
rlhm1	1	rlhm	2
rlh_nuc2	2	rlh_nuc	2
rmassmin1	1	rmassmin	1
rmassup2	2	rmassup	2
rmass2	2	rmass	2
rmrat1	1	rmrat	1
rnuclg3	3	rnuclg	3
rnucpe2	2	rnucpe	2
rup2	2	rup	2
(S)			

scrit2	2	scrit	3
solwtmol1	1	solwtmol	1
sol_ions1	1	sol_ions	1
supsatiold_1	1	supsatiold	2
supsati_1	1	supsati3	2
supsatlold_1	1	supsatlold	2
supsatl_1	1	supsatl3	2
(T)			
time0	0	time	0
time_nuc1	1	time_nuc	1
t_0	0	t3	1
(V)			
vf2	2	vf	3
volx5	5	volx	5
vol2	2	vol	2
(Z)			
zc3	1	zc3	1
INTEGER			
(I)			
icoagelem2	2	icoagelem	2
icoag2	2	icoag	2
icorelem2	2	icorelem	2
ienconc1	1	ienconc	1
ievp2elem1	1	ievp2elem	1
igelem1	1	igelem	1
iglow3	3	iglow	3
igrowgas1	1	igrowgas	1
igup3	3	igup	3
ilow3	3	ilow	3
imomelem1	1	imomelem	1
inucbin4	4	inucbin	4
inucelem2	2	inucelem	2
inucgas1	1	inucgas	1
inucmin1	1	inucmin	1
inucproc2	2	inucproc	2
inucstep1	1	inucstep	1
inuc2bin3	3	inuc2bin	3
inuc2elem2	2	inuc2elem	2
isolelem1	1	isolelem	1
itime0	0	itime	0
itype1	1	itype	1
iup3	3	iup	3
ixyz	0	ixyz	0
I_AERFREEZE0	0	I_AERFREEZE	0
I_COREMASS0	0	I_COREMASS	0
I_CORE2MOM0	0	I_CORE2MOM	0
I_DROPACT0	0	I_DROPACT	0
I_DROPFREEZE0	0	I_DROPFREEZE	0

I_ICEMELT0	0	I_ICEMELT	0
I_INVOLATILE0	0	I_INVOLATILE	0
I_VOLATILE0	0	I_VOLATILE	0
I_VOLCORE0	0	I_VOLCORE	0
(J)			
jglow3	3	jglow	3
jgup3	3	jgup	3
jlow3	3	jlow	3
jup3	3	jup	3
(M)			
maxsubsteps0	0	maxsubsteps	0
minsubsteps0	0	minsubsteps	0
(N)			
ncore1	1	ncore	1
nelemg1	1	nelemg	1
nnucbin3	3	nnucbin	3
nnucelem1	1	nnucelem	1
nnuc2elem1	1	nnuc2elem	1
npairl2	2	npairl	2
npairu2	2	npairu	2
ntsubsteps0	0	ntsubsteps	0
NBIN0	0	NBIN	0
NELEM0	0	NELEM	0
NGAS0	0	NGAS	0
NGROUP0	0	NGROUP	0
NSOLUTE0	0	NSOLUTE	0
LOGICAL			
(I)			
is_grp_ice1	1	is_grp_ice	1
if_nuc2	2	if_nuc	2
if_sec_mom1	1	if_sec_mom	1
(D)			
do_coag0	0	do_coag	0
do_grow0	0	do_grow	0
(T)			
totevap2	2	totevap	2
STRING			
(G)			
gasname1	1	gasname	1

APPENDIX VI

INPUT FOR THE IMPROVED VERSION OF CARMA

Table VI-1. Names of input variables in the improved version of CARMA and their functions

Variable name	Description of the variable	Sub-directory	File name	Special notes
Vertical grid-related information				
NZ	Number of vertical layers	\include\	aerad.h	NZ>2; for a box model, set NZ=1
dz	Thickness of a vertical layer (cm)	\aer\	initatm.f (g_cart_cart)	For uniform grids, set dzfix
Size distribution-related information				
NBIN	Number of bins	\include\	aerad.h	NBIN>4
rmin(1)	Minimum radius for CN (cm)	\aer\	setupaer.f	--
rmin(2)	Minimum radius for cloud droplets (cm)	\aer\	setupaer.f	--
rmrat(1)	Ratio of masses of CN between successive bins	\aer\	setupaer.f	rmrat(1) >1
rmrat(2)	Ratio of masses of cloud droplets between successive bins	\aer\	setupaer.f	rmrat(2) >1
gwtmol(1)	Molecular weight of the material of condensing gas, i.e. water vapor (g/mol)	\aer\	setupaer.f	--
solwtmol(1)	Molecular weight of the material of CN (g/mol)	\aer\	setupaer.f	--
rhoelem(1)	Density of the material of CN (g/cm^3)	\aer\	setupaer.f	rhoelem(1) = rhoelem(3)
rhoelem(2)	Density of the material of the cloud droplet's shell (g/cm^3)	\aer\	setupaer.f	--
rhoelem(3)	Density of the material of the cloud droplet's core (g/cm^3)	\aer\	setupaer.f	--
rhosol(1)	Density of the material of the cloud droplet's core (g/cm^3)	\aer\	setupaer.f	rhosol(1) = rhoelem(1) = rhoelem(3)
sol_ions(1)	Number of ions that cloud droplet's core material dissociates into	\aer\	setupaer.f	--

period_nuc	Nucleation update time interval (sec)	\aer\	setupaer.f	--
r0	Number mode radius for the initial CN size distribution (cm)	\aer\	initaer.f	--
rsig	Geometric standard deviation for the initial CN size distribution	\aer\	initaer.f	--
totn	Total number concentration for the initial CN size distribution ($1/cm^3$)	\aer\	initaer.f	--
rh_init	Initial relative humidity (%)	\aer\	initgas.f	w.r.t. liquid water
Initial vertical profiles-related information				
w	Initial vertical velocity at layer boundary (cm/sec)	\aer\	initatm.f (g_cart_cart)	--
t_sfc	Initial surface temperature (K)	\aer\	initatm.f (g_cart_cart)	--
dlapse	Initial temperature lapse rate (K/cm)			
p_surf	Initial surface pressure ($dyne/cm^2$)	\aer\	initatm.f (g_cart_cart)	--
rmu	Initial dynamic viscosity of dry air at layer mid-point (g/cm/sec)	\aer\	initatm.f (g_cart_cart)	Sutherland's equation
thcond	Thermal conductivity of dry air at layer mid-point (erg/cm/sec/K)	\aer\	initatm.f (g_cart_cart)	Pruppacher and Klett, Eq. 13-16.
dkz	Initial vertical diffusion coefficient at layer boundary (cm^2/sec)	\aer\	initatm.f (g_cart_cart)	--
Time varying vertical profile-related information				
w	Current vertical velocity at layer boundary (cm/sec)	\aer\	prestep.f	--
dkz	Current vertical diffusion coefficient at layer boundary (cm^2/sec)	\aer\	prestep.f	--
Control parameters				
ietime	Number of time steps	\aer\	carma_init.f	--
dtime	Time step size (sec)	\aer\	carma_init.f	--
minsubsteps	Minimum number of time sub-steps for fast microphysics	\aer\	carma_init.f	--
maxsubsteps	Maximum number of time sub-steps for fast microphysics	\aer\	carma_init.f	--

do_coag	Coagulation is on, if true	\aer\	carma_init.f	--
do_grow	Fast microphysics is on, if true	\aer\	carma_init.f	--
do_vtran	Vertical transport is on, if true	\aer\	carma_init.f	--
do_sed	Sedimentation is on, if true	\aer\	carma_init.f	--
Boundary conditions-related information				
itbnd_pc	Type of a boundary condition for particle concentrations on the top boundary	\aer\	carma_init.f	= I_FLUX_SPEC for specified flux = I_FIXED_CONC for specified concentration
ibbnd_pc	Type of a boundary condition for particle concentrations on the bottom boundary	\aer\	carma_init.f	
itbnd_gc	Type of a boundary condition for water vapor concentration on the top boundary	\aer\	carma_init.f	
ibbnd_gc	Type of a boundary condition for water vapor concentration on the bottom boundary	\aer\	carma_init.f	
itbnd_ptc	Type of a boundary condition for potential temperature concentration on the top boundary	\aer\	carma_init.f	
ibbnd_ptc	Type of a boundary condition for potential temperature concentration on the top boundary	\aer\	carma_init.f	
ftoppart	Downward particle flux across top boundary of model	\aer\	initaer.f	Used only if itbnd_pc = I_FLUX_SPEC
fbotpart	Upward particle flux across bottom boundary of model	\aer\	initaer.f	Used only if ibbnd_pc = I_FLUX_SPEC
ftopgas	Downward water vapor flux across top boundary of model	\aer\	initgas.f	Used only if itbnd_gc = I_FLUX_SPEC
fbotgas	Upward water vapor flux across bottom boundary of model	\aer\	initgas.f	Used only if ibbnd_gc = I_FLUX_SPEC
pc_topbnd	Particle concentration at center of top vertical layer ($1/cm^3$)	\aer\	initaer.f	Used only if itbnd_pc = I_FIXED_CONC
pc_botbnd	Particle concentration at center of bottom vertical layer ($1/cm^3$)	\aer\	initaer.f	Used only if ibbnd_pc = I_FIXED_CONC
gc_topbnd	Water vapor concentration at center of top vertical layer (g/cm^3)	\aer\	initgas.f	Used only if itbnd_gc = I_FIXED_CONC

gc_botbnd	Water vapor concentration at center of bottom vertical layer (g/cm^3)	\aer\	initgas.f	Used only if ibbnd_gc = I_FIXED_CONC
ptc_topbnd	Potential temperature concentration at center of top vertical layer ($K \cdot g/cm^3$)	\aer\	initatm.f	Used only if itbnd_ptc = I_FIXED_CONC
ptc_botbnd	Potential temperature concentration at center of bottom vertical layer ($K \cdot g/cm^3$)	\aer\	initatm.f	Used only if ibbnd_ptc = I_FIXED_CONC

APPENDIX VII

OUTPUT FILES FROM THE IMPROVED VERSION OF CARMA

For compiling the improved version of CARMA, cd to the subdirectory aer/ relative to the top level directory (cd aer), type “make” and hit the “Enter” key. If compilation is successful, the g77 compiler creates an executable file “carma.exe” in the subdirectory run/ relative to the top level directory. For running the model, cd to the subdirectory run/ (cd run), type “carma.exe” and hit “Enter”. The program creates several output files of both the ASCII type (editable by any text editor) and the binary type. Names of the output files along with their sizes and types are given in the table below. The first five files are files which are also created by the original CARMA v. 2.2. Instead of using the files for visualization purposes, we choose to output model data into our own files in the GrADS format (GrADS, which stands for Grid Analysis and Display System, is software for visualizing geophysical data). Thus, the last eight output files in the table are GrADS files. The .ctl files are descriptor (control) files which contain information regarding the binary .res files, such as names of variables, numbers of dimensions, time layers, etc.

Table VII-1. Names of output files, their sizes and types produced by the improved version of CARMA

Output file from the improved version of CARMA	Size of the output file (bytes)	Type of the output file (binary or ASCII)
Output files which are also created by the original CARMA v. 2.2		
carma_his.bin	4,961,926	Binary
carma.p	125,167	ASCII
carma_rad.out	0	ASCII
carma_res.out	0	ASCII
substep.out	0	ASCII
Additional output files in the GrADS format		
grads1.ctl	1,470	ASCII
grads1.res	8,664,000	Binary
grads23.ctl	1,980	ASCII
grads23.res	20,216,000	Binary
init_grads.ctl	1,626	ASCII
init_grads.res	2,600	Binary
phys_grads.ctl	2,491	ASCII
phys_grads.res	1,805,000	Binary

Grads binary files:

(1) grads1.res contains bin-dependent information for CN outputted on every time step.

(2) grads23.res contains bin-dependent information for cloud droplets outputted on every time step.

(3) init_grads.res contains bin-independent information for physical properties after initialization.

(4) phys_grads.res contains bin-independent information for physical properties outputted on every time step.

APPENDIX VIII

ADVECTION SUBROUTINES IN THE IMPROVED VERSION OF CARMA

File "driv_adv.f"

```
      subroutine driv_adv (
     &          NELEM, NGROUP, NBIN, NGAS
     &         , NX, NY, NZ, NXY, NXYZ, NZP1, NXYZP1
     &         , ixy, igridv, itbnd, ibbnd
     &         , itbnd_pc, ibbnd_pc, itbnd_gc, ibbnd_gc, itbnd_ptc, ibbnd_ptc
     &         , I_FIXED_CONC, I_FLUX_SPEC, I_CART, I_SIG
     &         , dtime, ftop, fbot, cvert_tbnd, cvert_bbnd, igelem, cvert
     &         , divcor,vtrans
     &         , pc_topbnd, pc_botbnd, gc_topbnd, gc_botbnd, ptc_topbnd, ptc_botbnd
     &         , w2, rhoa2, pc2, gc2, ptc2, ftopgas, fbotgas, ftoppart, fbotpart
     &         , dz2, zl2, zc2, dkz2, zmet2 )
c
c  @(#) driv_adv.f  Akberov July-2007
c  This routine drives vertical advection calculations.

      implicit none
c
c  Local declarations
c

c
c  Input/output parameters
c
      integer
     &   NELEM, NGROUP, NBIN, NGAS
     &  , NX, NY, NZ, NXY, NXYZ, NZP1, NXYZP1
     &  , ixy, igridv, itbnd, ibbnd
     &  , itbnd_pc, ibbnd_pc, itbnd_gc, ibbnd_gc, itbnd_ptc, ibbnd_ptc
     &  , I_FIXED_CONC, I_FLUX_SPEC, I_CART, I_SIG

      real  dtime, ftop, fbot, cvert_tbnd, cvert_bbnd

      integer igelem(NELEM)

      real
     &   cvert(NZ), divcor(NZ), vtrans(NZP1)
     &  , pc_topbnd(NXY,NBIN,NELEM), pc_botbnd(NXY,NBIN,NELEM)
     &  , gc_topbnd(NXY,NGAS), gc_botbnd(NXY,NGAS)
     &  , ptc_topbnd(NXY), ptc_botbnd(NXY)
     &  , w2(NXY,NZP1), rhoa2(NXY,NZ), pc2(NXY,NZ,NBIN,NELEM)
     &  , gc2(NXY,NZ,NGAS), ptc2(NXY,NZ)
     &  , ftopgas(NXY,NGAS), fbotgas(NXY,NGAS)
     &  , ftoppart(NXY,NBIN,NELEM), fbotpart(NXY,NBIN,NELEM)
     &  , dz2(NXY,NZ), zl2(NXY,NZP1)
     &  , zc2(NXY,NZ), dkz2(NXY,NZP1), zmet2(NXY,NZ)
c
```

```
c  Just local parameters for this subroutine
c
      integer k, ig, ielem, ibin, igas, nsubs, isubs

      real drho_dt_flux_spec(NZ), drho_dt_fixed_conc(NZ)

      real dtime_saved, Courant_max
c
c  Save current timestep
c
      dtime_saved = dtime

      ixy = 1
      do k = 1,NZP1
        vtrans(k) = w2(ixy,k)
      enddo

c  The output from subroutine "nsubsteps_adv" is
c  the number of substeps for advection "nsubs"
c  and the time step size for advection "dtime".
c  If the CFL condition can be satisfied with
c  the current time step size, then dtime doesn't change,
c  and the number of substeps will remain one.

      call nsubsteps_adv(ixy, NXY, NZ, dz2, dtime, vtrans, nsubs, Courant_max)

c=============================================
c  Loop over substeps
c=============================================
      do isubs = 1,nsubs

c=============================================
c  Loop over horizontal grid points
c=============================================
       do ixy = 1,NXY
c=============================================
c  Treat vertical transport of air density
c=============================================
c
c  First calculate change in density for the case when boundary
c  fluxes are zero
c
        itbnd = I_FLUX_SPEC
        ibbnd = I_FLUX_SPEC

        ftop = 0.
        fbot = 0.

        do k = 1,NZP1
          vtrans(k) = w2(ixy,k)
          if( k .le. NZ )then
            cvert(k) = rhoa2(ixy,k)
            divcor(k) = 0.
          endif
        enddo
```

```
        vtrans(1) = 0.
        vtrans(NZP1) = 0.
c
c   Calculate new air density due to vertical advection
c
        call vert_advection
     &                  (
     &        NZ, NXY, NXYZ, NZP1, NXYZP1
     &       , ixy
     &       , dtime
     &       , cvert
     &       , divcor
     &       , vtrans
     &       , dz2
     &                  )

        if( NXY .eq. 1 )then
c
c   In 1D, assume assume any vertical divergence is
c   balanced by horizontal convergence: don't allow air density
c   to change in time, but calculate rate of change that would have
c   resulted from advection -- this tendency is then used below for
c   a divergence correction.
c
          do k = 1,NZ
            drho_dt_flux_spec(k) = ( rhoa2(ixy,k) - cvert(k) ) /
     $                 ( rhoa2(ixy,k) * dtime )
          enddo

        else
c
c  Update air density when not in 1D.
c
          do k = 1,NZ
            rhoa2(ixy,k) = cvert(k)
          enddo

        endif    ! NXY .eq. 1

c=============================================
c  Treat vertical transport of particles
c=============================================
        itbnd = itbnd_pc
        ibbnd = ibbnd_pc

        do ielem = 1,NELEM          ! Loop over particle elements

          ig = igelem(ielem)        ! particle group

          do ibin = 1,NBIN          ! Loop over particle mass bins
c
c  Fluxes across top and bottom of model
c
            ftop = ftoppart(ixy,ibin,ielem)
            fbot = fbotpart(ixy,ibin,ielem)
```

```
c
c  Store temporary (work) values in <cvert>,
c  evaluate vertical velocities at layer boundaries, and
c  when 1D, assign divergence correction term
c
          do k = 1,NZP1
            vtrans(k) = w2(ixy,k)
          enddo

          if( ibbnd .eq. I_FLUX_SPEC ) vtrans(1) = 0.
          if( itbnd .eq. I_FLUX_SPEC ) vtrans(NZP1) = 0.

          do k = 1,NZ
            cvert(k) = pc2(ixy,k,ibin,ielem)
            if( NXY. eq. 1 ) divcor(k) = cvert(k) *
     $                                 drho_dt_flux_spec(k)
          enddo
c
c  Calculate particle concentrations due to vertical advection
c
          call vert_advection
     &                    (
     &           NZ, NXY, NXYZ, NZP1, NXYZP1
     &          , ixy
     &          , dtime
     &          , cvert
     &          , divcor
     &          , vtrans
     &          , dz2
     &                    )
c
c  Update particle concentrations.
c
          do k = 1,NZ
            pc2(ixy,k,ibin,ielem) = cvert(k)
          enddo

        enddo     ! ibin = 1,NBIN
      enddo       ! ielem = 1,NELEM
c
c=============================================
c  Treat vertical transport of gases
c  (for comments, look at the treatment of particles above).
c=============================================
      itbnd = itbnd_gc
      ibbnd = ibbnd_gc

      do k = 1,NZP1
        vtrans(k) = w2(ixy,k)
      enddo
      if( ibbnd .eq. I_FLUX_SPEC ) vtrans(1) = 0.
      if( itbnd .eq. I_FLUX_SPEC ) vtrans(NZP1) = 0.
```

```
        do igas = 1,NGAS

         ftop = ftopgas(ixy,igas)
         fbot = fbotgas(ixy,igas)

         do k = 1,NZ
          cvert(k) = gc2(ixy,k,igas)
          if( NXY. eq. 1 )
     &       divcor(k) = cvert(k) * drho_dt_flux_spec(k)
         enddo
c
c  Calculate water vapor concentration due to vertical advection
c
         call vert_advection
     &                  (
     &          NZ, NXY, NXYZ, NZP1, NXYZP1
     &         , ixy
     &         , dtime
     &         , cvert
     &         , divcor
     &         , vtrans
     &         , dz2
     &                  )

c
c  Update water vapor concentration.
c
         do k = 1,NZ
          gc2(ixy,k,igas) = cvert(k)
         enddo

        enddo      ! igas = 1,NGAS
c
c==========================================
c  Treat vertical transport of potential temperature concentration
c  (for comments, look at the treatment of particles above).
c==========================================
c
        itbnd = itbnd_ptc
        ibbnd = ibbnd_ptc

        ftop = 0.
        fbot = 0.

        do k = 1,NZP1
         vtrans(k) = w2(ixy,k)
        enddo

        if( ibbnd .eq. I_FLUX_SPEC ) vtrans(1) = 0.
        if( itbnd .eq. I_FLUX_SPEC ) vtrans(NZP1) = 0.

        do k = 1,NZ
         cvert(k) = ptc2(ixy,k)
         if( NXY. eq. 1 ) divcor(k) = cvert(k) * drho_dt_flux_spec(k)
        enddo
```

```
c
c  Calculate water vapor concentration due to vertical advection
c
        call vert_advection
     &                  (
     &        NZ, NXY, NXYZ, NZP1, NXYZP1
     &       , ixy
     &       , dtime
     &       , cvert
     &       , divcor
     &       , vtrans
     &       , dz2
     &                  )
c
c
c  Update potential temperature concentration.
c

        do k = 1,NZ
          ptc2(ixy,k) = cvert(k)
        enddo
c
c
      enddo   ! <ixy=1,NXY>
c=============================================
c  End of loop over horizontal grid points
c=============================================

     end do    ! isubs = 1,nsubs
c=============================================
c  End of loop over substeps
c=============================================

c
c  Restore normal timestep
c
     dtime = dtime_saved
c
c
c  Return to caller with new particle, gas, and potential temperature
c  concentrations (and air density in 2-d and 3-d cases).
c
     return
     end
```

```
c ------------------------------------------------------------------------
c       Subroutine for calculating the number of timesteps
c       and timestep size for advection
c ------------------------------------------------------------------------
      subroutine nsubsteps_adv(ixy, NXY, NZ, dz2, dtime, vtrans, nsubs
     &                                 , Courant_max)

      implicit none
      integer NXY,NZ,ixy,k,nsubs
      real vtrans(NZ+1),dz2(NXY,NZ)
      real dtime,dtime_old,dtime_new,veloc
      real Courant,Courant_max

      dtime_old = dtime

      Courant_max = 0.0

      do ixy=1, NXY
        do k=1, NZ

          veloc = max(abs(vtrans(k)),abs(vtrans(k+1)))
          if (veloc.lt.1.0e-10) veloc = 1.0e-10

          Courant = veloc*dtime/dz2(ixy,k)
          if (Courant .gt. Courant_max) Courant_max = Courant

          dtime_new = dz2(ixy,k)/veloc
          if (dtime_new.lt.dtime) then
            dtime = dtime_new
          end if

        end do    !   k=1, NZ
      end do      !   ixy=1, NXY

      if (dtime.lt.dtime_old) then
        nsubs = int(dtime_old/dtime) + 1
        dtime = dtime_old / nsubs
      else
        nsubs = 1
        dtime = dtime
      end if

      return
      end
```

File "vert_advection.f"

```
      subroutine vert_advection (
     &           NZ, NXY, NXYZ, NZP1, NXYZP1
     &          , ixy, dtime, cvert, divcor, vtrans, dz2 )
c
c  @(#) vert_advection.f  Akberov  July-2007.
c  This routine drives advection calculations.
c
      implicit none
c
c  Local declarations
c
c
c  Input/output parameters
c
      integer NZ,NXY,NXYZ,NZP1,NXYZP1,ixy

      real dtime

      real cvert(NZ), divcor(NZ), vtrans(NZP1), dz2(NXY,NZ)
c
c  Just local parameters for this subroutine
c
      integer k

      real drho_dt_flux_spec(NZ), drho_dt_fixed_conc(NZ)

      real cvert_new(NZ), F_above, F_below

      real sigma(0:NZP1),a,b,fminmod
c
c=============================================
c
c  Treat vertical advection
c
      sigma(0)=0.0  ! slopes for 2nd order REA with minmod limiter
      sigma(1)=0.0
      do k=2,NZ-1
        a = (cvert(k)-cvert(k-1))/(0.5*(dz2(ixy,k)+dz2(ixy,k-1)))
        b = (cvert(k+1)-cvert(k))/(0.5*(dz2(ixy,k)+dz2(ixy,k+1)))
        sigma(k) = fminmod (a, b)
      enddo
      sigma(NZ)=0.0
      sigma(NZ+1)=0.0
c
c
c  Calculate concentrations
c
c
      do k = 1,NZ

        if (vtrans(k) .gt. 0.0) then
          F_below = vtrans(k) * cvert(k-1)
     &     +0.5*vtrans(k)*(dz2(ixy,k)-vtrans(k)*dtime) * sigma(k-1)
        else if (vtrans(k) .lt. 0.0) then
```

```
        F_below = vtrans(k) * cvert(k)
     &     -0.5*vtrans(k)*(dz2(ixy,k)+vtrans(k)*dtime) * sigma(k)
       else
        F_below = 0.0
       end if  ! vtrans(k) .gt. 0.0

       if (vtrans(k+1) .gt. 0.0) then
        F_above = vtrans(k+1) * cvert(k)
     &     +0.5*vtrans(k+1)*(dz2(ixy,k)-vtrans(k+1)*dtime) * sigma(k)
       else if (vtrans(k+1) .lt. 0.0) then
        F_above = vtrans(k+1) * cvert(k+1)
     &     -0.5*vtrans(k+1)*(dz2(ixy,k)+vtrans(k+1)*dtime) * sigma(k+1)
       else
        F_above = 0.0
       end if  ! vtrans(k+1) .gt. 0.0

       cvert_new(k) = cvert(k) -
     &    (dtime/dz2(ixy,k)) * (F_above  - F_below)
     &   + divcor(k)*dtime

      end do  ! k = 1,NZ
c
c  Update concentrations
c
c
      do k = 1,NZ
       cvert(k) = cvert_new(k)
      enddo
c
c=============================================
c
c  Return to caller with new particle concentrations corrected to account
c  for vertical advection
c
      return
      end
c -----------------------------------------------------------
c             Function minmod
c -----------------------------------------------------------
c
      function fminmod (a, b)

      implicit none
      real fminmod, a, b

       if (abs(a) .lt. abs(b) .and. a*b .gt. 0.0d0) then
        fminmod = a
       else if (abs(b) .lt. abs(a) .and. a*b .gt. 0.0d0) then
            fminmod = b
          else if (a*b .le. 0.0d0) then
               fminmod = 0.0d0
       end if

      return
      end
```

REFERENCES

Ackerman A.S., Hobbs P.V. and Toon O.B., 1995. A model for particle microphysics, turbulent mixing, and radiative transfer in the stratocumulus-topped marine boundary layer and comparisons with measurements. *J. Atmos. Sci.,* 52, 1204–1236.

Barkstrom B. R., 1978. Some effects of 8-12 μm radiant energy transfer on the mass and heat budgets of cloud droplets. *J. Atmos. Sci.*, 35, 665-673.

Beard K.V. and Ochs H.T., 1984. Collection and coalescence efficiencies for accretion. *J. Geophys. Res.,* 89, 7165-7169.

Bougeault P., 1985. The diurnal cycle of the marine stratocumulus layer: A higher-order model study. *J. Atmos. Sci.*, 42, 2826-2843.

Brost R. and Wyngaard J.C., 1978. A model study of the stably stratified planetary boundary layer. *J. Atmos. Sci.*, 35, 1427-1440.

Colella P. and Woodward P.R., 1984. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.*, 54, 174-2001.

Covert D.S., 1988. North Pacific marine background aerosol: Average ammonium to sulfate molar ratio equals 1. *J. Geophys. Res.*, 93, 8455-8458.

Cunningham E., 1910. On the velocity of steady fall of spherical particles through a fluid medium. *Proc. Roy. Soc.,* 83, 357-365.

Duynkerke P.G. and Driedonks A.G.M., 1987. A model for the turbulent structure of the stratocumulus-topped atmospheric boundary layer. *J. Atmos. Sci.*, 44, 43-64.

Fuchs N.A., 1964. “The mechanics of aerosols”. Pergammon, 408 pp.

Hall W.D., 1980. Detailed microphysical model within a two-dimensional dynamic framework: model description and preliminary results. *J. Atmos. Sci.* 37, 2486-2507.

Hamill P., Toon O.B., Kiang C.S., 1977. Microphysical processes affecting stratospheric aerosol particles. *J. Atmos Sci.*, 34, 1104–1119.

Jaenicke R., 1993. “Tropospheric aerosols”, in Aerosol-Cloud-Climate Interactions, P. V. Hobbs ed., Academic, 1-31.

Junge C.E., Chagnon C.W., Manson J.E., 1961. Stratospheric aerosols. *J. Meteor.*, 18, 81–108.

Launder B.E. and Spalding D.B., 1972. “Mathematical models of turbulence”. Academic Press, London.

LeVeque R.J., 2002. “Finite volume methods for hyperbolic problems”, Cambridge University Press.

Miller A.M. and Slingo S., 2007. The ARM mobile facility and its first international deployment: measuring radiative flux divergence in West Africa. *BAMS*, 88, 1229-1244.

Millikan R.A., 1923. Coefficients of slip in gases and the law of reflection of molecules from the surfaces of solids and liquids. *Phys. Rev.*, 21, 217-238.

Nicholls S., 1984. The dynamics of stratocumulus: Aircraft observations and comparisons with a mixed layer model. *Quart. J. Roy. Meteor. Soc.,* 110, 783–820.

Nicholls S., and Leighton J., 1986. An observational study of the structure of stratiform cloud sheets: Part I. Structure. *Quart. J. Roy. Meteor. Soc.*, 112, 431-460.

Pruppacher H.R., Klett J.D., 1978. "Microphysics of clouds and precipitation". Reidel, 714 pp.

Schmidt G.A., Ruedy R., Hansen J.E., Aleinov I., Bell N., Bauer M., Bauer S., Cairns B., Canuto V., Cheng Y., Del Genio A., Faluvegi G., Friend A.D., Hall T.M., Hu Y., Kelley M., Kiang N.Y., Koch D., Lacis A.A., Lerner J., Lo K.K., Miller R.L., Nazarenko L., Oinas V., Perlwitz J., Perlwitz J., Rind D., Romanou A., Russell G.L., Sato M., Shindell D.T., Stone P.H., Sun S., Tausnev N., Thresher D., Yao M.-S., 2006. Present-day atmospheric simulations using GISS ModelE: Comparison to in situ, satellite, and reanalysis data. *Journal of Climate*, 19, 153-192.

Smithsonian Meteorological Tables. Smithsonian Institution Publication 4014, Washington, D.C., 1951.

Srivastava R.C., 1967. A study of the effect of precipitation on cumulus dynamics. *J. Atmos. Sci.,* 24, 36–45.

Timmreck C., 2001, Three-dimensional simulation of stratospheric background aerosol: First results of a multiannual general circulation model simulation. *J. Geophys. Res.*, 106, 28313-28332.

Toon O.B., Turco R.P., Westphal D., Malone R. and Liu M.S., 1988. A multidimensional model for aerosols: Description of computational analogs. *J. Atmos. Sci.,* 45, 2123-2143.

Tucker G.B. and Barry R. G., 1984. “Climate of the North Atlantic Ocean”, in Climates of the Oceans, H. van Loon ed., Elsevier, 716 pp.

Turco R.P., Hamill P., Toon O.B., Whitten R.C., Kiang, 1979. The NASA Ames Research Center stratospheric aerosol model. I. Physical processes and computational analogs, NASA Tech. Paper 1362.

Yanenko N. A., 1971. “The method of fractional steps”. Springer-Verlag, 160 pp.

Yum S.S. and Hudson J.G., 2001. Microphysical relationships in warm clouds. *Atmos. Res.*, 57, 81-104.

Zhang L., 2005. Numerical studies of the properties of low-level, warm stratiform clouds and precipitation and their interaction with aerosol particles and gaseous species. Ph.D. thesis, York University, Toronto, Canada.