International Hydrological Programme

Numerical Prediction of High-Impact Weather Systems

-The Textbook for Seventeenth IHP Training Course in 2007-

Kazuhisa Tsuboki and Atsushi Sakakibara

Hydorospheric Atmospheric Research Center, Nagoya University

United Nations Educational Scientific and Cultural Organization

Prepared for the Sventeenth IHP Training Course on Numerical Prediction of High-Impact Weather Systems, 2 - 15 December, 2007, Nagoya, Japan

Published in 2007 by the Hydrospheric Atmospheric Research Center (HyARC), Nagoya University, and United Nations Educational Scientific and Cultural Organization (UNESCO)

Printed by Nagoya University COOP
ISBN: 978-4-9980619-8-4
Cover: Beautiful clouds with precipitation over the Pacific Ocean near Palau. (Photo by K. Tsuboki)

Preface

The general aim of the 17th IHP Training Course is to help participants develop their basic knowledge of numerical prediction of high-impact weather systems, which is necessary for prediction and prevention/reduction of meteorological disasters. The course is part of UNESCO International Hydrological Programme (IHP) and is held every Japanese fiscal year that starts in April. The participants mainly from Asia-Pacific regions are given a series of lectures and practical trainings for about two weeks. Until 1996, the course was mainly focused on general hydrology and later concentrated on specified themes such as: snow hydrology (1998), remote sensing (1999), limnology (1999), hydrology related to head water management (2000), hydrogen and oxygen isotopes in hydrology (2002), precipitation and water resources (2003), effect of air-pollutants on the atmospheric environments and climate changes (2004), water and carbon cycles in terrestrial ecosystems (2005), and oceanography basics (2006).

This year, the Seventeenth IHP Training Course with a theme of "Numerical Prediction of High-Impact Weather Systems" will be held from 2 November to 15 December, 2007 at Nagoya and Tokyo with many lectures and practices. This IHP short course will focus on understanding basic knowledge of numerical prediction of high-impact weather systems. Participants will have practice to perform some simulation experiments of weather systems to understand how to use the cloud-resolving model, CReSS (cloud resolving storm simulator) developed in HyARC. High-resolution numerical simulation using the CReSS model will clarify a detailed structure of high-impact weather systems and make a quantitative prediction of the associated precipitation.

The Seventeenth course will be conducted by contributions from professionals/experts in universities and research institutes led by Prof. K. Tsuboki, Dr. T. Shinoda and Prof. T. Nagai, Nagoya University, Dr. T. Nasuno, Frontier Research Center for Global Change, Dr. T. Maesaka, National Research Institute for Earth Science and Disaster Prevention, Dr. M. Yoshizaki, Institute of Observational Research for Global Change, Dr. K. Saito, Meteorological Research Institute, Prof. T. Aoki, Tokyo Institute of Technology, Prof. K. Tanaka, Kyoto University, Dr. W. Ofuchi, The Earth Simulator Center, and Mr. A. Sakakibara, Chuden CTI Co.,Ltd.

The course would not have been possible to conduct without management supports. Dr. Giuseppe Arduino and his staff in UNESCO Jakarta Office worked for the arrangement for the participants. The IHP Training Course is under continuous support from the Ministry of Education, Culture, Sports, Science, and Technology.

November 28, 2007

Hiroshi Uyeda
Chairperson of
Working Group for IHP Training Course
Director,
Hydrospheric Atmospheric Research Center,
Nagoya University

Acknowledgments

I would like to express my sincere thanks to Prof. Hiroshi Uyeda, the chairperson of the working group of the IHP Training Course and Director of the Hydrospheric Atmospheric Research Center (HyARC), Nagoya University, Japan.

The support from the Japan National Commission for UNESCO, sub-committee of IHP led by Prof. Kuniyoshi Takeuchi from Yamanashi University, and the working group of IHP Training Course is highly acknowledged. Thanks are also extended to Mr. Kazuo Akiyama and Ms. Eriko Himata from the Ministry of Education, Culture, Sports, Science and Technology of Japan, Mr. Akihiro Yamamoto from Nagoya University, Japan for sharing the role of secretariat, and the staffs (Dr. Taro Shinoda, Ms. Machiko Toida, Ms. Saori Haga and Ms. Mariko Kayaba) of HyARC, Nagoya University, Japan for their local arrangements and technical editing of this textbook.

November 22, 2007

Kazuhisa Tsuboki
Hydrospheric Atmospheric
Research Center (HyARC),
Nagoya University

Editor

Contents

Acknowledgments

The development of the CReSS model was supported by the Special Coordination Funds for Promoting Science and Technology, "Development of parallel softwares for large-scale simulation to predict the global climate change with high resolution" led by Professor A. Sumi at the University of Tokyo, and by the Research Organization for Information Science and Technology (RIST) for the period from 1998 to 2003. The authors would like to thank Professor A. Sumi, Professor N. Kamiya at Aichi-Gakusen University and Mr. H. Nakamura of the RIST for their indispensable contributions to the model development.

The Hydrospheric Atmospheric Research Center (HyARC) at Nagoya University has supported the development of the CReSS model as a research program since 2003. We used the mainframe computers and supercomputers at the University of Tokyo and Nagoya University for the development and calculations of the CReSS model. We also used the Earth Simulator at the Earth Simulator Center for simulations and calculations of the CReSS model. Part of the textbook was written during K. Tsuboki's visit to the International Pacific Research Center (IPRC) at the University of Hawaii. The IPRC is partly funded by the JAMSTEC in Japan. Tsuboki would like to thank Dr. H. Kubota at the JAMSTEC and Professor Shang-Ping Xie at the University of Hawaii for this opportunity of visit.

Notifications

Preface

At the end of the twentieth century, massive computers became reality, with theoretical performance reaching Tera-FLOPS, main memory of Gigabytes, and storage of Terabytes. This led us to believe that we could perform super-massive calculations at the beginning of the twenty-first century. What should we do with such a computing capability that is available now? Meteorologists have longed for faster and more massive computers for their numerical models. A cloud-resolving model is one of such models, which includes complicated interactions among water vapor, cloud particles, and precipitation.

Research on cloud and precipitation using a cloud-resolving model is still at its developing stage. There remain many unsolved problems, such as parameterizations of various physical processes, techniques for efficient computations, minimization of computational errors, etc. There are also a lot of atmospheric phenomena to be discovered. Cloud-resolving models will help us to understand various phenomena related to cloud and precipitation systems.

What is needed for the development of cloud-resolving models? Researches with different viewpoints are needed for the development of cloud-resolving models. A large number of numerical experiments using these cloud-resolving models are also needed. We can advance our knowledge through trials and errors using these models. Only then, cloud-resolving models will be refined.

As numerical models become more sophisticated, it is inefficient, even impossible, for individual researcher to develop his own cloud-resolving model from a scratch. Most scientists are more interested in using a numerical model as a tool for their own research, which may involve computational methods, physical process studies, simulations of observed phenomena, weather forecasts for disaster prevention, and so on. Out of such a consideration, we have been developing a cloud-resolving model, which was supported by Special Coordination Funds for Promoting Science and Technology (SCF), "Study on Advanced Simulations of Downburst Phenomenon" (Principal Investigator: Professor Nobuhiko Kamiya). This cloud-resolving model could serve as a basic platform for these scientists.

Recently, some models have included non-hydrostatic and compressible equations for their cloud physical processes, such as the Advanced Regional Prediction System (ARPS) developed by the University of Oklahoma and the Non-Hydrostatic Model (NHM) developed by the Meteorological Research Institute of the Japan Meteorological Agency. We would like to acknowledge the useful information provided by these models. The development of our own cloud-resolving model is heavily based on many previous studies. We would like to express our appreciations for these authors.

Computers are been improved rapidly, especially their hard-wears. A few years ago, only a single-vector processor was used for massive calculations. Now, parallel computers with multiple processors are the main player for the computations; they are necessary for the amount of calculations involved in a cloud-resolving model. Our cloud-resolving model is designed to perform on parallel computers with high efficiency. It includes many physical processes with a high resolution, while its computation domain is smaller than those of global climate models. Parallel computations are necessary for the huge calculations using the cloud-resolving model.

We name our cloud-resolving model the Cloud Resolving Storm Simulator (CReSS), which can simulate clouds and precipitation systems. CReSS has high resolutions in both horizontal and vertical directions, and includes detailed physical processes of clouds and precipitation. Processes of clouds and precipitation, however, still need improvement. Both observational studies and model development are needed for further understanding of clouds and precipitation systems.

The goal of the development of CReSS is to predict heavy rainfalls based on this understanding.

CReSS is now open to the public. This textbook describes the usage of CReSS, the basics of model, and the program codes. We hope that this document will be useful for all CReSS users. You can help us to improve the document by contacting us at tsuboki@rain.hyarc.nagoya-u.ac.jp. Your comments will be appreciated.

Part I

Simulations of High-Impact Weather Systems

Chapter 1

Introduction

In this chapter, we present the objectives, history, and summary of the CReSS (Cloud Resolving Storm Simulator) model. CReSS is a non-hydrostatic and compressible model to simulate cloud and precipitation systems. These systems are highly complex, and their development and evolution processes are determined by non-linear interactions between the atmosphere and hydrometeors. CReSS is designed for a parallel computer that enables to perform large computations, such as this involving cloud and precipitation systems. Advantages of CReSS are as follows.

- Two versions of CReSS, the parallel version and the single version are available for users. The former is designed for parallel computations, and the latter for a single CPU. The computation output will not be dependent on the versions or on domain decompositions.
- Detailed cloud micro-physical processes are included for moist simulations. On the other hand, CReSS can perform dry simulations without condensation processes.
- CReSS can perform a simulation of a real weather system with a 3-dimensional inhomogeneous initial and boundary conditions, as well as a numerical experiment with idealized initial and boundary conditions.
- All codes of CReSS are written using the standard FORTRAN 90. They are easy to read and easy to modify, after a brief initial instruction. CReSS can be performed on most computers ranging from mainframe computers to PC Unix machines without any modifications of the codes.

1.1 Objectives

One of the most important objectives of regional models is a high-resolution simulation of high-impact weather systems, which can be used for detailed studies and accurate predictions. High-impact weather systems are most significant phenomena in the atmosphere and sometimes cause huge disasters to human society. Understanding their mechanisms and structures is necessary for prediction and prevention/reduction of disasters. Most high-impact weather systems that cause heavy rainfalls and/or violent winds consist of cumulonimbus clouds and their organized systems. They are usually embedded within a larger weather system and occasionally have a multi-scale structure, ranging from cloud-scale to synoptic-scale systems. These weather systems in East Asia include the Baiu front, typhoons, and winter snowstorms associated with a cold-air outbreak.

In order to perform simulations and numerical experiments of high-impact weather systems, we have been developing the cloud-resolving numerical model named CReSS. Since the multi-scale structure of the weather systems has a wide range of horizontal scales, a large computational domain and a very high-resolution grid to resolve individual classes of the multi-scale structure are necessary to simulate the evolution of the weather systems. In particular, an explicit calculation of cumulonimbus clouds is important for a quantitative simulation of precipitation associated with the high-impact weather. It is also required to formulate accurately cloud physical processes as well as the fluid dynamic and thermodynamic processes. For this type of computation, a large parallel computer with a huge memory is necessary.

The purpose of the present-day research is to explicitly simulate clouds and their organized systems in a large domain (larger than 1000 x 1000 km) with resolved individual clouds using a very fine grid system (less than 1 km horizontally). This will clarify a detailed structure of the high-impact weather systems and permits a more quantitative prediction of the associated precipitation. It will contribute to the accurate prediction of precipitation and the reduction of disasters caused by the high-impact weather systems.

1.2 Brief description of CReSS

The formulation of CReSS is based on the non-hydrostatic and compressible equation using terrain-following coordinates. Prognostic variables are 3-dimensional velocity components, perturbations of pressure and potential temperature, water vapor mixing ratio, sub-grid scale turbulent kinetic energy (TKE), and cloud physical variables. A finite difference method is used for the spatial discretization. The horizontal domain is rectangular, and variables are set on a staggered grid: the Arakawa-C grid in the horizontal and the Lorenz grid in the vertical. For time integration, the mode-splitting technique (Klemp and Wilhelmson 1978) is used. Terms related to sound waves of the basic equation are integrated with a small time step, and other terms with a large time step.

Cloud physical processes are formulated by a bulk method of cold rain, which is based on Lin et al. (1983), Cotton et al. (1986), Murakami (1990), Ikawa and Saito (1991), and Murakami et al. (1994). The bulk parameterization of cold rain considers water vapor, rain, cloud, ice, snow, and graupel. The microphysical processes implemented in the model are described in Fig.1.1.

Parameterizations of the sub-grid scale eddy motions in CReSS are one-order closure of Smagorin-

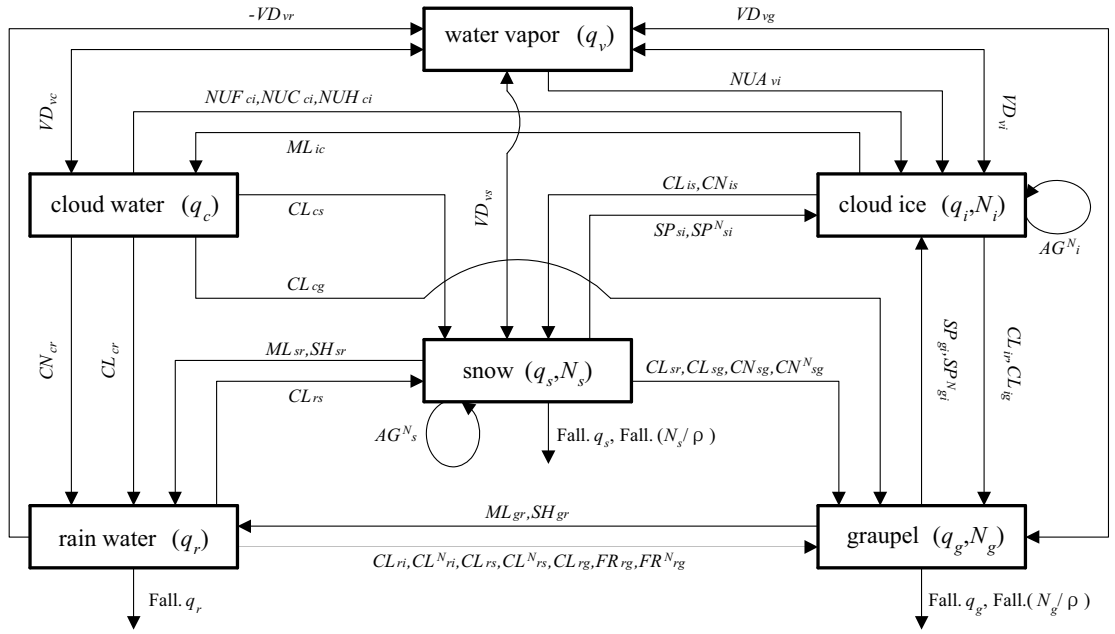

Figure 1.1. Diagram describing water substances and cloud microphysical processes in the bulk scheme of CReSS.

sky (1963) or the 1.5-order closure of turbulent kinetic energy (TKE). In the latter parameterization, the prognostic equation of TKE is used. All numerical experiments in this textbook use three-dimensional 1.5-order closure scheme. The surface process of CReSS is formulated by a bulk method, whose bulk coefficients are taken from Louis et al. (1981).

Several types of initial and boundary conditions are available. For a numerical experiment, a horizontally uniform initial field provided by a sounding profile will be used with an initial disturbance of a thermal bubble or random temperature perturbation. The boundary conditions are one of the following types; rigid wall, periodic, zero normal-gradient, and wave-radiation types.

CReSS can be nested within a coarse-grid model for a prediction experiment. In such an experiment, the initial field is provided by interpolating grid-point values and the boundary condition is provided by the coarse-grid model. For a computation within a large domain, conformal map projections are available, which include the Lambert conformal projection, the polar stereographic projection, and the Mercator projection.

For parallel computing of a large computation, CReSS provides two-dimensional domain decomposition in the horizontal direction (Fig.1.2). Parallel processing is performed using the Massage Passing Interface (MPI). Communications between individual processing elements (PEs) are performed by exchanging data of the two outermost grids. The OpenMP is also available.

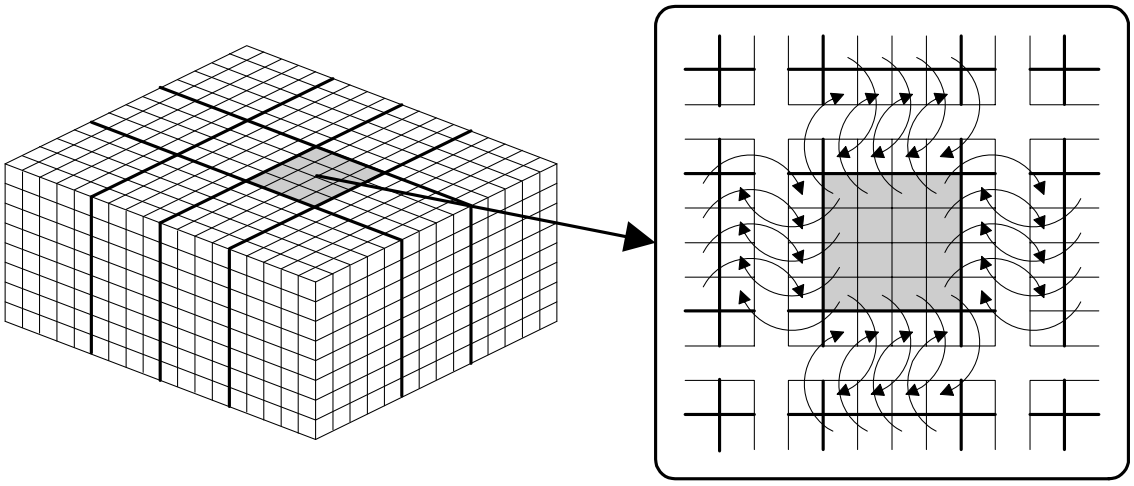

Figure 1.2. Schematic representation of a two-dimensional domain decomposition and the communication strategy for parallel computations using MPI.

1.3 History of CReSS development

CReSS was developed by Kazuhisa Tsuboki at the Hydrospheric Atmospheric Research Center (HyARC) of the University of Nagoya, and by Atsushi Sakakibara at Research Organization for Information Science and Technology, whose current affiliation is Chuden CTI Co.,Ltd.. The development of CReSS was performed as a part of the project named "Study for Advanced Simulation of Downburst Phenomena" (Principal Investigator: Professor N. Kamiya at Aichi-Gakusen University). The project was supported by Special Coordination Funds for Promoting Science and Technology and was one of a program to develop parallel software for large-scale simulations to be used for predictions of the global climate change led by Professor A. Sumi of University of Tokyo for the period from 1998 to 2003. At the end of the project, the version 1.4 of the CReSS model was accomplished.

Ever since 2003,HyARC at Nagoya University has been supporting the development of the CReSS model. It was optimized for the Earth Simulator in 2003. All codes were rewritten in FORTRAN 90 instead of FORTRAN 77 and the version 2.0 was accomplished in 2003. We evaluated the performance of CReSS on the Earth Simulator in 2003. The result is summarized in Table 1. The parallel operation ratio was measured using 128 nodes (1024 CPUs) and 64 nodes (512 CPUs) of the Earth Simulator. The vector and parallel operation ratios are sufficiently high to perform a large computation on the Earth Simulator.

Using the CReSS model, HyARC carried out a research program entitled "High Resolution Modeling of Atmospheric Water Circulation Systems Using the Cloud-Resolving Model" for the period from 2003 to 2006. The purpose of this research program was to explicitly simulate clouds and organized precipitation systems in a large domain and very fine grid (less than 1 km). Using the Earth Simulator, we performed simulation experiments of important cloud and precipitation systems.

One of the experiments is the simulation of the Niigata-Fukushima heavy rainfall event, which occurred in July 2004. It shows that an intense rainband forms and is maintained within the subsynoptic-scale low along the Baiu front. Other experiments include simulations of snowstorms over the Sea of Japan as well as those over the Great Lakes region of Canada. Snowstorm

experiments show various types of precipitation systems composed of intense convective clouds in cold air streams. One of the important objectives of this research program is to simulate typhoons, whose developments are attributed to close interaction between a large-scale motion and cumulonimbus convections. Simulations of Typhoons 0418, 0423, 0613 were performed with a horizontal resolution of 500-1,000 m. They show very realistic features of typhoon and individual clouds. All of these experiments show very detailed structures of individual precipitation clouds as well as overall structures of the large-scale systems.

The simulation experiments using the CReSS model range from typhoon scale to tatsumaki scale, and resolve every class of the multi-scale structure. Such super-high resolution calculations provide detailed information of weather systems. Using the output of these experiments, we will be able to clarify not only structures of weather systems but also interactions between different classes of scales. High-resolution simulations will contribute to an accurate and quantitative prediction, as well as to prevention/reduction of disaster caused by high-impact weather systems.

As a result of the research program at the HyARC, CReSS was updated to version 2.2 in 2006. This version is now fixed and open to the public. The development of version 3.0 is on-going, with substantial improvements on the way.

Chapter 2

Simulations and Numerical Experiments

The CReSS model has been used for various purposes. Calculations are roughly categorized into simulations and numerical experiments. The simulations are referred to those calculations that simulate a real weather system or real clouds. In these cases, initial and boundary conditions are provide by a coarse-mesh model or 3-dimensional grid-point data; artificial initial perturbations are generally not necessary. A real topography and surface conditions are also used. The numerical experiments are referred to those using an idealized condition of the atmosphere. The initial conditions for these experiments are provided by a single sounding or by a theoretical function. The initial state is usually horizontally uniform, and artificial perturbations are necessary if the land surface is flat and unvarying. The numerical experiments are further categorized into dry and moist types in 2- or 3-dimensions. The dry experiment does not consider condensation, while the moist one includes condensation in both cloud and precipitation processes.

In this chapter, we give some examples of simulations and numerical experiments under different conditions. We recommend users to start with a 2-dimensional dry experiment, such as cats-eye and mountain waves. They are relatively simple calculations and can be easily performed even on a laptop computer. Three-dimensional experiments are computationally heavier, and a large computer is necessary. Some simulations of real weather systems such as typhoons and storms are also presented in this chapter. Readers will find objectives and performance of the CReSS model at the end of the chapter.

2.1 Mountain waves: dry experiment in 2-dimensional geometry

When air is forced to flow over a mountain, individual air parcels are displaced vertically. If the atmosphere is stably stratified, the parcels descend and may oscillate about their equilibrium levels. The gravity waves that result, referred as *mountain waves*, have been observed in mountainous regions all over the world.

The presence of mountain waves is frequently revealed by distinctive orographic clouds. Various types of mountain waves produce different types of wave clouds. An identification of a particular type of wave cloud can be used to make qualitative deductions about the vertical variation in wind and stability over the mountains.

Large-amplitude mountain waves can produce several weather phenomena that significantly affect human activity and therefore require the attention of weather forecasters. The strong downslope winds observed along the lee of mountain barriers are usually associated with large amplitude waves. Dangerous regions of clear-air turbulence (CAT) are also produced by these waves.

Numerical experiments of mountain waves are performed using CReSS. Examples are shown as follows. The basic conditions of the experiments are

- Calculation is in 2-dimensional.
- A bell-shaped mountain is located at the center of the calculation domain.
- No moist processes are included.
- Horizontal wind and buoyancy frequency of the basic field are uniform in both horizontal and vertical directions.
- The vertical grid spacing is uniform.
- The lateral boundary condition is cyclic.
- The bottom boundary condition is slip.

Scorer parameter, which is a ratio of the buoyancy frequency to horizontal wind speed in this case and has the dimension of wave number, determines characteristics of the mountain waves theoretically. According to the Scorer parameter and the horizontal scale of bell-shaped mountain, different types of mountain waves are excited.

One example is mountain waves propagating to the downstream. The experimental design is summarized in Table 2.1.

Grid number	$2003 \times 4 \times 253$
Grid spacing	$50 \times 50 \times 50$ m
Mountain	Bell-shaped and isolated, height: 500 m, half-width: 500 m
Initial wind	u=10 m s^{-1}
Buoyancy frequency	0.01 s^{-1} and constant with height
Initial disturbance	none

Table 2.1: **Experimental design of mountain waves propagating to the downstream.**

In the experiment, the averaged gradient of the mountain is 26 degrees and the maximum is 34 degrees. Although the mountain slope is rather steep, mountain waves are successfully simulated (Fig. 2.1). The mountain waves propagate upward in the downstream direction, just as the linear theory of mountain waves predicts.

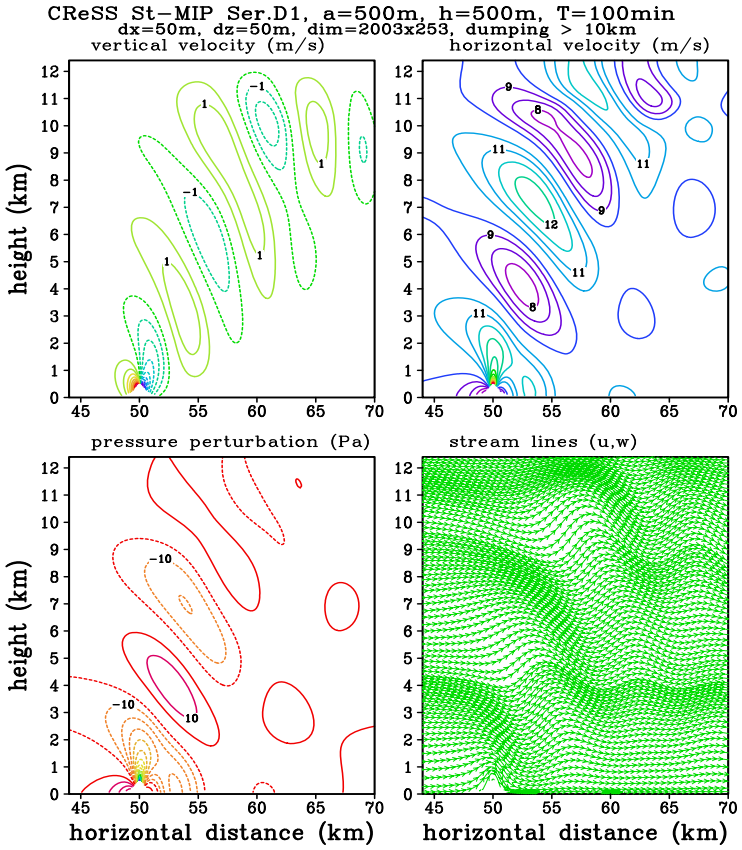

Figure 2.1. Numerical experiment of mountain waves. Velocity (u, w)[m s^{-1}], pressure perturbation[Pa] at 6000 seconds from the initial time.

The other example is large amplitude and non-steady waves. The experimental design is shown in Table 2.2.

Grid number	$2003 \times 4 \times 253$
Grid spacing	$50 \times 50 \times 50$ m
Mountain	Bell-shaped and isolated, height: 1000 m, half-width: 500 m
Initial wind	u=10 m s^{-1}
Buoyancy frequency	0.02 s^{-1} and constant with height
Initial disturbance	none

Table 2.2: **Experimental design of large amplitude and non-steady waves.**

The result shows that the non-steady waves are propagating on the downstream side of the mountain (Fig. 2.2). A very strong wind is present near the surface on the lee side of the mountain and a negative perturbation velocity occurs above the strong surface wind. This is a sort of the down-slope wind.

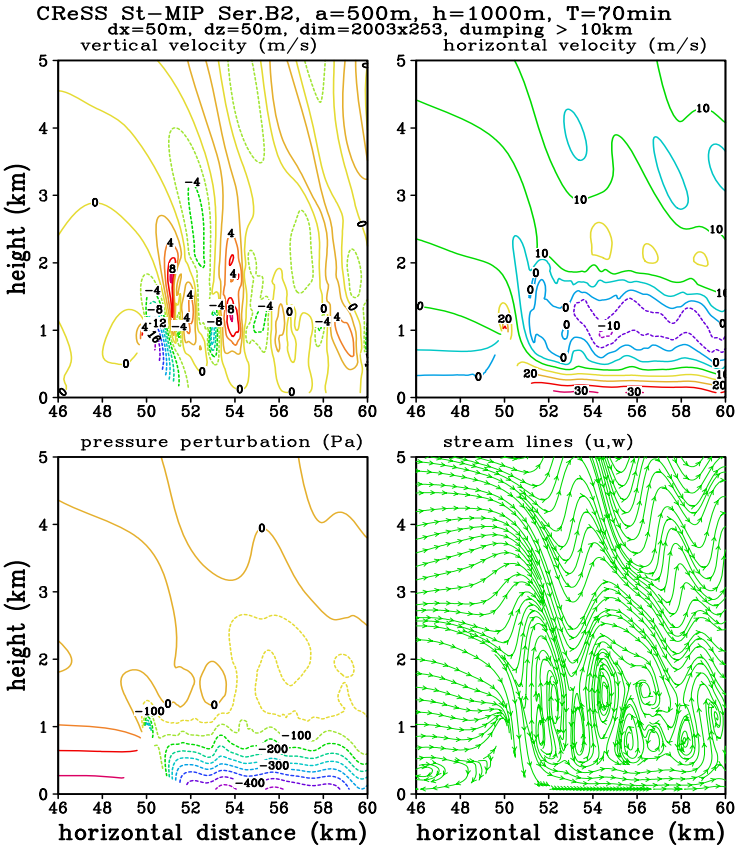

Figure 2.2. Numerical experiment of non-steady mountain waves. Velocity (u, w)[m s^{-1}], pressure perturbation[Pa] at 4200 seconds from the initial time.

2.2 Kelvin-Helmholtz billows: dry experiment in two-dimensional geometry

The numerical experiment of Kelvin-Helmholtz billows is performed in a two-dimensional geometry with a grid size of 20 m. The profile of the basic flow is the hyperbolic tangent type. The setting of the experiment is summarized in Table 2.3.

Grid number	$51 \times 4 \times 71$
Grid spacing	$20 \times 20 \times 10$ m
Time step	HE-VE, large: 0.8 s, small: 0.04 s
Topography	none
Basic wind	A shear wind with tanh type

Basic temperature	tanh type
Initial perturbation	A sine-shaped disturbance with maximum amplitude of 0.4 K

Table 2.3: **Experimental design of Kelvin-Helmholtz billows.**

Streamlines of u and w components show a clear cats eye structure of the Kelvin-Helmholtz billows (Fig.2.3). This result is very similar to that of Klaassen and Peltier (1985). The model also simulates the overturning of potential temperature associated with the billows (Fig. 2.4). This example shows that the model works well with a grid size of a few tens of meters in the dry experiment.

2.3 Simulation of Tornado within a Supercell: moist experiment in three-dimensional geometory

For a simulation experiment of a moist atmosphere, we choose a tornado-producing supercell observed on 24 September 1999 in the Tokai District of Japan. The experiment is aimed at resolving the vortex of the tornado within the supercell.

The supercell is highly three-dimensional and its horizontal scale is several tens of kilometers. A large domain in the order of 100 km is necessary to simulate a supercell using a cloud model. A tornado, on the other hand, has a horizontal scale of a few hundred meters. The simulation of the tornado requires a fine horizontal grid spacing in the order of 100 m or less. In order to simulate a supercell and its associated tornado by a cloud model, a huge memory and high-speed CPU are crucial.

Numerical experiments of a supercell thunderstorm using a cloud model have been performed during the past 20 years (Wilhelmson and Klemp, 1978; Weisman and Klemp, 1982, 1984). Recently, Klemp and Rotunno (1983) attempted to increase the horizontal resolution to simulate the fine structure of a meso-scale cyclone within the supercell. An intense tornado occasionally occurs within a supercell thunderstorm. It is, however, more difficult to resolve a tornado. To overcome this difficulty, Wicker and Wilhelmson (1995) used an adaptive grid method, a two-way nesting technique, to simulate tornado genesis. Their grid spacing of the fine mesh was 120 m. Grasso and Cotton (1995) also used a two-way nesting procedure of a cloud model to simulate a tornadic vorticity. Nesting methods introduce complications of communication between the coarse-grid model and the fine-mesh model at the boundary. A cloud-resolving model does not require any nesting methods. We simulate both the supercell and the tornado using a uniform grid, therefore avoid any complication of the boundary communication. The computational domain of the present simulation is about 50 $\times$ 50 km and the grid spacing is 100 m. The integration time is about 2 hours.

The basic field is give by a sounding at Shionomisaki, Japan at 00 UTC, 24 September 1999 (Fig.2.5). The initial perturbation is a warm thermal bubble placed near the surface, which can induce an initial convective cloud.

One hour from the initial time, a quasi-stationary supercell is simulated by CReSS (Fig.2.6). The hook-shaped precipitation area and the bounded weak echo region (BWER), which are characteristic features of a supercell, are formed in the simulation. An intense updraft occurred along the surface flanking line. At the central part of BWER or of the updraft, a tornadic vortex

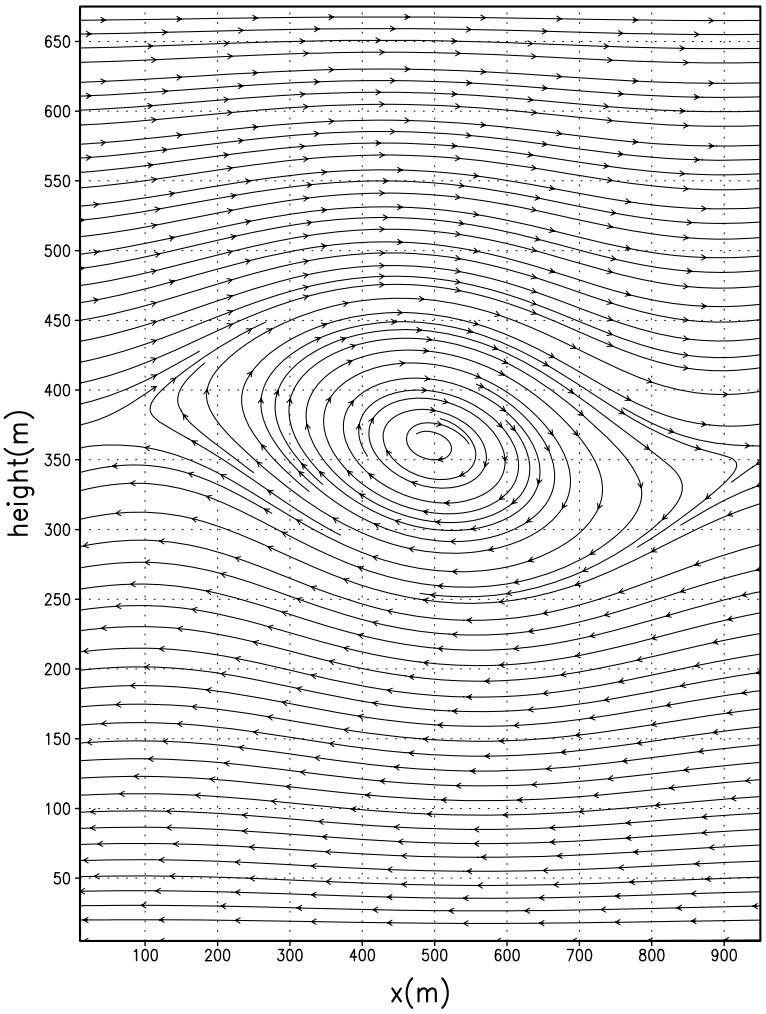

Figure 2.3. Stream lines of the Kelvin-Helmholtz billow at 240 seconds from the initial simulated in the two-dimensional geometry.

was formed 90 minutes from the initial time.

A close view shows closed contours in the central part of the vorticity (Fig.2.7). The diameter of the vortex is about 500 m and the maximum vorticity is about 0.1 s^{-1}, similar to the observed tornado. The pressure perturbation also shows closed contours that correspond to those of the vorticity (Fig.2.8). This indicates that the flow in the vortex is in cyclostrophic balance. The vertical cross section of the vortex shows that the axis of the vorticity and the associated pressure perturbation is inclined to the left hand side and extends to a height of 2 km (Fig.2.9). At the center of the vortex, the downward extension of cloud is simulated.

Though this is a preliminary result of a supercell and tornado, some characteristic features of the observation are successfully simulated. The important point of this simulation is that both the supercell and the tornado are simulated with the same grid size. The tornado is produced purely by the physical processes in the model. A detailed analysis of the simulation will provide us important information on tornado genesis within a supercell.

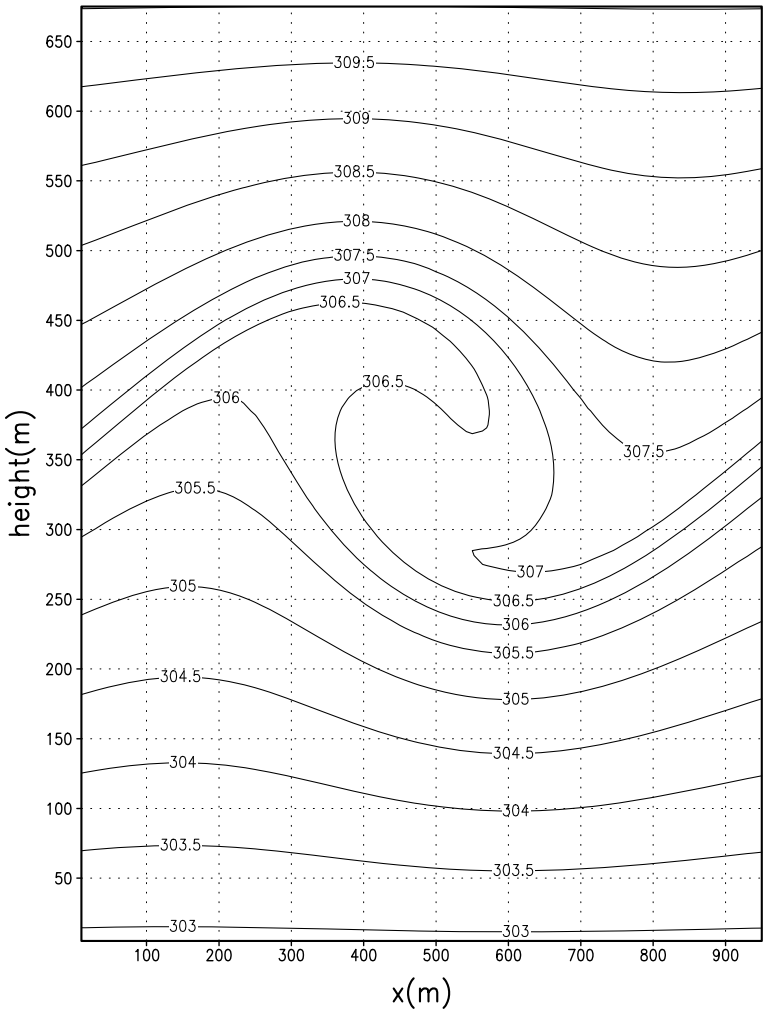

Figure 2.4. Same as Fig.2.3, but for potential temperature.

2.4 Simulation of a squall line: moist experiment in three-dimensional geometory

A squall line is a significant mesoscale convective system. It is usually composed of an intense convective leading edge and a trailing stratiform region. An intense squall line was observed by three Doppler radars on 16 July 1998 during an intensive field observation of GAME/HUBEX (the GEWEX Asian Monsoon Experiment/Huaihe River Basin Experiment) in China. The squall line extended from the northwest to the southeast with a width of a few tens of kilometers and moved northeastward at a speed of 11 m s^{-1}. Radar observation showed that the squall line consisted of intense convective cells along its leading edge. Some of the cells reached a height of 17 km. The rear-inflow was present at a height of 4 km, which descended to cause the intense lower-level convergence at the leading edge. After the squall line passed over the radar sites, a stratiform precipitation was extending behind the convective leading edge.

The design of the simulation experiment using CReSS is as follows. Both the horizontal and vertical grid sizes are 300 m within a domain of 170 km $\times$ 120 km. Cloud microphysics is the cold rain type. The boundary condition is the wave-radiating type. The initial condition is provided by a dual Doppler analysis and sounding data. The inhomogeneous velocity field within the

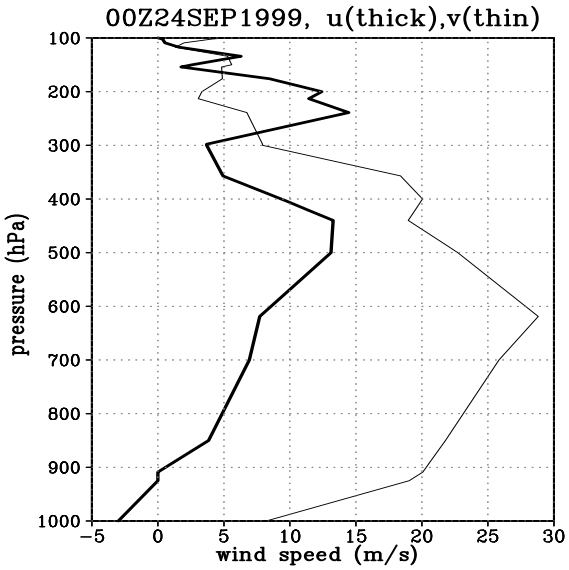

Figure 2.5. Vertical profiles of zonal component (thick line) and meridional component (thin line) observed at Shionomisaki at 00 UTC, 24 September 1999.

storm is determined by the dual Doppler radar analysis directly, while that outside the storm and thermodynamic field are provided by the sounding observation. Mixing ratios of rain, snow and graupel are estimated from the radar reflectivity, while mixing ratios of cloud and ice are set to be zero at the initial time. A horizontal cross section of the initial field is shown in Fig.2.10.

The simulated squall line extending from the northwest to the southeast moved northeastward (Fig.2.11). The convective leading edge of the simulated squall line is maintained by the replacement of new convective cells and moved to the northeast, similar to the behavior of the observed squall line. Convective cells reached a height of about 14 km with large production of graupel above the melting layer. The rear inflow is as significant as that in the observation. A stratiform region extends with time behind the leading edge. Cloud extends to the southwest to form a cloud cluster. These results show that CReSS can successfully simulate the development and movement of a squall line.

2.5 Typhoons and the associated heavy rainfall: T0418 and T0423

Typhoons develop by close interaction between a large-scale disturbance and embedded intense cumulonimbus clouds. The horizontal scale of typhoons ranges from several 100 km to a few 1000 km while that of the cumulonimbus clouds is an order of 10 km. Typhoons often bring a heavy rain and a strong wind. The heavy rain is usually localized in the eyewall and spiral rainbands which develop within typhoons. Since cumulonimbus clouds are essentially important for typhoon development, the cloud-resolving model is necessary for a detailed numerical simulation of typhoons.

Some typhoons usually attain Japan and its surroundings and cause severe disasters. In particular, ten typhoons landed over the main lands of Japan in 2004. In the present section, we show two simulation experiments of the typhoons. One is the typhoon T0418 which brought a very intense wind and caused huge disasters due to the strong wind. The other is the typhoon T0423 which brought heavy rainfalls and caused severe floods.

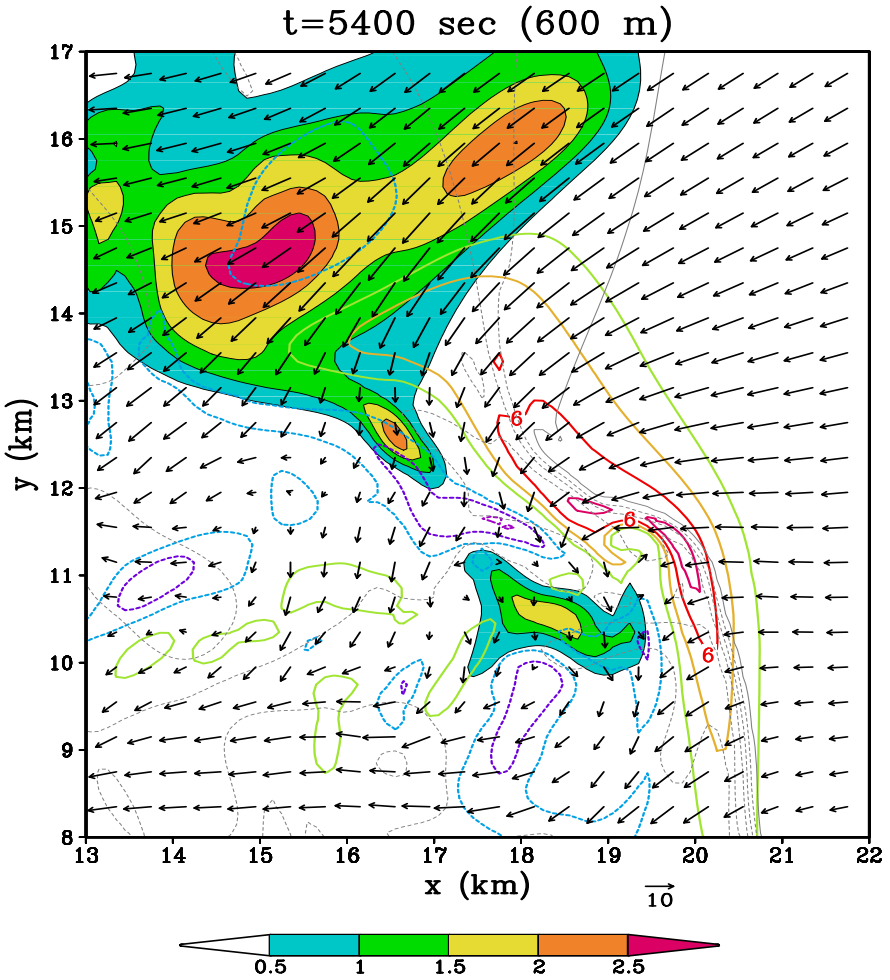

Figure 2.6. Horizontal display at 600 m of the simulated supercell at 5400 seconds from the initial. Mixing ratio of rain (gray scales, g kg^{-1}), vertical velocity (thick lines, m s^{-1}), the surface potential temperature at 15 m (thin lines, K) and horizontal velocity vectors.

Typhoon T0418 moved northwestward over the northwest Pacific Ocean and passed Okinawa Island on 5 September 2004. Its center passed Nago City around 0930 UTC, 5 September with the minimum sea level presser of 924.4 hPa. When T0418 pass over Okinawa Island, double eyewalls were observed. This is a distinctive feature of the typhoon. T0418 was characterized by strong winds and caused huge disasters due to the strong winds over Japan.

The main objectives of the simulation experiment of T0418 are to study the eyewall as well as spiral rainbands, and to examine structure of the strong wind associated with the typhoon around Okinawa Island. The simulation experiment of T0418 started from 0000 UTC, 5 September 2004. The experimental design of T0418 is summarized in Table 2.4.

The simulation experiment shows very detailed structure of the eye and the spiral rainbands (Fig.2.12). Individual cumulus clouds are resolved. They are simulated within the eyewall and along the spiral rainband. A weak precipitation forms around the central part of the eye. The maximum tangential velocity is present along the eyewall and at a height of 1 km. It is larger than 70 m s^{-1}. The high-resolution experiment shows detailed structure of the cloud and precipitation

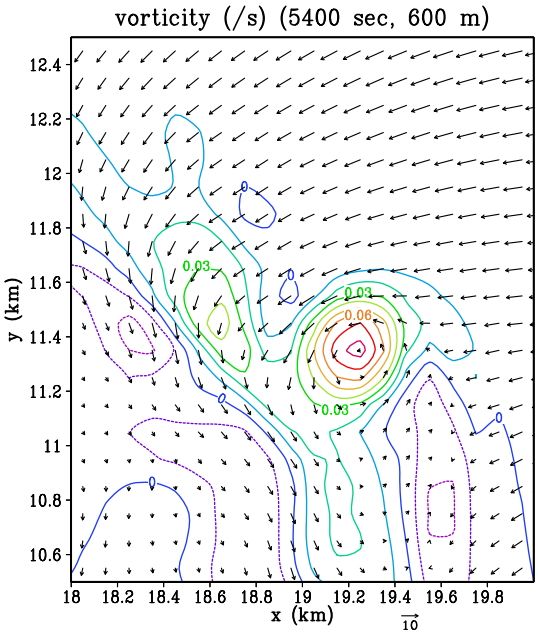

Figure 2.7. Close view of the simulated tornado within the supercell. The contour lines are vorticity (s^{-1}) and the arrows are horizontal velocity. The arrow scale is shown at the bottom of the figure.

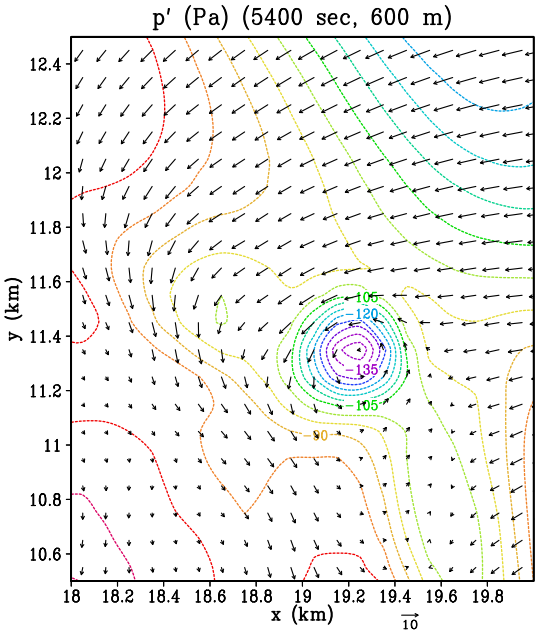

Figure 2.8. Same as Fig.2.7, but for pressure perturbation.

systems associated with the typhoon, and simulates the overall structure of the typhoon and its movement.

Typhoon T0423 moved along the Okinawa Islands on 19 October 2004 and landed over Shikoku Island on 20 October. In contrast to T0418, T0423 is characterized by heavy rainfall over Japan.

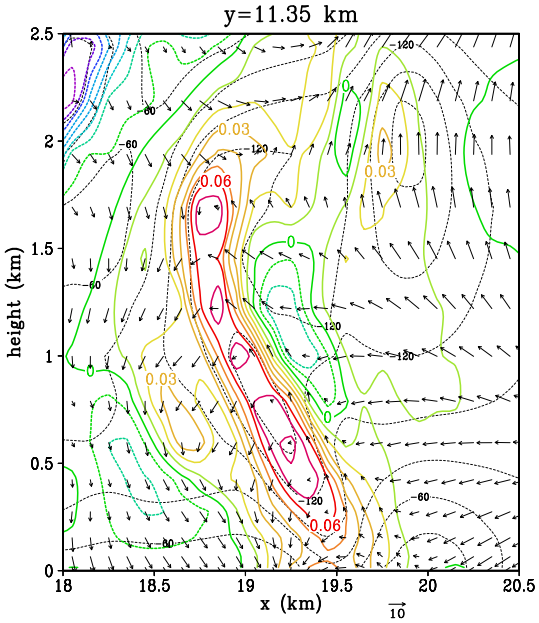

Figure 2.9. Vertical cross section of the simulated tornado. Thick lines are vorticity (s^{-1}), dashed lines are pressure perturbation and arrows are horizontal velocity.

Table 2.4. Experimental design of Typhoon T0418

domain	x 1536 km, y 1280 km, z 18 km
grid number	x 1539, y 1283, z 63
grid size	H 1000m, V 150 ∼300m
integration time	18 hrs
ES node numbers	128 nodes (1024 CPUs)

Heavy rainfalls associated with T0423 occurred in the eastern part of Kyushu, Shikoku, the east coast of the Kii Peninsula, and the Kinki District. They caused severe floods and disasters in these regions.

The purpose of the simulation experiment of T0423 is to study process of the heavy rainfall. Experimental design of T0423 is summarized in Table 2.5. At the initial time of 1200 UTC, 19 October 2004, T0423 was located to the NNE of Okinawa.

The movement of T0423 and the rainfall were successfully simulated. In the simulation, a northward moisture flux is large in the east side of the typhoon center. When the large moisture flux reaches to the Japanese Islands, heavy rainfalls occur along the Pacific Ocean side. The heavy rainfall moves eastward with the movement of the typhoon from Kyushu to Shikoku. When the typhoon reaches to the south of Shikoku, heavy rainfall begins in the Kinki District (the rectangle in Fig.6) and intensifies at 0630 UTC, 20 October (Fig.2.13). The distribution and intensity of precipitation well correspond to those of the radar observation.

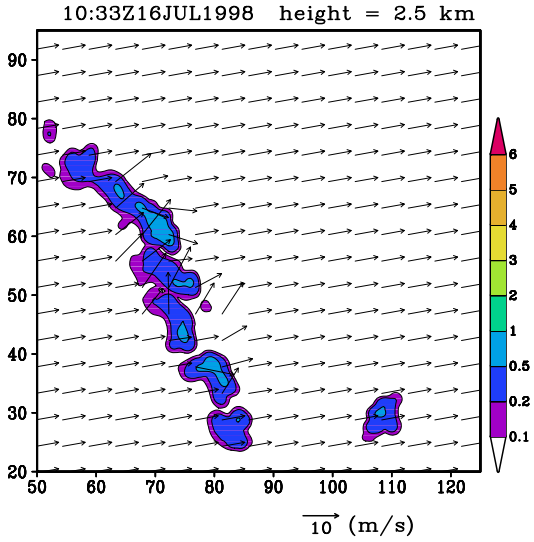

Figure 2.10. Horizontal cross section of the initial field at a height of 2.5km at 1033 UTC, 16 July 1998. The color levels mixing ratio of rain (g kg^{-1}). Arrows show the horizontal velocity obtained by the dual Doppler analysis and sounding.

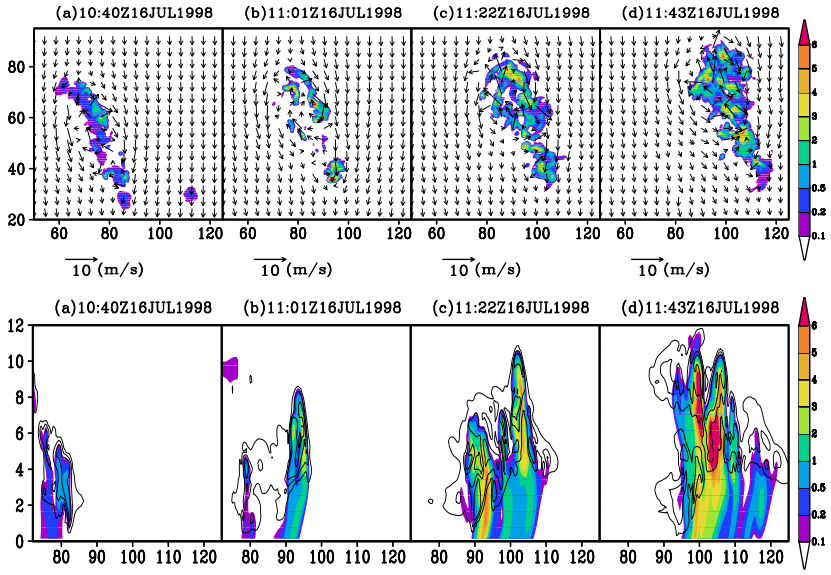

Figure 2.11. Time series of horizontal displays (upper row) and vertical cross sections (lower row) of the simulated squall line. Color levels indicate total mixing ratio (g kg^{-1}) of rain, snow and graupel. Contour lines indicate total mixing ratio (0.1, 0.5, 1, 2 g kg^{-1}) of cloud ice and cloud water. Arrows are horizontal velocity.

The close view of northern Kinki shows that a large amount of solid precipitation is present around a height of 6 km (Fig.2.14). The heavy rainfall in the region forms below the large mixing ratio of the solid precipitation. The heavy rainfall along the Pacific Ocean side moved eastward, while that in the Kinki District lasted until 12 UTC, 20 October. After the typhoon moved to the east of the Kinki District, the northeasterly was intensified significantly. Consequently, orographic rainfall formed in the northern Kinki District. As a result, the accumulated rainfall

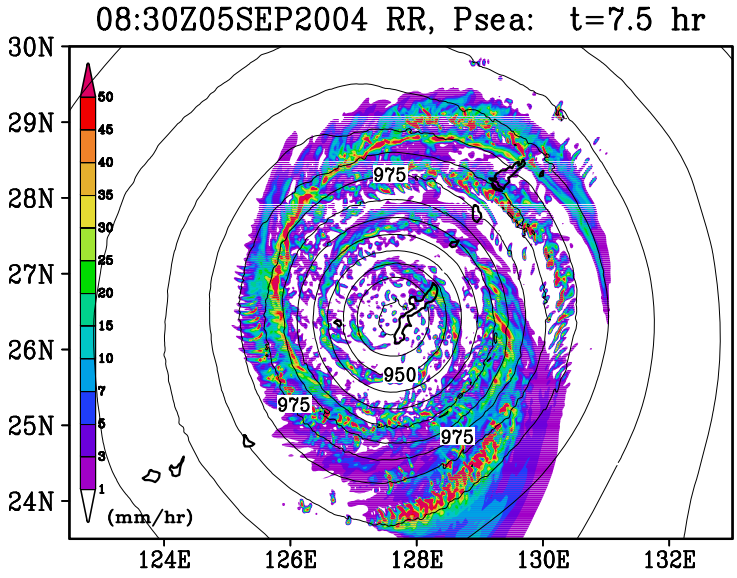

Figure 2.12. Surface pressure (contour lines; hPa) and rainfall intensity (color levels; mm hr^{-1}) of the simulated Typhoon T0418 at 0830 UTC, 5 September 2004.

Table 2.5. Experimental design of Typhoon T0423

domain	x 1536 km, y 1408 km, z 18 km
grid number	x 1539, y 1411, z 63
grid size	H 1000m, V 200 ∼300m
integration time	30 hrs
ES node numbers	128 nodes (1024 CPUs)

became a large amount and the severe flood occurred.

2.6 Typhoon RUSA

A typhoon named RUSA attacked Korea on 31 August 2002 and caused a heavy rain and flood. In particular, a large amount of rainfall was brought to the southern part of Korea and the east coast. We performed a simulation experiment of Typhoon RUSA with a very high resolution using CReSS. The horizontal resolution is 1 km and the domain is 1320 times 1440 km in horizontal and 18 km in vertical (Table 2.6). The initial and boundary conditions are provided by the Regional Spectral Model of JMA.

Typhoon RUSA moved northward over the East China Sea and landed on the southern part of the Korean Peninsula. The experiment well simulated the track of the typhoon. The distribution of total amount of rainfall well corresponds to observation. The overall characteristics of Typhoon

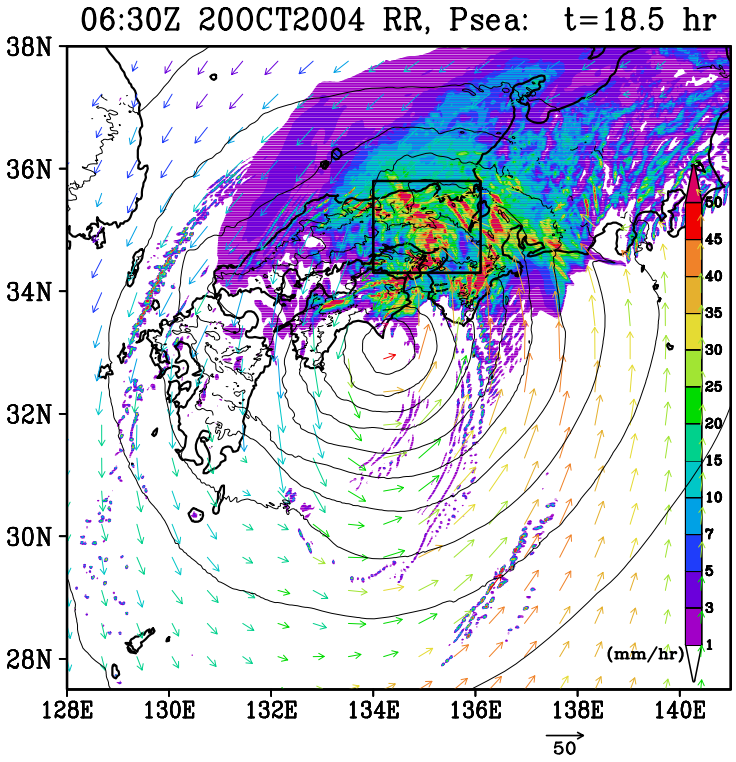

Figure 2.13. Same as Fig.2.12 but for the Typhoon T0423 at 0630 UTC, 20 September 2004. Arrows are horizontal wind velocity at a height of 974 m and warmer colored arrows means moister air. The rectangle indicates the region of Fig.2.14.

RUSA are successfully simulated.

The result shows that an intense spiral band forms on the northeast side of the typhoon center (Fig.2.15). When the spiral band approaches to the southernmost part of the Korean Peninsula, the rain intensifies by the topographic forcing. This causes the heavy rainfall in the southern Korean Peninsula.

Another intense rainfall occurs along the east coast of the Korean Peninsula. It was caused by convergence of a northeasterly wind and the cyclonic flow around the typhoon. The topography along the east coast also intensifies the rainfall. As a result, the intense rainfall occurs along the east coast of the peninsula.

The experiment with the high resolution and the large domain successfully simulated the whole typhoon and individual spiral bands as well as the intensification of rain due to topography. This

Table 2.6. Experimental design of Typhoon RUSA

domain	x 1320, y 1440, z 18 km
grid number	x 1323, y 1443, z 63
grid size	H 1000 m, V 300 m
integration time	24 hrs
node number	120 (960 CPUs)

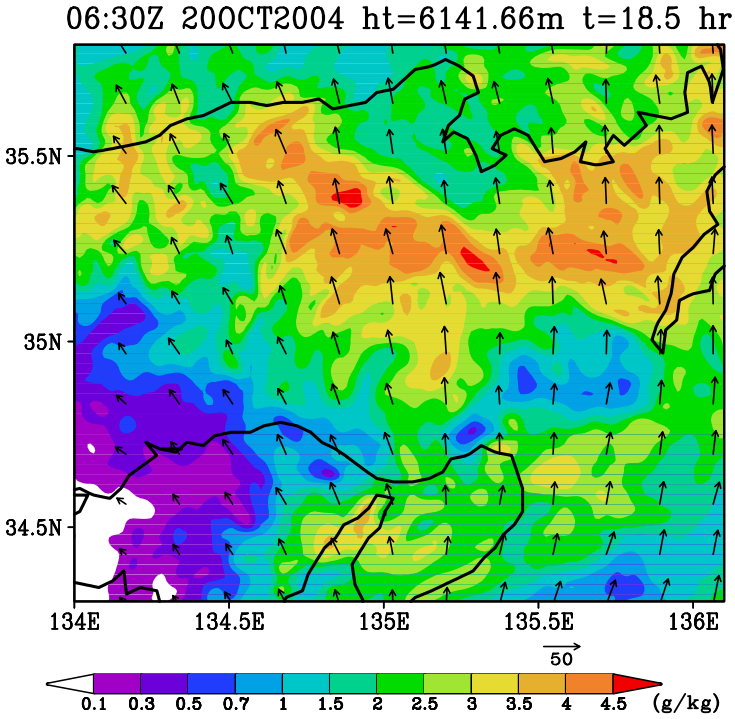

Figure 2.14. Mixing ratio of precipitation (color levels; g kg^{-1}) and horizontal velocity (arrows) at a height of 6142 m at 0630 UTC, 20 September 2004.

experiment enables to study the formation and development processes of the intense rain within the typhoon.

2.7 Localized heavy rainfall

Precipitation systems associated with the Baiu front occasionally show a multi-scale structure. The Baiu front extends zonally for several thousand kilometers while a localized heavy rainfall has a horizontal scale of a few hundred kilometers. To clarify the water circulation process and the role of each class of the multi-scale systems, we performed a simulation experiment of the heavy rainfall in a large domain and with a high resolution. The explicit representation of cumulonimbus clouds in the model is essentially important for accurate and quantitative simulation of the localized heavy rainfall.

The localized heavy rainfall occurred in Niigata and Fukushima prefectures on 13 July 2004. Radar observation of the Japan Meteorological Agency (JMA) showed that an intense rainband extended zonally and maintained for more than 6 hours. The Baiu front was present to the north of Niigata and a sub-synoptic scale low (SSL) moved eastward along the Baiu front.

The experimental setting of the simulation is summarized in Table 2.7. The initial and boundary condition were provided by the JMA Regional Spectral Model (RSM). Initial time is 1200 UTC, 12 July 2004. The simulation showed that the SSL moved eastward along the Baiu front. Figure 2.16 shows that the SSL reaches Japan at 0020 UTC, 13 July 2004. Moist westerly wind is intense to the south of the SSL. Large precipitation extends to the east of the SSL. On the other hand, a very intense rainband forms to the south of the SSL. Enlarged display (Fig.2.17) of the rainband

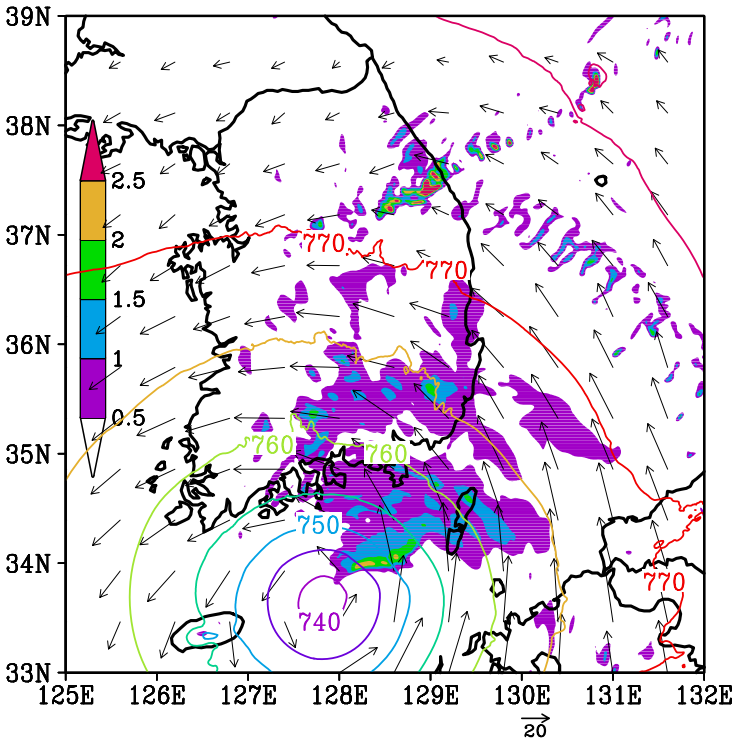

Figure 2.15. Pressure (hPa), mixing ratio of rain (gray scale; g kg^{-1}) and horizontal wind at a height of 2.15km at 03 UTC, 31 August 2002 obtained from the simulation experiment.

shows that it extends from the northern part of the Noto Peninsula and reaches Niigata with intensification. The rainband forms between the southwesterly and westerly winds at the low level. The rainband is composed of intense convective cells. It maintains until the SSL moves to the Pacific Ocean. The long time maintenance of the intense rainband results in the severe flood in Niigata Prefecture.

Table 2.7. Experimental design of the Niigata-Fukushima heavy rainfall event.

domain	x 1792 km, y 1536 km, z 18 km
grid number	x 1795, y 1539, z 63
grid size	H 1000 m, V 100 $\sim$ 300 m
integration time	24 hrs
ES node number	128 nodes (1024 CPUs)

2.8 Idealized experiment of snow cloud bands

When an outbreak of a cold and dry polar airmass occurs over the sea, many cloud streets or cloud bands form in the polar air stream. Large amounts of sensible heat and latent heat are supplied from the sea to the atmosphere. Intense modification of the airmass results in development of the mixing layer and convective clouds develop to form the cloud bands along the mean wind direction. Their length reaches an order of 1000 km while individual convective

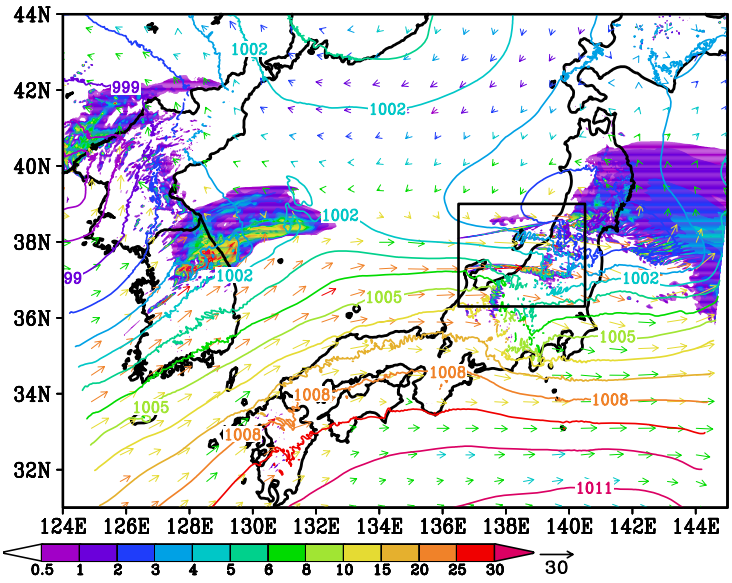

Figure 2.16. Surface pressure (contour lines; hPa) and rainfall intensity (color levels; mm hr^{-1}) and horizontal velocity (arrows) at a height of 1610 m at 0020 UTC, 13 July 2004. Warmer colored arrows means moister air. The rectangle indicates the region of Fig.2.17.

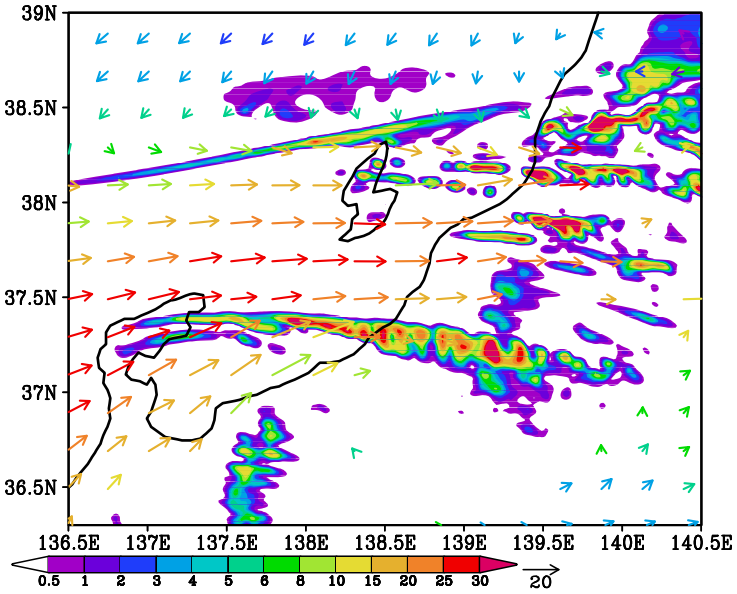

Figure 2.17. Same as Fig.2.16 but for the region of the rectangle in Fig.2.16 and at a height of 436 m.

cells have a horizontal scale ranging from a few kilometer to a few tens kilometers. In order to perform a 3-dimensional simulation of cloud bands, a large computation is necessary. In order to study the detailed structure of convective cells and the formation process of the organized cloud bands, we performed 3-dimensional simulation using the CReSS model on the Earth Simulator.

The experimental setting is summarized in Table 2.8. The initial condition was provided by a sounding observation on the east coast of Canada at 06 UTC, 8 February 1997.

Table 2.8. Experimental design of the snow cloud bands in the cold air stream.

domain	x 457 km, y 153 km, z 11 km
grid number	x 1527, y 515, z 73
grid size	H 300 m, V 50 ∼150 m
integration time	20 hrs
ES node number	32 nodes (256 CPUs)

The calculation domain in this simulation is 457 km and 153 km in x- and y-directions, respectively with a horizontal grid spacing of 300 m. Sea ice is placed on the upstream side. Each model grid of the surface is occupied by ice or open sea according to the probability of sea ice or sea ice density. In the experiment, density of sea ice is 100 0 outbreak is used for the initial condition.

The atmosphere over the packed ice is stably stratified and the vertical shear is large. Mixing layer develops with the distance from the edge of the packed ice. The cloud bands develop within the mixing layer. Figure 2.18 shows formation and development of cloud bands over the sea. They begin to form in the region of sea ice density of 50-70 cloud bands form in the cold air stream. Some cloud bands merge each other and selectively develop. Consequently, the number of lines decreases with the distance along the basic flow.

Close view of the upstream region (Fig.2.19) shows upward and downward motions are almost uniform in the x-direction. As a result, cloud ice and precipitation extend in the x-direction uniformly. This indicates that the convections are the roll convection type in the upstream region.

In the downstream region, the roll convections change to alignment of cellular convections (Fig.2.20). While the upward motions are centered and downward motions are located on their both sides, cloud and precipitation show cellular pattern. In this region, the mixing layer fully developed and the vertical shear almost vanishes in the mixing layer.

In the region of far downstream, convections change to randomly distributed cells (Fig.2.21). Band shape of cloud almost disappears and convections become closed cell type. The morphological transformation from alignment of cells to random cells is often observed by satellite. The experiment successfully simulates the formation process of cloud bands, their extension and merging processes, and the morphological changes of convection from the roll to cellular types.

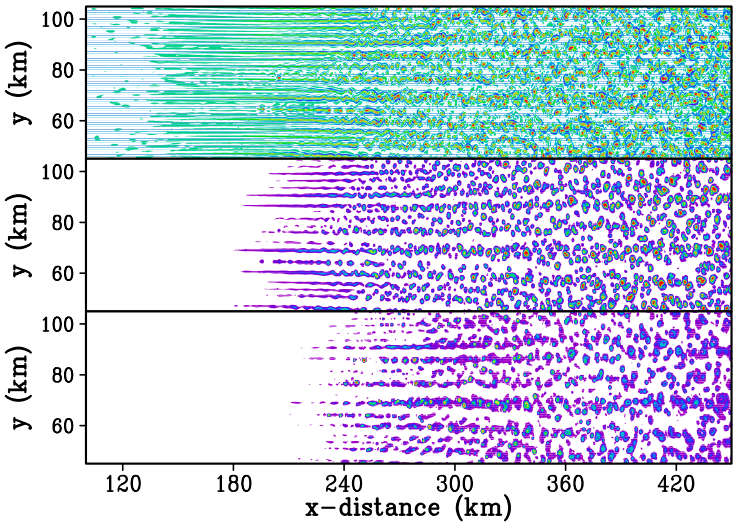

Figure 2.18. Horizontal cross sections of vertical velocity (upper panel), mixing ratio of precipitation (snow, graupel and rain) (middle panel) at 1000 m in height and mixing ratio of cloud ice at 1300 m in height (lower panel) for x=100-450 km at 18 hours from the initial time.

2.9 Snow storms

One of the major precipitation systems in East Asia is snow cloud in a cold polar air stream. In particular, various types of precipitation systems develop over the Sea of Japan: longitudinal and transversal cloud bands, convergence zone and vortexes. Their horizontal scale ranges from a few hundred kilometers to 1000 km while they are composed of convective clouds whose horizontal scale is a few kilometers.

To study development process and detailed structure of longitudinal and transversal cloud bands, we performed simulation experiment of the cold air outbreak over the Sea of Japan on 14 January 2001. The initial field at 0600 UTC, 13 January 2001 and boundary condition were provided by the JMA RSM. The Domain of the simulation covered most part of the Sea of Japan and horizontal resolution was 1 km to resolve convective clouds (Table 2.9).

Table 2.9. Experimental design of the snow storm over the sea of Japan.

domain	x 1350, y 1350, z 16 km
grid number	x 1353, y 1353, z 43
grid size	H 1000 m, V 200 ∼400m
integration time	18 hrs
ES node number	36 (288 CPUs)

Snow cloud bands over the Sea of Japan is realistically simulated (Fig. 2.22). An intense and thick cloud band composed of cumulonimbus clouds extends from the root of the Korean Peninsula to the Japan islands. Plenty of thin cloud bands develop over the sea. Longitudinal

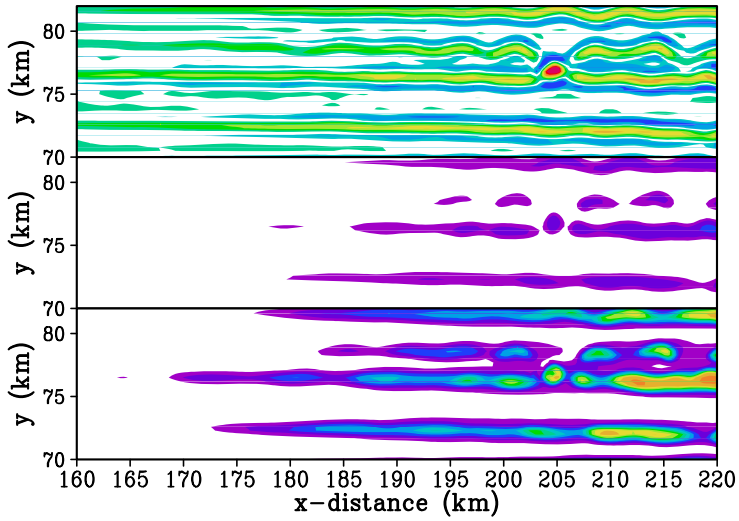

Figure 2.19. Horizontal cross sections of vertical velocity (upper panel), mixing ratio of precipitation (middle panel) at 900 m in height and mixing ratio of cloud ice at 1100 m in height (lower panel) for x=160-220 km.

and transversal cloud bands form to the west and east of the intense cloud band, respectively. The enlarged display of the transversal cloud bands shows that the cloud bands extend the SW-NE direction, which is almost parallel to the vertical wind shear between levels of the top of clouds and the surface (Fig. 2.23). This is consistent with the dynamic theory shown by Asai (1972)[4].

Cold air outbreak and associated precipitation systems occurs over the Great Lakes region and the Labrador Sea to the east of Canada. The snow cloud bands over the Great Lakes are known as the 'lake effect storm (LES)' and have been studied observationally and numerically. Liu et al. (2004) studied snow cloud bands using CReSS. The LES is also composed of cumulonimbus clouds and brings severe snowfall. We performed a simulation experimet (Table 2.10) to study the formation of the LES associated with a cold air outbreak. Figure 2.24 shows vertically integrated cloud and precipitation obtained from the simulation experiment. Cloud bands begin to develop from the lake on the upwind side (Lake Superior). They are composed of convective clouds over the lake while they become stratiform over the land. When they arrive at the lakes on the downwind side (Lakes Michigan and Huron), convection in the cloud bands is restored. The pattern and structure of cloud are very similar to those observed by meteorological satellites.

Table 2.10. Experimental design of the snow storm over the Great Lakes.

domain	x 1300, y 660, z 18 km
grid number	x 2603, y 1323, z 43
grid size	H 500m, V 40 ∼250m
integration time	12 hrs
ES node number	120 (960 CPUs)

Cold air outbreak over the Labrador Sea is similar to that over the Sea of Japan. Sea ice is present along the east coast of Canada. Observation shows that plenty of snow cloud bands

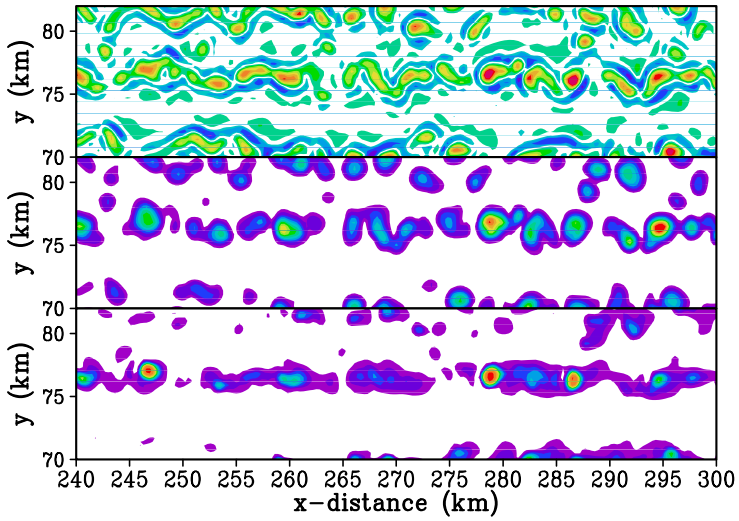

Figure 2.20. Horizontal cross sections of vertical velocity (upper panel), mixing ratio of precipitation (middle panel) at 1000 m in height and mixing ratio of cloud ice at 1300 m in height (lower panel) for x=240-300 km.

develop over the sea. Some of them begin to form over the sea ice. Liu et al. (2005) studied the effect of the sea-ice zone on the development of the band clouds using CReSS. In order to study more detailed structure and formation process of cloud bands and a vortex over the Labrador Sea, we performed a simulation experiment using a large calculaton domain and very fine horizontal grid size (Table 2.11). In the simulation experiment, sea ice is placed along the coast according to the density of sea ice.

Table 2.11. Experimental design of the snow storm over the Labrador Sea.

domain	x 1000, y 840, z 12 km
grid number	x 2003, y 1683, z 48
grid size	H 500m, V 60 ∼250m
integration time	10 hrs
ES node number	80 (640 CPUs)

Figure 2.25 shows that cloud bands begin to develop from the sea ice region and convection composing the cloud bands intensifies over the open sea. In this experiment, a vortex is also simulated over the Labrador sea as observed. This is a sort of polar lows. The simulation experiment shows the mesoscale vortex, cloud bands as well as convective clouds composing these precipitation systems with the very high resolution.

2.10 Heavy rainfall in Baiu season

The heavy rain event occurred on 19–20 July 2003 in Kyushu which is the western part of Japan. During this period, the Baiu front was located to the north of Kyushu. The most intense rain occurred during the period of 16–22 UTC, 20 July and the area of the heavy rain was roughly

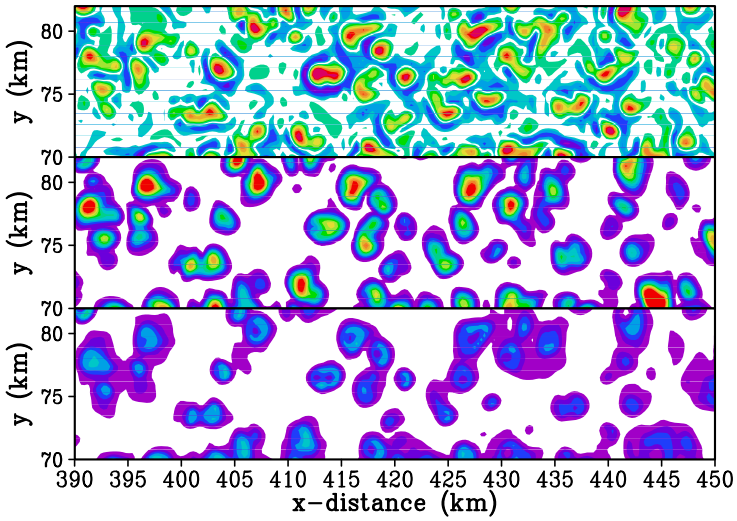

Figure 2.21. Same in Fig.2.20 , but for x=390-450 km.

100×100 km. The total amount of rain of this period reached about 210 mm at Minamata which is located in the western Kyushu. We consider that this is a localized heavy rain. The heavy rain caused a flood and 21 people were killed.

Satellite images show that a convective cloud cluster developed on the west coast of Kyushu. The AMeDAS-radar composite data provided by JMA (the Japan Meteorological Agency) shows that intense echo systems developed within the cloud cluster and moved into the west coast of Kyushu (Fig.2.26). Figure 2.26 shows that an intense rainband extends in east-west direction and some orographic rainfall echo developed in the western Kyushu.

We performed a prediction experiment of the localized heavy rain using CReSS. The experimental design is summarized in Table 2.12. The initial time is 00 UTC, 19 July 2003 and 24-hour experiment was performed. The horizontal domain is centered at the west coast of Kyushu.

Table 2.12. Experimental design of the localized heavy rain in Kyushu using 2 km horizontal resolution.

domain	x 600, y 600, z 18 km
grid number	x 603, y 603, z 63
grid size	H 2km, V 100 ∼150m
integration time	24 hrs
node number	10 (80 CPUs)

The experiment successfully simulated the localized heavy rain. Figure 2.27 shows that an intense rainband developed to the west of Kyushu and the intense rain occurred along the west coast of Kyushu. The rainfall pattern in Fig.2.27 is quite similar to the observation (Fig.2.26) with regard to the pattern and intensity of rain. The intense rain in the western Kyushu is simulated during the period from 16–22 UTC, 19 July 2003.

We also performed prediction experiment with a fine gird size of 500 m (Table 2.13). The

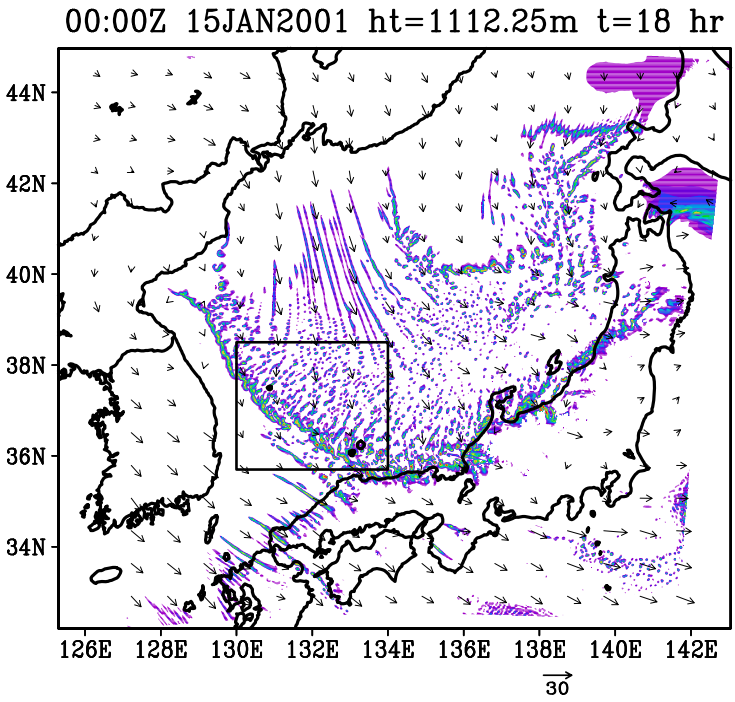

Figure 2.22. Mixing ratio of precipitation (color levels; g kg^{-1}) and horizontal velocity (arrows) at a height of 1112 m at 0000 UTC, 15 January 2001.

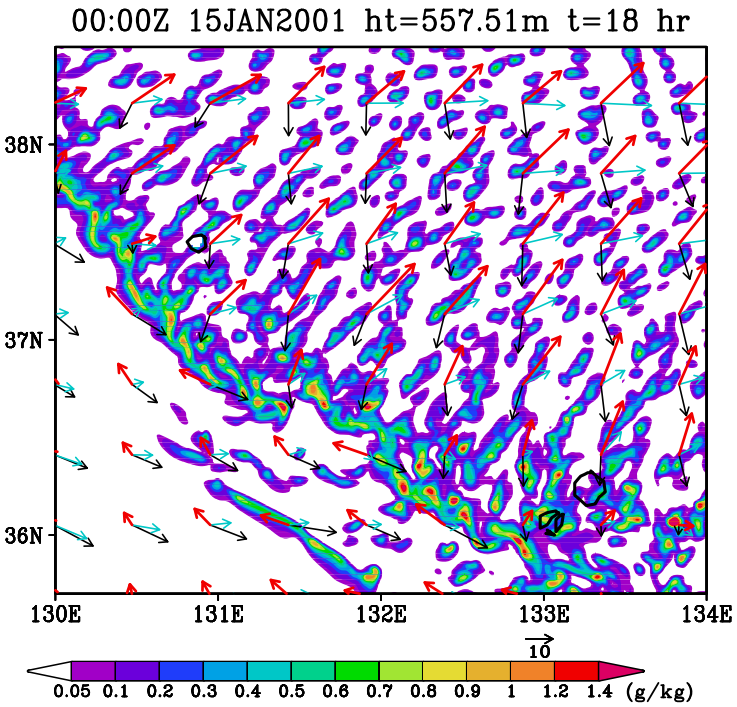

Figure 2.23. Mixing ratio of precipitation (gray scale; g kg^{-1}). Black and blue arrows are horizontal wind velocity at height of 315 and 2766 m, respectively. Read arrows are wind shear between these levels.

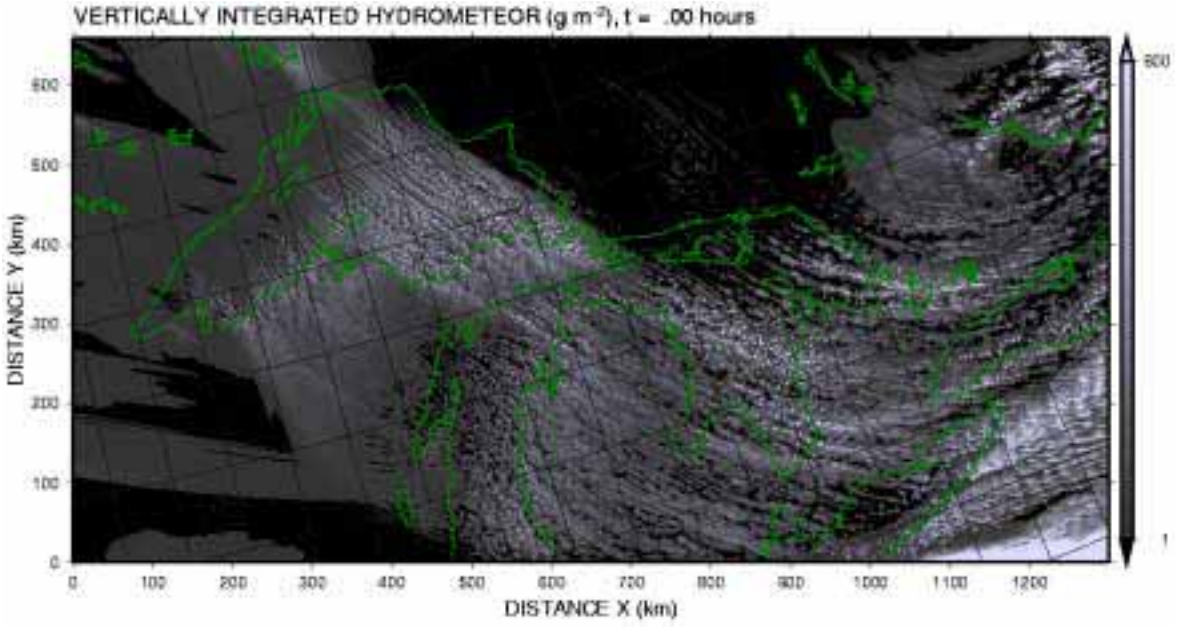

Figure 2.24. Vertically integrated mixing ratio of precipitation (gray levels; g m^{-2}) over the Great lakes at 12 hours from the initial time.

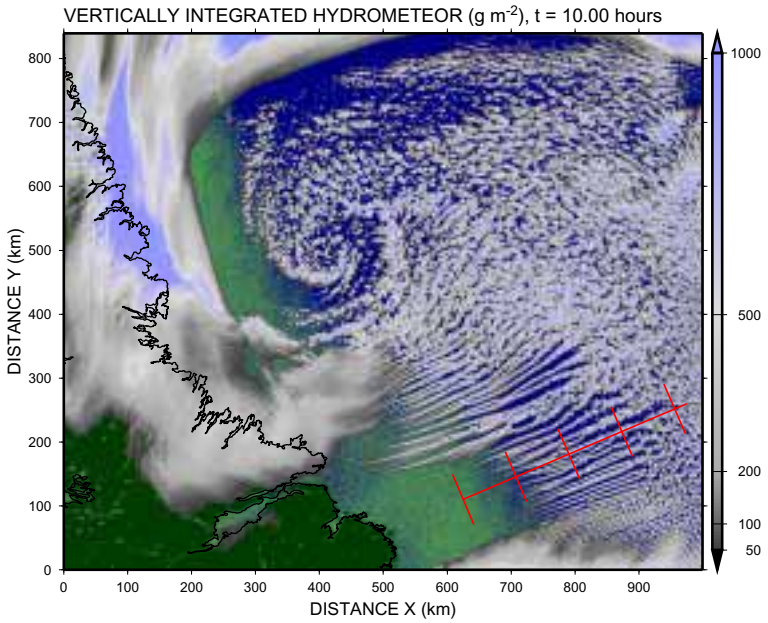

Figure 2.25. Verticall integrated mixing ratio of prcipitation (gray levelsg m^{-2}) over the Labrador Sea at 10 hours from the initial time. Dark green and light green mean land and sea ice, respectively.

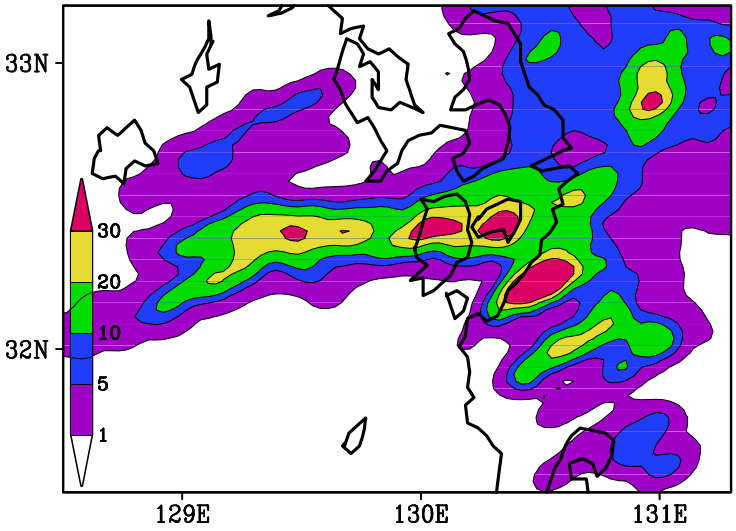

Figure 2.26. Rainfall rate of AMeDAS-radar composite (mm hr^{-1}) at 16 UTC, 19 July 2003.

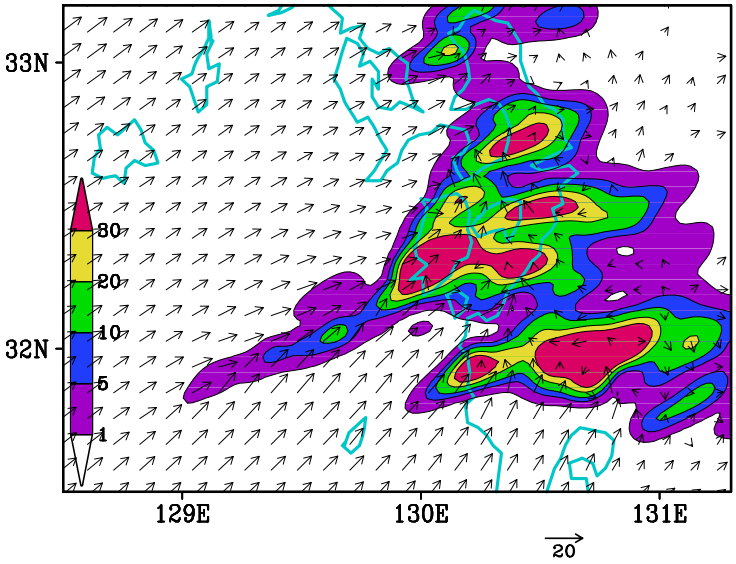

Figure 2.27. Rainfall amount during 15-16 UTC and horizontal wind at 500 m ASL at 16 UTC, 19 July 2003 obtained from the prediction experiment of 2 km resolution.

increase of horizontal resolution shows the detailed structure of the convective system (Fig.2.28). The rainband extends from SW to NE direction. It is composed of intense convective cells. The simulation resolves individual convective cells. The experiment with the high resolution also shows orographic precipitation. Figure 2.28 shows that another rainband is present to the south of the main rainband. It forms on the lee side of the small islands over the sea, which is named the Koshiki islands. We consider that both the two rainbands is related to the localized heavy rainfall.

Table 2.13. Experimental design of the localized heavy rain in Kyushu using 500 m horizontal resolution.

domain	x 600, y 600, z 18 km
grid number	x 1203, y 1203, z 63
grid size	H 500m, V 100 ∼150m
integration time	21 hrs
node number	120 (960 CPUs)

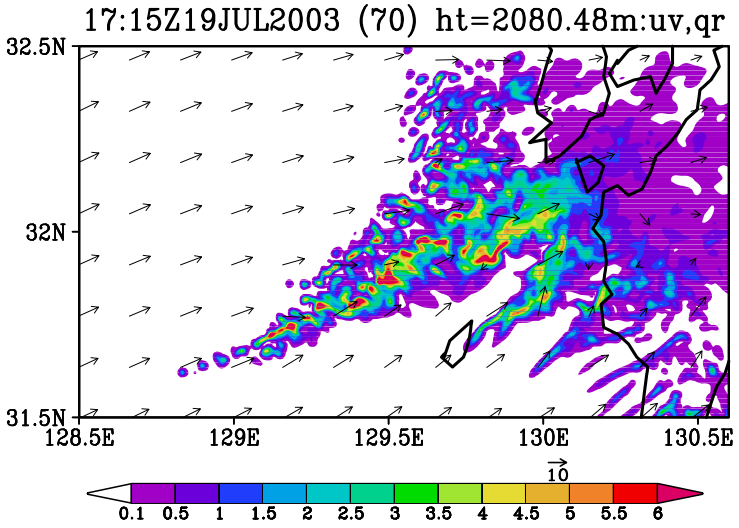

Figure 2.28. Mixing ratio of rain (gray scale; g/kg) and horizontal velocity (arrows) at a height of 2080m at 1715 UTC, 19 July 2003 obtained from the experiment of 500 m resolution.

2.11 Typhoon 0613 and associated tornadoes

Typhoons occasionally accompany tornadoes and cause disasters due to strong winds. When the typhoon 0613 (T0613) approached the western Japan on 17 September 2006, it accompanied a severe tornado (tatsumaki in Japanese). It killed three people and caused the train accident in Kyushu. We performed a simulation experiment of T0613 and the associated tornado using the CReSS model to reveal structure of clouds along the typhoon rainband and the mechanism of the tornado. The horizontal grid size is 500m in the simulation of the typhoon and the calculation

domain is sufficiently large to simulate overall structure of the typhoon. The initial condition (00 UTC, 17 September 2006) and the boundary conditions were provided by the regional model output of the Japan Meteorological Agency (JMA). We used 128 nodes of the Earth Simulator for the simulation. Intense spiral rainbands were formed in the eastern part of T0613 in the simulation at about 5hours from the initial time. One of them passed over Nobeoka City at around 05 UTC, 17 September. This corresponds to the observation of JMA. The enlarged display of the rainbands showed that some clouds composing the rainband are intense supercells.

A high-resolution experiment was performed to study the relationship between the supercells and the tornado with a grid spacing of 75m. The result shows that hook-shaped structure forms in the southernmost part of the supercell and that the intense tornado is simulated in the hook-shaped part of the supercell (Fig.2.29). The horizontal diameter of the tornado is about 300m and its maximum vorticity is larger than 0.9 s^{-1} (Fig.2.30). Pressure perturbation is about -24 hPa at the center of the tornado and maximum horizontal velocity reaches to 70 m s^{-1}. The vertical cross-section (Fig.2.31) shows that the tornado extends up to a height of 2.5 km in strong upward motion. The simulation showed the detailed structures of the tornado and the supercell in the typhoon rainband.

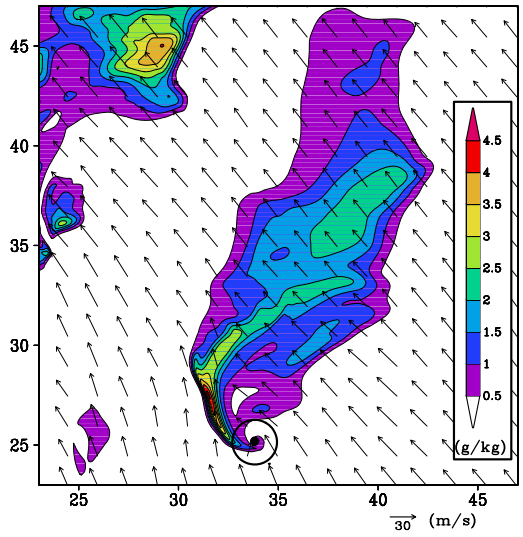

Figure 2.29. Supercell cloud simulated in the experiment with a horizontal grid spacing of 75m at 0500 UTC, 17 September 2006. Color levels indicate rain mixing ratio (g kg^{-1}) at 200 m in height. Arrows indicate horizontal velocity. The circle indicates that the tornado occurs in the southernmost part of the supercell.

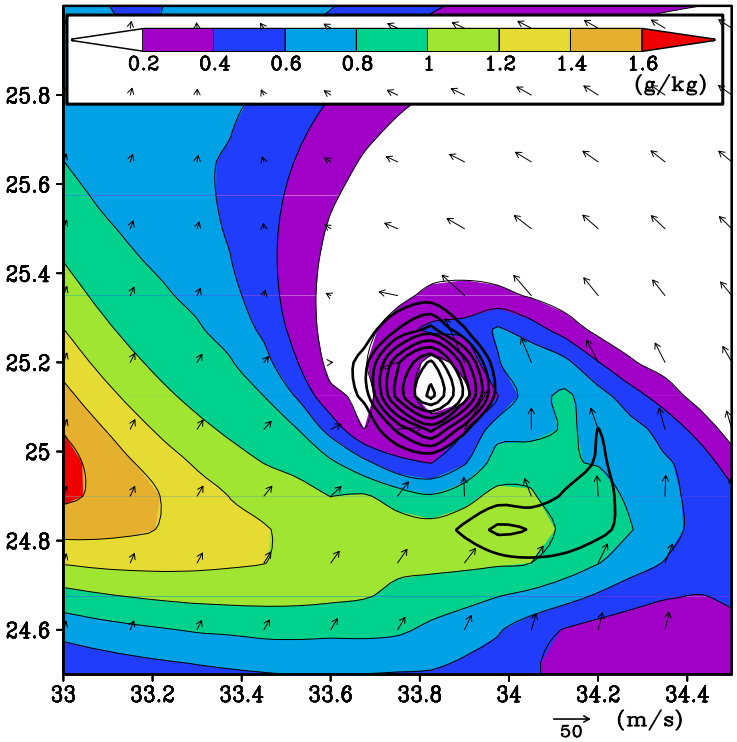

Figure 2.30. Enlarged view of the southernmost part of the supercell. Contours are vorticity drawn every 0.1 s^{-1} from 0.1 s^{-1} and color levels are cloud mixing ratio (g kg^{-1}).

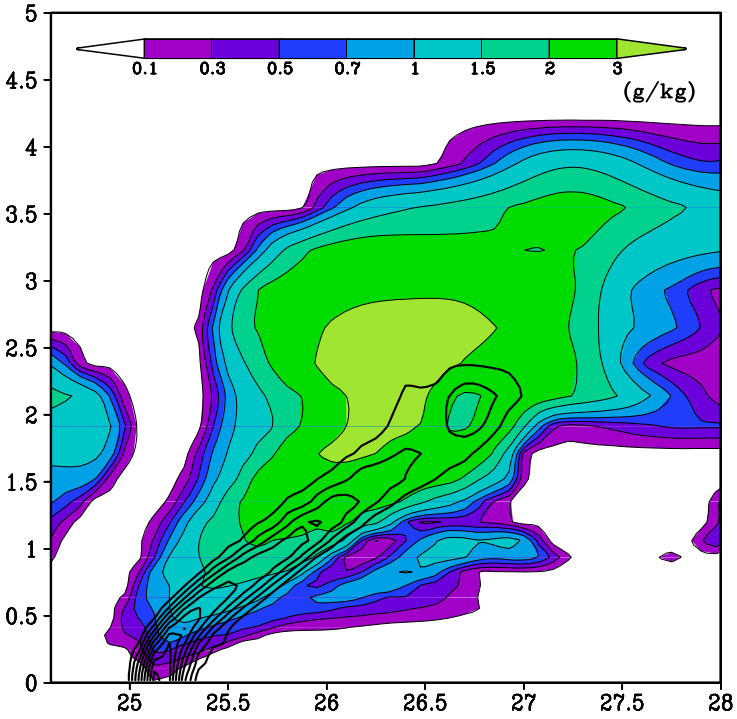

Figure 2.31. Vertical cross-section of the tornado in the north-south direction. Contours are vertical vorticity drawn every 0.1 s^{-1} from 0.1 s^{-1} and color levels are cloud mixing ratio (g kg^{-1}).

Part II

Technical Description

Chapter 3

Start to Use CReSS

This chapter is devoted to those who start to use the CReSS model for the first time. Most documents of this chapter are included in the file of readme. If you are already familiar with CReSS, you can skip this chapter. Uncompressing archived files, compilation and some example to use CReSS will be shown in the preset chapter.

CReSS has two versions: one is single version and the other is parallel version. The former is indicated by "s" in the file name and the latter by "m". These two versions are totally identical. The result of calculations by these two versions will be completely same. The only difference of the two versions is necessity of the MPI library for parallel computation. The single version can be performed without the MPI library on a single processor.

3.1 Download source files

At first, please download the following files from our download page at http://www.rain.hyarc.nagoya-u.ac.jp/ cress/download/ with your ID and password. If you have not have your ID and password, please send an e-mail to tsuboki@rain.hyarc.nagoya-u.ac.jp, to request for getting ID and password. From the URL, you can download the following files.

Table 3.1. CReSS archived files

cress2.2m.tar.Z	all file of CReSS parallel version.
cress2.2s.tar.Z	all file of CReSS serial version.

Since the users' guide is not always updated, find the readme files in the tar files and refer them. They are always updated and give a proper description.

In order to start to use the CReSS model, take the following processes. We assume that you use a UNIX system such as LINUX or other type of UNIX systems. If you use parallel system, use the cress2.2m.tar.Z. In the following part, we will explain the single version. The basic usage is totally the same except to specify the number of processing elements.

3.2 Uncompress archived file

To uncompress the archived file of CReSS, use the following command of UNIX.

```
% uncompress -c cress2.2s.tar.Z |tar -xvf -
```

Then, the following files will be found in the directory.

```
% cd CReSS2.2s/
% ls
./              Doc/            Src/            compile.conf
../             Form/           User_Mod/       compile.csh*
```

The document files will be found in the directory.

```
% cd Doc
% ls
/                ../              readme.first       readme.user.conf
```

Read the "readme.first", then you can find how to compile the source. Then, read "readme.user.conf". This readme is important to modify the namelist input.

3.3 Compilation and samples

To compile the solver of CReSS which is main processor of the model, enter the following commands. If necessary, you should edit the configuration file "compile.conf".

```
% vi compile.conf
% compile.csh solver
```

After you compile the executable of solver.exe, please try a sample.

```
%cd Form
%ls
./                                  sounding.txt.toyohashi.form
../                                 user.conf.cats.eye.form
sounding.txt.cats.eye.form          user.conf.dry.downburst.form
sounding.txt.dry.downburst.form     user.conf.mountain.wave.form
sounding.txt.mountain.wave.form     user.conf.toyohashi.form

% cp user.conf.mountain.wave.form  ../user.conf
% cp sounding.txt.mountain.wave.form  ../test.sounding.txt
```

The first part of the name of the sounding file "test" is important. All I/O files of the CReSS files are controlled by the name which is find in the user.conf:

```
&runame
  exprim = 'test'
/
```

The prefix of all files is should be the same. In this case, it is "test".

To run the model, send the following commands.

```
% ./solver.exe < user.conf > & log.solver &
```

To check the log file,

```
% tail -f  log.solver
```

The output history dump files will be produced such as,

```
% ls  -l  *.bin

-rw-r--r--   1 tsuboki  kisho    1866240 038 12:13 test.dmp000000.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:14 test.dmp000600.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:16 test.dmp001200.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:18 test.dmp001800.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:20 test.dmp002400.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:21 test.dmp003000.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:23 test.dmp003600.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:25 test.dmp004200.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:27 test.dmp004800.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:29 test.dmp005400.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:30 test.dmp006000.bin
-rw-r--r--   1 tsuboki  kisho    1866240 038 12:32 test.dmp006600.bin
-rw-r--r--   1 tsuboki  kisho       4320 038 12:13 test.geography.bin
```

3.4 To Display the result

The output history dump files are displayed by the application soft wear "grads". You can find it in the URL at http://www.iges.org/grads/grads.html

The Grads need control file. In the above case, the ctl file will be

```
%cat dmp.ctl

DSET ^test.dmp%y400.bin
OPTIONS template
UNDEF -1.E35
XDEF  360 LINEAR  0.200    0.400
YDEF    1 LINEAR  0.200    0.400
ZDEF  144 LINEAR  0.062.5  0.125
TDEF  100 LINEAR 00Z01jan0000 6yr
VARS  9
ub  144  99  base state x components of velocity [m/s].
u   144  99  x components of velocity [m/s].
vb  144  99  base state y components of velocity [m/s].
v   144  99  y components of velocity [m/s].
w   144  99  z components of velocity [m/s].
pb  144  99  base state pressure [Pa].
pp  144  99  pressure perturbation [Pa].
ptb 144  99  base state potential temperature [K].
ptp 144  99  potential temperature perturbation [K].
ENDVARS
```

Use the dmp.ctl file to display the result by Grads. Then, you will find the beautiful mountain waves.

Chapter 4

Configuration and Execution

In this chapter, configuration, flowchart of data, and execution of CReSS will be described. Two types of CReSS are present: one is the single version for a single processor and the other is the parallel version for multiple processors. This chapter describes mostly about the parallel version of CReSS. The configuration and execution of the serial version are essentially the same as those of the parallel version and the differences between the parallel and serial versions will be unspecified.

CReSS has two configuration files: one is `compile.conf` which configures the compile commands and their options, and the other is `user.conf` which controls all the executables of pre- and post processors and main solver. The former depends on the machine to be used while the latter is independent of mostly except `&sysdep`. In this chapter, the latter configuration file will be explained in detail.

The configuration file "user.conf" is written as the format of the Fortran namelist. The user.conf is used for not only run but also the compile. All the dimensions of array arguments are specified by the user.conf. Users, therefore, will never change the source codes of CReSS.

The flowchart of data will also described as well as data formats. There are different data flows which depend on the configuration of calculation.

At the last part of this chapter, an example run of experiment of the Kelvin-Helmholtz waves will be shown: compilation of solver, run the solver and output the result. Compilations and executions of the pre-processors will be also presented.

4.1 User configuration file

At present, CReSS version 2.2 and 2.3 are open to the public. The version 2.3 is almost the same as the version 2.2 but grads ctl file is automatically produced in the version 2.3. The configuration files are almost same. Most parts of the descriptions are based on the version 2.2. The parameters used only in the version 2.2 is indicated by the double dagger (‡). Those used only in the version 2.3 is indicated by the dagger (†).

4.1.1 Composition of user.conf

The user configuration file of CReSS, "user.conf" is composed of the following 7 regions.

common region: All pre-processors, solver, and post-processors refer to the region.

gridata region: The pre-processor "gridata.exe" refers to create initial and boundary files.

radata.region: The pre-processor "radata.exe" refers the region to create model grid values of radar data.

terrain region: The pre-processor "terrain.exe" refers the region to create model topography data.

surface region: The pre-processor "surface.exe" refers the region to set surface parameters on the model grid.

unite region: The post-processor "unite.exe" refers the region to unite output data in parallel computations.

rstruct region: The post-processor "rstruct.exe" refers the region to rearrange restart files of CReSS. Ordinary users will never use this region. Description of this region will be, therefore, omitted.

The user configuration file of CReSS is a namelist of Fortran. The followings are important notices to use the file.

- Since the configuration file is used by a shell script in the compilation of CReSS, a space is necessary to separate between a variable name and "=" and between a value and "=".
- The solver and the pre-processors have dependency of some namelist variables. When a discrepancy of variables between pre-processors and the solver is found, the program will stop execution and output an error.

4.1.2 Common region of the user configuration file

& sysdep

The namelist variable is machine-dependent. All programs use the variable, while no dependence is present between the programs.

`savmem` *integer*(4)

To avoid memory limitation of small computer, the 3-dimensional array is arranged into 2-dimensional array.

`0`: 2-dim. arrays are used.

`1`: 3-dim. arrays are used. [**recommended**]

& runame

The namelist specifies the name of the experiment, which is used by *solver*, all pre-processor, and all post-processor. Dependency is present between these programs. All I/O data set is controlled by the namelist.

`exprim` *character*(80)

Name of experiment. The number of character used in the variable must be less than or equal to 80 characters. The variable is used by all input and output data files as a part of name. For example, the sounding data file will be *exprim*`.sounding.txt`. No special characters are allowed to be used.

& dmiset

This namelist variables set numbers of grid points of calculation domain and numbers of decomposition in parallel computing. All programs use the variables and have dependency.

`xpedim,ypedim` *integer*(4)

Numbers of decomposition of the calculation domain in x and y directions. The total numbers of processing elements (CPUs) are $x \times y$.

`xdim,ydim` *integer*(4)

Numbers of grid points of the domain in the x and y directions, respectively. Since the horizontal grid system is the Arakawa C type, the numbers of the scalar grid points are $xdim - 1$ and $ydim - 1$ in the x and y directions, respectively. These numbers must satisfy the following relations in order to each processing element is loaded with the same number of grid points.

$$(\texttt{xdim} - 3)/\,\texttt{xpedim} = integer$$

$$(\texttt{ydim} - 3)/\,\texttt{ypedim} = integer$$

These relations means that numbers of grid points of the physical domain $(\texttt{xdim} - 3, \texttt{ydim} - 3)$ must be divided by each number of decomposition.

`zdim` *integer*(4)

Number of grid point in the z direction. Since the vertical grid system is the Lorenz type, the number of scalar grid points is $zdim - 1$. Using the parameters of `cphopt = 2` or `cphopt = 3`, `zdim` must be larger or equal 12 because of use of temporary array.

& project

The namelists define the horizontal coordinate systems of the calculation domain. They are used *solver*, pre-processors, and post- processors. Dependency is present between these programs.

`mpopt` *integer*(4)

Option for map projection of the domain. The following four projections are available.

- `1`: polar stereo graphic projection
- `2`: Lambert conformal projection **[recommended]**
- `3`: Mercator projection
- `4`: No map projection (latitude and longitude)
- `13`: Mercator projection of zonal circuit of the globe

`nspol` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

- `1`: the origin of coordinate in the northern hemisphere. **[recommended]**
- `-1`: the origin of coordinate in the southern hemisphere.

`tlat1,tlat2` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitude (`tlat1` and `tlat2`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative. The y-axis of the model domain corresponds to the longitude. Even in the case of `mpopt = 4`, `tlon` must be specified.

& gridset

The namelists determine grid intervals and reference longitude and latitude. They are used in *solver*, pre-processors, and post- processors. Dependency is present between these programs.

`dx,dy,dz` *real*(4)

Grid intervals in the x, y and z directions, respectively. The unit is meter. If non-uniform vertical grid is used (`sthopt = 1,2`) , `dz` is the averaged grid interval.

`ulat,ulon` *real*(4)

Reference longitude and latitude, respectively, which corresponds to a single point within the domain. The unit is degree. Degrees in the southern hemisphere and in the West Longitude are negative. CReSS specifies the region of the domain using the reference longitude and latitude with the following real number of `riu` and verb"rju".

`riu,rju` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat,ulon`).

& gridsth

The namelists determine vertical grid spacings in non-uniform coordinate. They are used in *solver*, pre-processors, and post- processors. Dependency is present between these programs.

`zsfc` *real*(4)

Altitude of the sea surface. The unit is meter. In most cases, `zsfc = 0.0` m.

`zflat` *real*(4)

The lowest height of the flat level. Unit is meter.

`sthopt` *integer*(4)

Option for vertical stretching.

- `0`: No stretching (uniform grid interval).
- `1`: Stretching using a cubic function.
- `2`: Stretching using a function of tanh.

`dzmin` *real*(4)

Minimum grid interval at the lowest level. Unit is meter.

`layer1,layer2` *real*(4)

In stretching, the grid interval of `dzmin` is used up to a height of `layer1`, the intervals determined by functions between heights of `layer1` and `layer2`, and the grid interval at a height of `layer2` is used above the height of `layer2`. Unit is meter.

& terrain

The namelists specify terrain topography of the model. They are used in the program of *solver*, pre-processors, and post- processors. Dependency is present between these programs.

`trnopt` *integer*(4)

Option for model terrain.

- `0`: the flat ground.
- `1`: the bell-shaped mountain.
- `2`: user specified terrain from the external data file.

`mnthgh` *real*(4), *dimension*(1 : 2)

For `trnopt = 1`, first number is height (m) of the summit of the bell-shaped mountain, and the second number is height (m) of the base of the bell-shaped mountain. The second number is usually 0.0 (m) and is possible to be omitted. The altitude of the summit is `mnthgh(1)-mnthgh(2)` .

`mntwx,mntwy` *real*(4)

For `trnopt = 1`, half width (distance from the center of a half height) of the bell-shaped mountain in the x and y directions, respectively. Unit is meter.

`mntcx,mntcy` *real*(4)

For `trnopt = 1`, The coordinates x, y of the center of the bell-shaped mountain. Unit is meter. The relationship between the coordinates and their array number `i,j` is as follows.

$x = (\mathtt{i} - 2) \times \mathtt{dx}$

$y = (\mathtt{j} - 2) \times \mathtt{dy}$

& flength

The namelists are with respect to time of calculation. They are used in *solver*, pre-processors, and post- processors. The parameter with $*$ has dependency between these programs.

`sfcast`* *character* $*$ 16

Date and time of the initial time of calculation in the Universal Time Coordinate (UTC). The format is `sfcast = 'yyyy/mm/dd hh:mm'`. For example, when the initial time is 13:00 UTC, 22 September 2007, `sfcast = '2007/09/22 13:00'`.

`stime` *real*(4)

Initial time of calculation. Unit is second. The time is measured from the time of `sfcast`. In the first calculation, `stime = 0.0`, and In a restart, `stime` is must be the restart time.

`etime` *real*(4)

Time of the termination of calculation. Unit is second. The time is also measured from the time of `sfcast`. For example, `etime` must be 5400.0 to calculate 1800.0 seconds from 3600.0 seconds.

& boundry

The namelists determine the lateral boundary condition. They are used in *solver*, pre-processors, and post- processors. The parameters with ∗ have dependency between these programs.

`wbc`* *integer*(4)

Western lateral boundary condition. Option is shown below.

`ebc`* *integer*(4)

Eastern lateral boundary condition. Option is shown below.

`sbc`* *integer*(4)

Southern lateral boundary condition. Option is shown below.

`nbc`* *integer*(4)

Northern lateral boundary condition. Option is shown below.

- `1`: periodic boundary condition.
- `2`: rigid wall boundary condition.
- `3`: zero-gradient boundary condition.
- `4`: wave-radiating boundary condition.
- `5`: wave-radiating boundary condition using vertically averaged normal velocity component as phase velocity.
- `6`: wave-radiating boundary condition using constant phase velocity and advection speed.
- `7`: wave-radiating boundary condition using constant phase velocity. [**recommended**]

`bbc` *integer*(4)

Bottom boundary condition. Option is shown below.

`tbc` *integer*(4)

Top boundary condition. Option is shown below.

- `2`: rigid wall boundary condition.
- `3`: zero-gradient boundary condition.
- `4`: wave-radiating boundary condition at the top boundary condition.

If `impopt = 1`, then these parameters are forced to be `3`.

`lbcvar` *character*(80); `lbcvar = 'xxxxxxxxxxxx'`

If `exbopt=0`, the lateral boundary can be dumped to the base state. The parameter specifies variables along the lateral boundary to be dumped. If the base state is given, the variable is dumped to the base state. If no base state is given, then, it is dumped to zero. The first 13 characters of 80 are used. Each character from the first corresponds to u and v-components of velocity, vertical velocity component, pressure, potential temperature (or temperature), water vapor mixing ratio (or relative humidity), mixing ratios of cloud water, rain, cloud ice, snow and graupel, the turbulent kinetic energy and mixing ratio of tracer $(u, v, w, p, pt, qv, qc, qr, qi, qs, qg, tke, q_{trace})$. If `exbopt`$\neq$0, this option is ignored.

- `o`: the variable is dumped to the base state or zero along the lateral boundary.
- `x`: the variable is not dumped.

`lbnews` *real*(4)

Dumping coefficient of all variables except the normal velocity components along the lateral boundary. Unit is [s^{-1}].

`lbnorm` *real*(4)

Dumping coefficient of the normal velocity components along the lateral boundary. Unit is [s^{-1}].

† `gwave` *real*(4)

Fastest gravity wave speed for constant phase speed for radiation boundary condition. Unit is [m s^{-1}].

& gpvpram

The namelists control GPV external data and specify lateral external forcing and upper sponge dumping. They are used in *solver*, pre-processors, and post-processors. The parameters with $*$ have dependency between these programs.

`gpvvar`* *character*(80); `gpvvar = 'xoxxxxx'`

The parameter specifies variables to be inputted to the model from an external data. The first 7 characters of 80 are used. Each character from the first corresponds to the vertical velocity component (w), water vapor mixing ratio (qv)(or relative humidity), mixing ratios of cloud water, rain, cloud ice, snow and graupel (qc, qr, qi, qs, qg). On the other hand, z-coordinate, horizontal velocity components, pressure, potential temperature (or temperature) are necessary and omitted form this parameter. For example, the vertical velocity component, water vapor mixing ratio (or relative humidity) and mixing ratio of rain are used (when they are available), then `extvar = 'ooxoxxx'`

o: the variable is available in the dataset.
x: the variable is not available in the dataset.

`gpvitv`* *real*(4)
Time interval of the external data. Unit is second.

`gsmopt` *integer*(4)
Option for smoothing GPV data.
0: No smoothing is performed.
1: Perform smoothing for boundary data.
2: Perform smoothing for boundary and initial data.

`gsmcnt` *integer*(4)
Iteration count for GPV data smoothing.

`gsmcoe` *real*(4)
Smoothing coefficient using in the iteration. Unit is $[s^{-1}]$.

`nggopt` *integer*(4)
Option for nudging of analysis data.
0: no nudging.
1: perform the nudging.

`nggvar` *character*(80); nggvar = 'xxxxxxxxxxx'
The parameter specifies variables to be used in nudging. The first 11 characters of 80 are used. Each character from the first corresponds to u and v-components of velocity, vertical velocity component, pressure, potential temperature (or temperature), water vapor mixing ratio (or relative humidity), mixing ratios of cloud water, rain, cloud ice, snow and graupel $(u, v, w, p, pt, qv, qc, qr, qi, qs, qg)$. For example, when all the three components of velocity, pressure and potential temperature are used for nudging, `nggvar = 'ooooxxxxxx'`. All variables besides the necessary variables must be specified by `extvar`.
o: the variable is used in nudging.
x: the variable is not used in nudging.

`nggcoe` *real*(4)
Coefficient of dumping (e-folding time of dumping) of the nudging of the analysis data. Unit is $[s^{-1}]$.

`nggdlt` *real*(4)
Time interval of nudging. Unit is second. The parameter must be divided by `dtb` and larger than or equal to `dtb`.

`nggend` *real*(4)

Time to terminate the nudging (measured from the initial). Unit is second.

`exbopt`* *integer*(4)

Option for forcing to external data at the lateral boundary. The forcing is performed with respect to the necessary variables and variables specified by `exbvar`.

- `0`: No forcing at the lateral boundary.
- `1`: Forcing to the external data at the lateral boundary is performed.
- `2`: forcing to the external data at the lateral boundary is performed and the terrain is also smoothed to be the external terrain in the lateral sponge layer.
- `11`: Forcing to the external data at the lateral boundary is performed and horizontal velocity is adjusted.
- `12`: Forcing to the external data at the lateral boundary is performed. The terrain is also smoothed to be the external terrain in the lateral sponge layer. Horizontal velocity is adjusted.

`exbvar` *character*(80); exbvar = '–x—xxxxx'

The parameter specifies variables to be externally forced. The first 11 characters of 80 are used. Each character from the first corresponds to u and v-components of velocity, vertical velocity component, pressure, potential temperature (or temperature), water vapor mixing ratio (or relative humidity), mixing ratios of cloud water, rain, cloud ice, snow and graupel $(u, v, w, p, pt, qv, qc, qr, qi, qs, qg)$.

- `o`: Both the vector and scalar variables at the outer most grids are replaced by the external data. In this option, the coefficients `exnews` and `exnorm` are not used.
- `-`: Both the vector and scalar variables along the boundary are dumped to the external data and wave-radiating lateral boundary condition is applied. The dumping coefficient of normal velocity to the boundary is `exnorm`. The dumping coefficient of all other variables is `exnews`.
- `+`: The normal velocity component of (u, v) along the outermost girds are replaced by the external data and the parallel velocity components and scalar variables are dumped to the external data with the coefficient of `exnews`.
- `x`: No external forcing is performed the boundary condition is the wave-radiating type.

`exnews` *real*(4)

Coefficient of the external forcing at the outermost lateral boundary. Unit is $[s^{-1}]$.

`exnorm` *real*(4)

Coefficient of the lateral forcing for normal velocity component at the outermost lateral boundary. Unit is [s^{-1}].

`exbwid` *integer*(4)

For `exbopt = 2` or `exbopt = 12`, the model terrain is modified to connect the coarse data terrain smoothly along the boundary within the width specified by this parameter. Unit is grid number.

`lspopt` *integer*(4)

Option for lateral sponge.

- `0`: No lateral sponge dumping.
- `1`: Lateral sponge damping to external data.
- `2`: lateral sponge damping to base state or initial values.
- `11`: Lateral sponge damping to external data with smoothing within the lateral sponge layer.
- `12`: lateral sponge damping to base state or initial values with smoothing within the lateral sponge layer.

`lspvar` *character*(80); lspvar = '++xxooxxxxxxx'

The parameter specifies variables to be dumped in the lateral sponge layer. The first 11 characters of 80 are used. Each character from the first corresponds to u and v-components of velocity, vertical velocity component, pressure, potential temperature (or temperature), water vapor mixing ratio (or relative humidity), mixing ratios of cloud water, rain, cloud ice, snow and graupel $(u, v, w, p, pt, qv, qc, qr, qi, qs, qg)$.

- `o`: The lateral sponge dumping is applied with the coefficient `lsnews` to the specified variables.
- `+`: This option is available for only (u, v). Both the normal and parallel components in the sponge layer dumped with the coefficient `lsnews` and the normal velocity component additionally dumped with the coefficient `lsnorm`.
- `-`: This option is available for only (u, v). Only the normal velocity component of (u, v) is dumped with the coefficient of `ls norm`, but the parallel component is not dumped.
- `x`: No dumping is performed.

`lspsmt`* *real*(4)

For `lspopt = 11` or `lspopt = 12`, the parameter is coefficient of smoothing within the lateral sponge layer. Unit is [s^{-1}].

`lsnews` *real*(4)

Coefficient of the dumping in the lateral sponge layer. Unit is [s^{-1}]. The forcing decrease with the distance from the lateral boundary.

`lsnorm` *real*(4)

Coefficient of the dumping of the normal velocity components in the lateral sponge layer. Unit is [s^{-1}]. The forcing decrease with the distance from the lateral boundary.

`wdnews` *integer*(4)

Grid number of the lateral sponge layer.

`wdnorm` *integer*(4)

Grid number of the lateral sponge layer for the normal velocity components.

`vspopt` *integer*(4)

Option for dumping in the top sponge layer.

- `0`: no dumping in the top sponge layer.
- `1`: dumping in the top sponge layer is performed. The variables are forced to the external grid data.
- `2`: dumping in the top sponge layer is performed. The variables are forced to the initial data (base state).

`vspgpv` *real*(4)

For `vspopt = 1`, dumping coefficient in top sponge layer of variables for which external data is given. The forcing decreases with the distance from the top of the model. Unit is [s^{-1}]. Variables which external data is not given are dumped to zero.

`vspgpv` *real*(4)

For `vspopt = 1` or `vspopt = 2`, dumping coefficient in top sponge layer. The coefficient is used for all variables in `vspopt = 1`, and for variables for which external data is not given in `vspopt = 2`. The forcing decreases with the distance from the top of the model. Unit is [s^{-1}].

`botgpv` *real*(4)

For `vspopt = 1`, the lowest height of the top sponge layer for variables for which external data is given. Unit is meter.

`botgpv` *real*(4)

For `vspopt = 1` or `vspopt = 2`, the lowest height of the top sponge layer for all variables in `vspopt = 1`, and for variables for which external data is not given in `vspopt = 2`.

& radpram

The namelists control radar data nudging. They are used in *solver* and the pre-processor *radata.exe*.

`radvar`* *character*(80); `radvar = 'xxxx'`

The parameter specifies variables to be inputted to the model from an external data. The first 4 characters of 80 are used. Each character from the first corresponds to u- and v-components of horizontal velocity, the vertical velocity component, radar echo intensity (dBZ) or mixing ratio of precipitation which is usually converted from the radar echo intensity (u, v, w, qp).

- `o`: the variable is available in the dataset.
- `x`: the variable is not available in the dataset.

`raditv`* *real*(4)

Time interval of the radar data. Unit is second.

`ngropt` *integer*(4)

Option for nudging of the radar data.

- `0`: no nudging of radar data.
- `1`: perform the nudging of radar data.

`ngrvar` *character*(80); ngrvar = 'xxxxxxxxxxx'

The parameter specifies variables to be used in radar nudging. The first 6 characters of 80 are used. Each character from the first corresponds to u and v-components of velocity, vertical velocity component, mixing ratios of rain, snow and graupel (u, v, w, qr, qs, qg).

- `o`: the variable is used in nudging.
- `x`: the variable is not used in nudging.

`ngrcoe` *real*(4)

Coefficient of nudging the radar data. Unit is $[s^{-1}]$.

`ngrdlt` *real*(4)

Time interval of nudging. Unit is second. The parameter must be divided by `dtb` and larger than or equal to `dtb`.

`ngrstr` *real*(4)

Time to start the nudging (measured from the initial). Unit is second.

`ngrend` *real*(4)

Time to terminate the nudging (measured from the initial). Unit is second.

`ngraff` *real*(4)

Time width of the nudging. The nudging is performed within twice of `ngraff` with centering at each time when data is given. The parameter must be smaller than half of `raditv`. Unit is second.

& sfcphys

The namelists determine the surface conditions and the ground model.

---

† `sfcdat`* *character*(80)

The parameter specifies the external data used for calculation of the surface process. The first 3 characters of 80 are used. Each character from the first corresponds to land use, sea surface temperature, and sea ice distribution. For example, when the land use and the land surface temperature are available, then, `sfcdat = 'oox'`.

- `o`: The variable is available and will be used.
- `x`: The variable is not used.

`sfcopt` *integer*(4)

Option for the surface process.

- `0`: No surface process is performed.
- `1`: Surface process is performed. Ground temperature and sea surface temperature are calculated and vary with time.
- `2`: Surface process is performed. Ground temperature is calculated and varies with time while sea surface temperature is constant with time.
- `11`: Surface process is performed. Calculations of ground temperature and sea surface temperature are started from the previous calculation. The re-start file of the previous calculation is defined by `prvres`
- `12`: Surface process is performed. Calculation of ground temperature is started from the previous calculation while sea surface temperature is constant with time. The re-start file of the previous calculation is defined by `prvres`

`levpbl` *integer*(4)

For `tubopt = 0`, Vertical 1-dimensional diffusion of the Meller-Yamada closure 2 is performed. The parameter specifies grid number of the planetary boundary layer. The thickness of each layer is dz and `1` $<$ `levpbl` $<$ `zdim - 2`. If For `tubopt` $\neq 0$, then, this optional process is ignored.

`levund` *integer*(4)

Number of layers of the ground and sea.

`dzgrd` *real*(4)

Thickness of the layers of the ground model. Unit is meter. The total depth of the ground model is desirable to be larger than a few meters.

`dzsea` *real*(4)

Thickness of the layers of the sea model. Unit is meter. The total depth of the sea model is desirable to be larger than a few meters.

`tgdeep` *real*(4)

Constant temperature at the bottom of the ground model. Unit is [K].

`prevres` *character*(80); `prevres='test.res000000`

Restart file name without extension of previous calculation. This file is used for `sfcopt=11` or `sfcopt=12`.

‡ `sfcdat`* *character*(80)

The parameter specifies the external data used for calculation of the surface process. The first 3 characters of 80 are used. Each character from the first corresponds to land use, sea surface temperature, and sea ice distribution. For example, when the land use and the land surface temperature are available, then, `sfcdat = 'oox'`.

- `o`: The variable is available and will be used.
- `x`: The variable is not used.

`lnduse` *integer*(4)

For `sfcdat (1)= 'x'`, ground surface condition is specified by this parameter. For every options, the area lower than 0 m in altitude is considered as a sea.

- `1`: The ground higher than 0 m in altitude is not covered with snow nor ice. The surface parameters are specified by user with the following parameters.
- `2`: The ground higher than 0 m in altitude is covered with snow. The surface parameters are specified inside of solver.
- `3`: The ground higher than 0 m in altitude is covered with ice. The surface parameters are specified inside of solver.

`gralbe` *real*(4); `gralbe=0.2`

For `sfcdat (1)= 'x'` and `lnduse=1`, a constant albedo of the ground.

`grbeta` *real*(4); `grbeta=0.5`

For `sfcdat (1)= 'x'` and `lnduse=1`, a constant evapotranspiration coefficient (moisture availability form the surface).

`grz0m` *real*(4); `grz0m=0.4`
For `sfcdat (1)= 'x'` and `lnduse=1`, a constant roughness length for velocity of the ground surface [m].

`grz0h` *real*(4); `grz0h=0.1`
For `sfcdat (1)= 'x'` and `lnduse=1`, a constant roughness length for scalar of the ground surface [m].

`grcap` *real*(4); `grcap=2.3e6`
For `sfcdat (1)= 'x'` and `lnduse=1`, a constant thermal capacity of ground [J m^{-3}].

`grnuu` *real*(4); `grnuu=7.0e-7`
For `sfcdat (1)= 'x'` and `lnduse=1`, a constant thermal diffusivity of ground [m^{-2}].

`sstcst` *real*(4); `sstcst=300.15e0`
For `sfcdat (2)= 'x'`, a constant sea surface temperature [K].

`dstopt` *real*(4); `dstopt=1`
For `sfcdat (3)= 'o'`, Option to produce of model sea ice distibution.
- `1`: Ice density weighted average distribution.
- `2`: All-or-nothing distribution.

& initype

The namelists determine the initialization of model. Used in only *solver*.

`iniopt` *integer*(4)
Option for the model initialization.
- `1`: Initial data is from the sounding data.
- `2`: From restart files.
- `3`: From 3-dim. non-uniform grid data (GPV data).
- `12`: From restart files without checking configuration of the previous run. variables.

`snddim` *integer*(4)
For `iniopt = 1`, number of levels of the sounding data.

‡ `sndhed` *integer*(4)

For `iniopt = 1`, number of comment lines in the header of the sounding data. The comment lines must be at the beginning of the sounding file.

`sndtyp` *character*(80)

For `iniopt = 1`, specify data type of the sounding. The following combinations are available except the horizontal velocity.

option	1st column	2nd column	5th column
`ppm:`	pressure [Pa],	θ [K] ,	q_v [kg kg^{-1}]
`ppm:`	pressure [Pa],	θ [K] ,	q_v [kg kg^{-1}]
`zpm:`	height [m],	θ [K] ,	q_v [kg kg^{-1}]
`ptm:`	pressure [Pa],	temperature [K] ,	q_v [kg kg^{-1}]
`ztm:`	height [m],	temperature [K] ,	q_v [kg kg^{-1}]
`ppr:`	pressure [Pa] ,	θ [K] ,	Rh [%]
`zpr:`	height [m],	θ [K] ,	Rh [%]
`ptr:`	pressure [Pa],	temperature [K] ,	Rh [%]
`ztr:`	height [m],	temperature [K] ,	Rh [%]

`zsnd0,psnd0` *real*(4)

Height [m] and pressure [Pa] of the lowest level of the sounding data, respectively. If `sndtyp(1:1) = 'p'`, then `zsnd0` is used and if `sndtyp(1:1) = 'z'`, then `psnd0` is used.

`masopt` *integer*(4)

Option of mass-conservation calculation for the initial field.

- `0`: Mass-conservation model is not performed.
- `1`: Mass-conservation model is performed.

`maseps` *real*(4)

Value of convergence in iteration of the mass-conservation model calculation.

`alpha1` *real*(4)

For `masopt = 1`, horizontal weighting coefficient of the mass-conservation calculation.

`alpha2` *real*(4)

For `masopt = 1`, vertical weighting coefficient of the mass-conservation calculation.

& gridmove

The namelists configure the horizontal displacement of the calculation domain and are used only in *solver*.

`movopt` *integer*(4)
Option for the displacement of the domain. This option is valid when `trnopt = 0`, `iniopt = 1` and `exbopt = 0`.
- `0`: No displacement of the domain.
- `1`: Displacement of the domain is switched on.

`umove,vmove` *real*(4)
Speeds of the displacement of the domain in the x and y directions, respectively. Unit is [m s^{-1}].

& ptinicon

The namelists set perturbations of the potential temperature at the initial field and are used only in *solver*.

`pt0opt` *integer*(4)
Option for perturbations of potential temperature at the initial time. This parameter is valid when `iniopt = 1`.
- `0`: No initial perturbation of potential temperature.
- `1`: One or more elliptic perturbations are placed in the x direction.
- `2`: One or more elliptic perturbations are placed in the y direction.
- `3`: Perturbation of trigonometric function in the x direction is set.
- `4`: Perturbation of trigonometric function in the y direction is set.
- `5`: Random perturbations are generated between the specified two levels.

`pt0num` *integer*(4)
Number of the elliptic perturbation of potential temperature when `pt0opt = 1` or `2`.

`ptp0` *real*(4)
Maximum potential perturbation at the initial time . Unit is Kelvin.

`pt0rx,pt0ry, pt0rz` *real*(4)
Radii of the elliptic perturbations, or half wave length of the trigonometric function or half width of the random perturbation layer in z direction (`pt0opt = 5`). Unit is meter.

`pt0cx,pt0cy, pt0cz` *real*(4)
Center of the elliptic perturbations, or the origin of the trigonometric function or middle height of the random perturbation layer in z direction (`pt0opt = 5`). Unit is meter.

`pt0ds` *real*(4)
Distance between bubbles of potential temperature in the case of `pt0opt = 1` or 2. Unit is meter.

& integrat

The namelists define the parameters of time integration such as time intervals and coefficients of the Asselin's time filter. Only in *solver*.

`dtbig` *real*(4)
Time intervals of the large time step. The terms which are not related acoustic waves are integrated in the large time step. Unit is second. These time intervals must satisfy the CFL condition.

`dtsml` *real*(4)
Time intervals of the small time step. The terms related with the acoustic waves are integrated in the small time step. Unit is second. These time intervals must satisfy the CFL condition.

`gwmopt` *integer*(4)
Option for the time integration of the terms related with the gravity waves. [`gwmopt = 0` **is recommended.**]

- `0`: They are integrated in the large time step.
- `1`: They are integrated in the small time step.

`impopt` *integer*(4)
Option for time integration scheme of the acoustic wave mode in the vertical direction. [`impopt = 1` **is recommended.**]

- `0`: Explicit scheme.
- `1`: Implicit scheme using Gaussian elimination.
- `2`: Implicit scheme using Gaussian elimination with pivot option.
- `3`: Implicit scheme using Gauss-Zaidel method.

† `advopt` *integer*(4)
Option for the advection scheme. The center difference is used in the advection scheme. The numerical smoothing is, therefore, necessary to suppress the computational instability.

- `0`: the second order center difference.
- `1`: the fourth order center difference in horizontal and the second order in vertical directions.
- `2`: the fourth order center difference in both horizontal and vertical directions.

`gsdeps` *real*(4)
For `impopt = 3`, threshold value for iteration convergence of the Gauss-Seidel method.

`weicoe` *real*(4)
Weighting coefficient of the implicit scheme.
[`weicoe = 0.6e0` **is recommended.**]

`filcoe` *real*(4)
Weighting coefficient of the Asselin's time filter.
[`filcoe = 0.1e0` **is recommended.**]

& advction

The namelist is the option of the advection scheme. Only in *solver*. This namelist is used only in the version 2.2. This namelist is included in the namelist "& integrat" in the version 2.3.

---

‡ `ad4opt` *integer*(4)
Option for the advection scheme. The center difference is used in the advection scheme. The numerical smoothing is, therefore, necessary to suppress the computational instability.

- `0`: the second order center difference.
- `1`: the fourth order center difference in horizontal and the second order in vertical directions.
- `2`: the fourth order center difference in both horizontal and vertical directions.

& smoother

Configurations of the numerical smoothing are determined in the namelists. Only in *solver*.

`smtopt` *integer*(4)
Option for the numerical smoothing which works as an artificial viscosity to suppress the numerical noise of the advection scheme. [`smtopt = 3` **is recommended.**]

- `0`: No numerical smoothing.
- `1`: The second order numerical smoothing.
- `2`: The fourth order numerical smoothing in horizontal direction.
- `3`: The fourth order numerical smoothing both in horizontal and vertical directions.
- `12`: The fourth order numerical smoothing in horizontal direction and nonlinear smoothing.
- `13`: The fourth order numerical smoothing both in horizontal and vertical directions and nonlinear smoothing.

`smhcoe` *real*(4); `smhcoe=0.0008333e0`
Coefficient of smoothing in the horizontal direction. Unit is [s^{-1}].

`smvcoe` *real*(4); `smvcoe=0.0008333e0`
Coefficient of smoothing in the vertical direction. Unit is [s^{-1}].

`nlhcoe` *real*(4)
Coefficent of nonlinear smoothing in the horizontal direction. Unit is [s^{-1}].

`nlvcoe` *real*(4)
Coefficient of nonlinear smoothing in the vertical direction. Unit is [s^{-1}].

& mapfcter

The namelist is an option for map factors. Only in *solver*.

`mfcopt` *integer*(4)
Option for map factor.
- `0`: No map factor is used.
- `1`: Map factor is used.

& coriolis

Configuration of the Coriolis force terms. Only in *solver*.

`coropt` *integer*(4)
Option for the Coriolis force terms.

0: No Coriolis force terms.
1: Horizontal components of the Coriolis force are calculated.
2: Both horizontal and vertical components of the Coriolis force are calculated.

& earthcrv

The namelist is an option for map factors. Only in *solver*.

`crvopt` *integer*(4)
Option for calculation of the earth curvature term.
0: No earth curvature term is calculated.
1: The earth curvature term is switched on.

& buoyancy

Switching the buoyancy term of pressure perturbation in the vertical component of the momentum equation. This term is related to the acoustic waves. Only in *solver*.

`buyopt` *integer*(4)
Option for the buoyancy term.
0: The buoyancy term is omitted.
1: The buoyancy term is included.

& diabatic

Switching the diabatic term in the pressure equation. Only in *solver*.

`diaopt` *integer*(4)
Option for the diabatic term.
0: The diabatic term is omitted.
1: The diabatic term is included.

& ddamping

Configuration of the divergence dumping of the pressure gradient force in the momentum equations. Only in *solver*.

`divopt` *integer*(4)

Option for the divergence dumping of the pressure gradient force, which is an artificial term to suppress the numerical instability.

- `0`: No divergence dumping is performed.
- `1`: Divergence dumping is performed.

`divndc` *real*(4)

Non-dimensional dumping coefficient. The recommended value is 0.05.

& turbulent

The subgrid-scale turbulence process is configured in the namelist. Only in *solver*.

`tubopt` *integer*(4)

Option for the sub-grid turbulence process. [`tubopt = 2` **is recommended.**]

- `0`: No turbulence process is performed.
- `1`: The Smagorinsky scheme (1 order closure).
- `2`: The 1.5 order closure with TKE.
- `3`: The 1.5 order closure with TKE and set initial mixing coefficient.

`isoopt` *integer*(4)

Option for directionality of the grid.

- `1`: Grid is directional in horizontal and vertical directions.
- `2`: Grid is non-directional in horizontal and vertical directions.

`prnum` *real*(4)

Turbulent Prandtl number ($Pr = \nu_\tau / \nu_H$). It is used only for stability check when `tubopt = 2`.

`csnum` *real*(4)

The Smagorinsky constant when `tubopt = 1`.

& cloudphy

Configuration for the cloud microphysics. Only in *solver*.

`cphopt` *integer*(4)

Option for the cloud microphysics.

- `0`: No cloud microphysics (the dry model).
- `1`: The bulk warm rain parameterization.
- `2`: The bulk cold rain parameterization.
- `3`: The bulk cold rain parameterization with solving the tendency equation of number densities of ice phase hydrometeors (ice, snow and graupel).

`nclcst` *real*(4); `nclcst=1.e8`

The constant concentration of cloud water. Unit is [m^{-3}].

`thresq` *real*(4); `thresq=1.e-12`

The Minimum threshold value of mixing ratios. Unit is [kg kg^{-1}].

& mixtrace

The tracer option is available using a tracer mixing ratio which has no source terms nor falling velocity. The time change of the mixing ratio is due to only the advection. The namelists set configuration of tracer mixing ratio in the initial field. Used only in *solver*.

`trkopt` *integer*(4)

Option for tracer capability.

- `0`: No tracer is used.
- `1`: Calculation of tracer mixing ratio is performed. The source of tracer is set only in the initial field.
- `2`: Calculation of tracer mixing ratio is performed. The source of tracer is set in the initial field and constant source of the tracer is present at all time in the initial locations.

`qt0opt` *integer*(4)

Options for configuration of tracer.

- `1`: One or more elliptic source of tracer are placed in the x direction.
- `2`: One or more elliptic source of tracer are placed in the y direction.
- `3`: Source of tracer of trigonometric function in the x direction is set.
- `4`: Source of tracer of trigonometric function in the y direction is set.
- `5`: Random source of tracer are generated between the specified two levels.

`qt0num` *integer*(4)

Number of the elliptic source of tracer when `qt0opt = 1` or `2`.

`qt0` *real*(4)

Magnitude or amplitude of source of tracer at the initial time . Unit is [kg kg^{-1}].

`qt0rx,qt0ry, qt0rz` *real*(4)
Radii of the elliptic source of tracer, or half wave length of the trigonometric function or half width of the random source of tracer in z direction (`qt0opt = 5`). Unit is meter.

`qt0cx,qt0cy, qt0cz` *real*(4)
Center of the elliptic source of tracer, or the origin of the trigonometric function or middle height of the random source of tracer in z direction (`qt0opt = 5`). Unit is meter.

`qt0ds` *real*(4)
Distance between bubbles of source of tracer in the case of `qt0opt = 1` or `2`. Unit is meter.

`qtdt` *real*(4)
Intensity of tracer source in the case of `trkopt = 2`. Unit is [s^{-1}].

& outfomat

The namelists determine the format and time intervals of outputs. Only in *solver*.

`dmpfmt` *integer*(4)
Option for the output format of the history and geographic files.
- `1`: ASCII text format.
- `2`: Binary data of the unformatted direct access.

`dmpcmp` *integer*(4)
Option of the vertical coordinate of the history file. When `dmpcmp = 2` or 1, An undefined value of -1.0×10^{35} are outputted below the surface and above the top of the domain.

1 The vertical coordinate of history files is the terrain following type(z^* (ζ)). Horizontal velocity components (u, v) are x and y components of the coordinate. In this option, all grids are not the undefined value. For nesting calculation, this option should be used for coarse grid calculation.

2 The output variables are interpolated at horizontal levels with an interval of `dz`. Horizontal velocity components (u, v) are x and y components of the coordinate.

3 The output variables are interporlated at horizontal levels defined by the stretching functions. Horizontal velocity components (u, v) are x and y components of the coordinate.

11 Same as `dmpcmp=1`, but horizontal velocity components (u, v) are zonal and meridional components.

12 Same as `dmpcmp=2`, but horizontal velocity components (u, v) are zonal and meridional components.

13 Same as `dmpcmp=3`, but horizontal velocity components (u, v) are zonal and meridional components.

`dmpvar` *character*(80); dmpvar = '–o—oooxoox'

The parameter specifies variables to be outputted to the history files. The first 13 characters of 80 are used. Each character from the first corresponds to

`dmpvar(1:1)='-'/'o'/'x'`:	x- or zonal component of horizontal velcity.
`dmpvar(2:2)='-'/'o'/'x'`:	y- or meridioanl component of horizontal velcity.
`dmpvar(3:3)='o'/'x'`:	vertical component of velocity.
`dmpvar(4:4)='-'/'o'/'x'`:	pressure.
`dmpvar(5:5)='-'/'o'/'x'`:	potential temperature.
`dmpvar(6:6)='-'/'o'/'x'`:	mixing ratio of water vapor.
`dmpvar(7:7)='o'/'x'`:	mixing ratios of hydrometeors.
`dmpvar(8:8)='o'/'x'`:	concentrations of ice, snow and graupel.
`dmpvar(9:9)='o'/'x'`:	TKE (turbulent kinetic energy).
`dmpvar(10:10)='o'/'x'`:	mixing ratio of tracer.
`dmpvar(11:11)='o'/'x'`:	surface monitor variables.
`dmpvar(12:12)='o'/'x'`:	precipitation at the surface.
`dmpvar(13:13)='o'/'x'`:	z physical coordinates.

The options are

`o`: the variable is outputted to history files.

`-`: For (u, v, qv), the variable is outputted to history files while base state (initial state) of these variables are not outputted. For pressure and potential temperature, sum of base state and perturbation is outputted.

`x`: the variable is not outputted.

`dmpitv` *real*(4)

Time interval to be outputted the history-file outputs. Unit is second. The interval is counted from the initial time even the calculation is started from a restart file. For example, if `dmpitv=300.0` seconds and restarted from 450.0 sec, the first output time will be 600.0 second from the initial time.

`resopt` *integer*(4)

Option for restart files.

- `0`: no restart files are issued.
- `1`: restart files are outputted at the specified time interval.

`resitv` *real*(4)

Time interval of the restart-file outputs. Unit is second. The time interval is also counted from the initial time even if started from a restart file.

`mxnopt` *integer*(4)

Option for log file.

- `0`: no log file is issued.
- `1`: log file is outputted at the specified time interval.

`mxnvar` *character*(80); mxnvar = 'ooooooooo'

The parameter specifies variables to be logged to the log file (the standard output). The first 10 characters of 80 are used. Each character from the first corresponds to u, v, w, p, θ, qv, mixing ratios of hydrometeors, concentrations of ice, snow and graupel, TKE and mixing ratio of tracer.

- `o`: maximum and minimum values are logged.
- `x`: no log of the variable is issued.

`mxnitv` *real*(4)

Time interval to output maximum and minimum of prognostic variables to the standard output. Unit is second. The time interval is also counted from the initial time even if started from a restart file.

4.1.3 Region for the pre-processor, *gridata.exe*

The pre-processor "*gridata.exe*" converts grid point value (GPV) data of the data grid to initial and boundary data of the model grid in 3-dimension. This region specifies the grid system of the GPV data and options of interpolation.

& project_grd

The namelists describe the coordinate system of a data grid. The parameters are used in only *gridata.exe*.

`mpopt_grd` *integer*(4)

Option for the coordinate of the external data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_grd` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

- `1`: the origin of coordinate in the northern hemisphere.
- `-1`: the origin of coordinate in the southern hemisphere.

`tlat1_grd`, `tlat2_grd` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_grd` and `tlat2_grd`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_grd` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_grd

The namelists describe grid intervals and the reference points. They are used only in *gridata.exe*.

`xdim_grd`, `ydim_grd` `zdim_grd` *integer*(4)

Dimensions of external data in the x, y and z directions, respectively. The vertical dimension includes the surface.

`dx_grd`, `dy_grd` *real*(4)

Grid intervals of the external data in the x and y directions, respectively. Unit is degree if `mpopt_grd = 0, 10` and otherwise meter. The vertical levels should be included variables of the external data.

`ulat_grd`, `ulon_grd` *real*(4)

Longitude and latitude of a reference point of the external data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of the external data (`riu_grd,rju_grd`) to determine the region in *gridata.exe*.

`riu_grd`, `rju_grd` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_grd`, `ulon_grd`).

& datconf_grd

The namelists determine the scheme to interpolate the external GPV data to the model grid and indicate properties of the GPV data.

`intopt_grd` *integer*(4)

Option for scheme of horizontal interpolation of the external GPV data to the model grid. The vertical interpolation is always the linear interpolation.

- `1`: The linear interpolation.
- `2`: The interpolation using the parabolic function.

`rotopt_grd` *integer*(4)

This name-list indicates property of horizontal wind of the GPV data.

- `0`: The u and v components of the GPV horizontal velocity are zonal and meridional components, respectively.
- `1`: The u and v components of the GPV horizontal velocity are x and y components of the projection specified by the namelist `mpopt_grd`, respectively.

`datype_grd` *character*(80)

The name-list indicates data type of variables in the external GPV data. The following combinations are available.

- `tm`: temperature [K] and water vapor mixing ratio [kg kg^{-1}]
- `pm`: potential temperature [K] and water vapor mixing ratio [kg kg^{-1}]
- `tr`: temperature [K] and relative humidity [%]
- `pr`: potential temperature [K] and relative humidity [%]

`refsfc_grd` *integer*(4)

The name-list indicates data property of the first level of the GPV data. This option is related to the vertical interpolation. If `refsfc_grd =1`, the first level of `zdim_grd` is considered as the surface. No extrapolation is performed vertically. If `refsfc_grd =0`, the model grids lower than the first level of the data grid are extrapolated.

- `0`: The first level of the data grid is not a surface data and the model grids lower than the first data level is extrapolated.
- `1`: The first level of the data grid is considered as the surface and all model grids are determined by interpolation.

`extvar_grd` *character*(80); `extvar_grd = 'xxxxxxx'`

For `refsfc_grd=0`, the parameter specifies extrapolation method. The first 7 characters of 80 are used. Each character from the first corresponds to u, v, w, p, θ, qv, mixing ratios of hydrometeors.

- `o`: Extrapolation is performed using the first and second levels of the GPV data grid.
- `x`: No extrapolation is performed but the data of the first GPV level is copied to all model grids lower the first GPV level.

4.1.4 Region for the pre-processor, *radata.exe*

The pre-processor "*radata.exe*" converts radar data from the data grid to the model grid in 3-dimension. This region specifies the grid system of radar data and options of interpolation.

& project_rad

The namelists describe the coordinate system of a data grid. The parameters are used in only *radata.exe*.

`mpopt_rad` *integer*(4)

Option for the coordinate of radar data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_rad` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

1: the origin of coordinate in the northern hemisphere.
-1: the origin of coordinate in the southern hemisphere.

`tlat1_rad`, `tlat2_rad` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_rad` and `tlat2_rad`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_rad` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_rad

The namelists describe grid intervals and the reference points.

`xdim_rad`, `ydim_rad` *integer*(4)

`zdim_rad` Dimensions of radar data in the x, y and z directions, respectively.

`dx_rad`, `dy_rad` *real*(4)

Grid intervals of radar data in the x and y directions, respectively. Unit is degree if `mpopt_rad = 0, 10` and otherwise meter.

`ulat_rad`, `ulon_rad` *real*(4)

Longitude and latitude of a reference point of radar data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of the external data (`riu_rad,rju_rad`) to determine the region in *radata.exe*.

`riu_rad`, `rju_rad` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_rad`, `ulon_rad`).

& datconf_rad

The name-lists determine the scheme to interpolate radar data to the model grid and indicate properties of radar data.

`rotopt_rad` *integer*(4)

This name-list indicates property of horizontal wind of radar data.

- `0`: The u and v components of the horizontal velocity are zonal and meridional components, respectively.
- `1`: The u and v components of the horizontal velocity are x and y components of the projection specified by the namelist `mpopt_rad`, respectively.

`datype_rad` *character*(80)

The name-list indicates data type of variables in radar data. The following combinations are available.

- `m`: mixing ratio [kg kg^{-1}]
- `r`: radar echo intensity [dBZ]

`radcoe_rad` *real*(4)

Coefficient of the Z-M relationship to convert the radar echo intensity to mixing ratio of rain: $Z = BM^{\beta}$. In this equation, `radcoe_rad` is the coefficient B.

`radpwr_rad` *real*(4)

The exponent of the Z-M relationship to convert the radar echo intensity to mixing ratio of rain: $Z = BM^{\beta}$. In this equation, `radpwr_rad` is the exponent β.

4.1.5 Region for the pre-processor, *terrain.exe*

The pre-processor "*terrain.exe*" converts terrain data of the data grid to the model grid in 2-dimension. This region specifies the grid system of terrain data and options of interpolation.

& project_trn

The name-lists describe the coordinate system of a data grid. The parameters are used in only *terrain.exe*.

`mpopt_trn` *integer*(4)

Option for the coordinate of terrain data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_trn` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

- `1`: the origin of coordinate in the northern hemisphere.
- `-1`: the origin of coordinate in the southern hemisphere.

`tlat1_trn`, `tlat2_trn` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_trn` and `tlat2_trn`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_trn` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_trn

The namelists describe grid intervals and the reference points. They are used only in *terrain.exe*.

`xdim_trn`, `ydim_trn` *integer*(4)

Dimensions of external data in the x, y directions, respectively.

`dx_trn`, `dy_trn` *real*(4)

Grid intervals of the external data in the x and y directions, respectively. Unit is degree if `mpopt_trn = 0, 10`, otherwise, meter.

`ulat_trn`, `ulon_trn` *real*(4)

Longitude and latitude of a reference point of terrain data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of terrain data (`riu_trn,rju_trn`) to determine the region in *terrain.exe*.

`riu_trn`, `rju_trn` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_trn`, `ulon_trn`).

& datconf_trn

The namelists determine the scheme to interpolate terrain data to the model grid.

`intopt_trn` *integer*(4)

Option for scheme of horizontal interpolation of terrain data to the model grid.

- `1`: The linear interpolation.
- `2`: The interpolation using the parabolic function.

4.1.6 Region for the pre-processor, *surface.exe*

The pre-processor "*surface.exe*" converts surface data from the data-grid system to the model-grid system in 2-dimension. CReSS enable to use 3 types of surface data: land use data, sea surface temperature data, and snow and ice distribution data. The region of the configuration file for *surface.exe* includes name-list parameters for these data.

& project_lnd

The name-lists describe the coordinate system of a data grid. The parameters are used only in *surface.exe*.

`mpopt_lnd` *integer*(4)

Option for the coordinate of land-use data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_lnd` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

`1`: the origin of coordinate in the northern hemisphere.

`-1`: the origin of coordinate in the southern hemisphere.

`tlat1_lnd`, `tlat2_lnd` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_lnd` and `tlat2_lnd`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_lnd` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_lnd

The namelists describe grid intervals and the reference points. They are used only in *surface.exe*.

`xdim_lnd`, `ydim_lnd` *integer*(4)

Dimensions of external data in the x, y directions, respectively.

`dx_lnd`, `dy_lnd` *real*(4)

Grid intervals of the external data in the x and y directions, respectively. Unit is degree if `mpopt_lnd = 0, 10`, otherwise, meter.

`ulat_lnd`, `ulon_lnd` *real*(4)

Longitude and latitude of a reference point of surface data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of surface data (`riu_lnd,rju_lnd`) to determine the region in *surface.exe*.

`riu_lnd`, `rju_lnd` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_lnd`, `ulon_lnd`).

& datconf_lnd

The namelists determine the scheme to interpolate surface data from data-grid system to the model grid system, and define parameters of surface condition.

`intopt_lnd` *integer*(4)

Option for scheme of horizontal interpolation of land use data to the model grid.

- `0`: A model grid point gets values of nearest data points. No interpolation is performed. **[recommended]**
- `1`: The linear interpolation of land use data is performed.

`numctg_lnd` *integer*(4)

Number of land-use categories.

`lnduse_lnd` *integer*(4), *dimension*(1 : 100)

`lnduse_lnd = -1, 11, 12, 13, 14 . . .`

User specified values of land use. The number of values is the number of categories `numctg_lnd`.

`albe_lnd` *real*(4), *dimension*(1 : 100)

`albe_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of abledo corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

`beta_lnd` *real*(4), *dimension*(1 : 100)

`beta_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of evapotranspiration efficiency corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

`z0m_lnd` *real*(4), *dimension*(1 : 100)

`z0m_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of roughness length ([m]) for velocity corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

`z0h_lnd` *real*(4), *dimension*(1 : 100)

`z0h_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of roughness length ([m]) for scalar variables (heat and moisture) corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

`cap_lnd` *real*(4), *dimension*(1 : 100)

`cap_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of thermal capacity of ground ([K J m^{-3}]) corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

`nuu_lnd` *real*(4), *dimension*(1 : 100)

`nuu_lnd = 0.e0, 0.e0, 0.e0, 0.e0, 0.e0 . . .`

Values of thermal diffusivity of ground ([m^2 s^{-1}]) corresponding to the parameter `lnduse_lnd`. The number of values is the number of categories `numctg_lnd`.

& project_sst

The name-lists describe the coordinate system of a data grid. The parameters are used only in *surface.exe*.

`mpopt_sst` *integer*(4)

Option for the coordinate of SST data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_sst` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

- `1`: the origin of coordinate in the northern hemisphere.
- `-1`: the origin of coordinate in the southern hemisphere.

`tlat1_sst`, `tlat2_sst` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_sst` and `tlat2_sst`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_sst` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_sst

The namelists describe grid intervals and the reference points. They are used only in *surface.exe*.

`xdim_sst`, `ydim_sst` *integer*(4)

Dimensions of external data in the x, y directions, respectively.

`dx_sst`, `dy_sst` *real*(4)

Grid intervals of the external data in the x and y directions, respectively. Unit is degree if `mpopt_sst = 0, 10`, otherwise, meter.

`ulat_sst`, `ulon_sst` *real*(4)

Longitude and latitude of a reference point of surface data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of surface data (`riu_sst,rju_sst`) to determine the region in *surface.exe*.

`riu_sst`, `rju_sst` *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_sst`, `ulon_sst`).

& project_ice

The name-lists describe the coordinate system of a data grid. The parameters are used only in *surface.exe*.

`mpopt_ice` *integer*(4)

Option for the coordinate of ice and snow distribution data grid.

- `0`: latitude and longitude coordinates
- `1`: The polar stereo graphic projection
- `2`: The Lambert conformal projection
- `3`: The Mercator projection
- `4`: No map projection
- `10`: The global latitude and longitude coordinates
- `13`: The cyclic Mercator projection around the globe

`nspol_ice` *integer*(4)

Switch to define origin of the coordinate in the northern or southern hemisphere.

- `1`: the origin of coordinate in the northern hemisphere.
- `-1`: the origin of coordinate in the southern hemisphere.

`tlat1_ice`, `tlat2_ice` *real*(4)

The true latitudes of map projections. No deformation due to projection occurs at the latitudes. The unit is degree [°] and they are negative in the southern hemisphere. If the Lambert conformal projection is used, two true latitudes (`tlat1_ice` and `tlat2_ice`) are necessary. If no map projection is used, they are not necessary to be specified.

`tlon_ice` *real*(4)

The true longitude of coordinate. The unit is degree [°] and degree in the West Longitude is negative.

& gridset_ice

The namelists describe grid intervals and the reference points. They are used only in *surface.exe*.

`xdim_ice`, `ydim_ice` *integer*(4)

Dimensions of external data in the x, y directions, respectively.

`dx_ice`, `dy_ice` *real*(4)

Grid intervals of the external data in the x and y directions, respectively. Unit is degree if `mpopt_ice = 0, 10`, otherwise, meter.

`ulat_ice`, `ulon_ice` *real*(4)

Longitude and latitude of a reference point of surface data. Unit is degree and they are negative in the southern hemisphere and/or the West Longitude. The point is corresponded to the real number of surface data (`riu_ice,rju_ice`) to determine the region in *surface.exe*.

`riu_ice`, `rju_ice` — *real*(4)

The real number of the reference point which corresponds to the point determined by the reference longitude and latitude (`ulat_ice`, `ulon_ice`).

4.1.7 Region for the post-processor, *unite.exe*

The post-processor "*unite.exe*" is used for parallel version. The output files of the parallel version are separately created. The post-processor "*unite.exe*" unites all the output files.

& uniconf_uni

The name-lists control the files to be united. The parameters are used only in *unite.exe*.

`fltyp_uni` — *character*(80)

Control flag of files to be processed.

- `dmp`: History-dump files are united.
- `geo`: Geography files are united
- † `all`: Both History-dump files and Geography files are united

`flitv_uni` — *real*(4)

Time interval to unite history-dump files. Unit is second.

`rmopt_uni` — *integer*(4)

Control flag whether separately outputted files are deleted or remain.

- `0`: Separately outputted files remain.
- `1`: Separately outputted files are deleted after they are united. [**recommended for saving storage**]

4.1.8 Region for the post-processor, *rstruct.exe*

The post-processor "*rstruct.exe*" rearranges restart files from the original processing element number to another processing element number.

& rstconf_rst

The name-lists control dimension and restart files to be rearranged. The parameters are used only in *rstruct.exe*.

`xpedim_rst`, `ypedim_rst` *integer*(4)

Numbers of decomposition of rearranged restart files in x and y directions, respectively. They are numbers of processing elements which is used for restart computation.

`flitv_rst` *real*(4)

Time interval of the original restart files. Unit is second.

`rmopt_rst` *integer*(4)

Control flag whether original restart files are deleted or remain.

- `0`: Original restart files remain.
- `1`: Original restart are deleted after they are rearrenged. [**recommended for saving storage**]

4.2 Data flow in CReSS

4.2.1 Outline of data flow

The outline of the data flow in CReSS is shown in Fig.4.1. The figure includes all types of data flows. According to the configuration, some of programs and data depicted in the figure are not necessary. Each case of configuration will be described in the following subsections.

The italic character part of file names in Fig.4.1 are summarized in the following table.

exprim	The experimental name which is defined by the parameter `exprim`. The beginning part of all input and output files is the name.
xxxxxx	Elapsed time from the initial (second). For example, if the elapsed time is 1200.0 seconds, then *xxxxxx* = `001200`.

These file names are used in the serial version of CReSS. In the parallel version, the number of processing element (CPU or nodes) *yyyy* is appended to the file names as `.pe`*yyyy*.

Details of the data format will be described in Subsection 4.3.

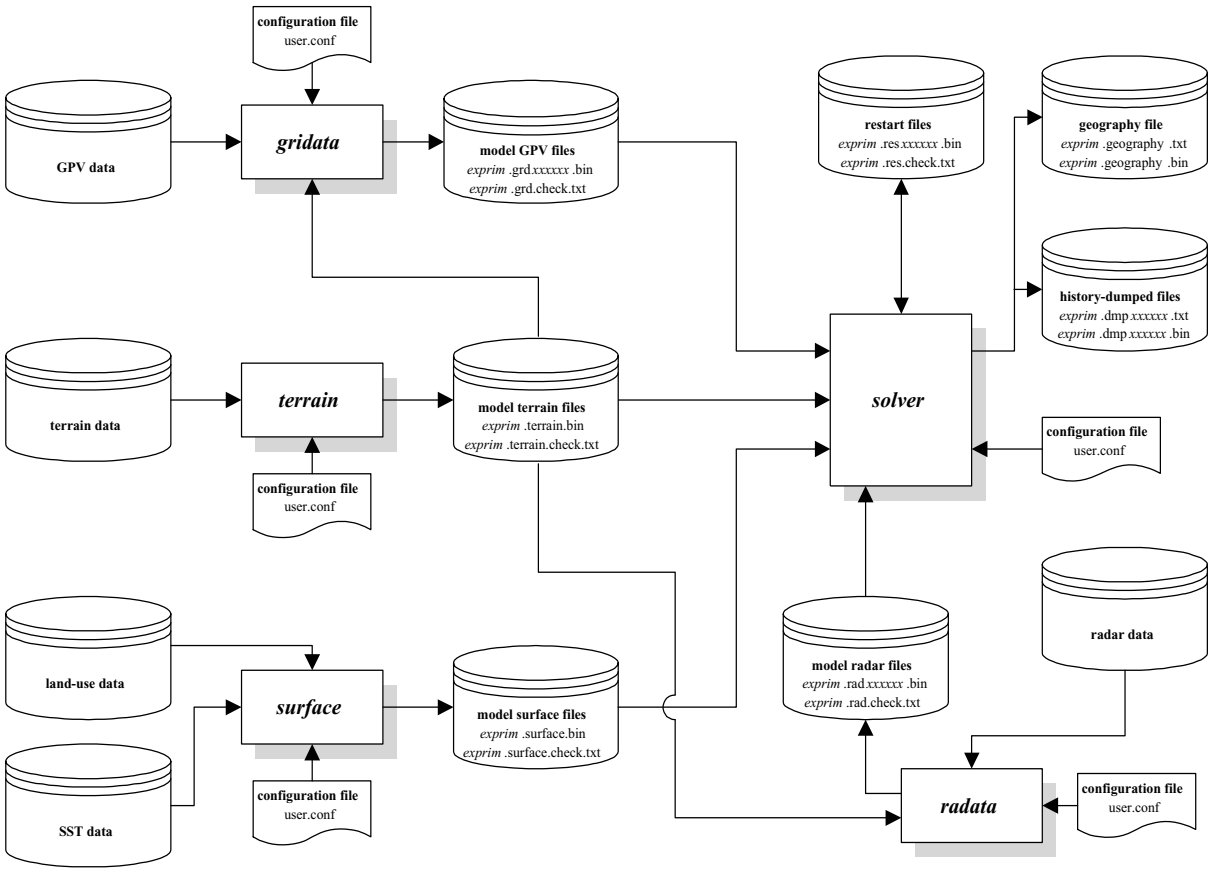

Figure 4.1. The outline of data flow in CReSS.

The data files in Fig.4.1 are summarized in the following table.

file name	description
exprim.terrain.bin	Altitude of the model topography. (unformatted direct access binary file).
exprim.terrain.check.txt	Parameter file to check the configuration of the above *exprim*.terrain.bin. (formatted text file).
exprim.grd*xxxxxx*.bin	GPV data coordinated at the model grid. Time series of the file is necessary when the nudging is performed. If nudging is not performed, the data file is used at the initial time. (unformatted direct access binary file).
exprim.grd.check.txt	Parameter file to check the configuration of the above *exprim*.grd*xxxxxx*.bin (formatted text file).
exprim.sounding.txt	One-dimensional sounding data to provide the horizontally uniform initial field. (formatted text file).
exprim.dmp*xxxxxx*.txt or .bin	History dumped file. The format is optional: formatted text file or unformatted direct access binary file.
exprim.geography.txt or .bin	Geographic information of the domain. The format is optional: formatted text file or unformatted direct access binary file.
exprim.res*xxxxxx*.bin	File to restart. (unformatted direct access binary file).
exprim.res.check.txt	Parameter file to check the configuration of the above *exprim*.res*xxxxxx*.bin. (formatted text file).

4.2.2 Data flows of each configuration

Using sounding data

(1) In the case of no terrain data

The configuration in `user.conf` is summarized in the following table and the data flow is shown in the figure.

`iniopt = 1`	Initial data is provided by the sounding data.
`trnopt = 0`	No external terrain data is used.
`exbopt = 0`	No external boundary forcing is performed.

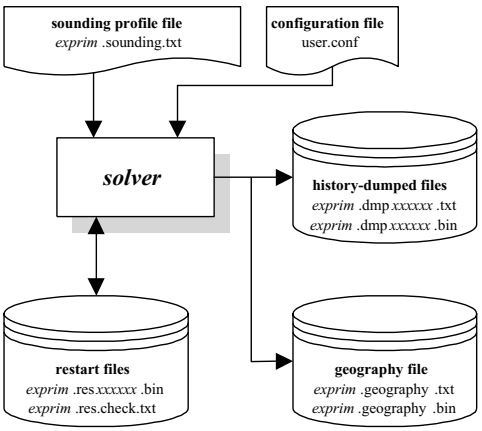

Figure 4.2. The outline of data flow in CReSS using a sounding data.

(2) Using the external terrain data

The Configuration in `user.conf` and the data flow are as follows.

`iniopt = 1`	Initial data is provided by the sounding data.
`trnopt = 2`	External terrain data is used for model topography.
`exbopt = 0`	No external boundary forcing is performed.

Using the 3-dimensional GPV data

(1) No external lateral boundary data and no terrain data

The Configuration in `user.conf` and the data flow are as follows. (This configuration will not generally used.)

`iniopt = 3`	The initial data is a 3-dimensional GPV. Time series of the data is necessary if the nudging is performed.
`trnopt = 0`	No external terrain data is used.
`exbopt = 0`	No external boundary forcing is performed.

(2) A lateral boundary data is used while no external terrain data

The Configuration in `user.conf` and the data flow are as follows. (This configuration will not generally used.)

`iniopt = 3`	The initial data is a 3-dimensional GPV. Time series of the data is necessary if the nudging is performed.
`trnopt = 0`	No external terrain data is used.
`exbopt = 1`	External boundary forcing is performed.

(3) An external terrain data is used while no external lateral boundary data

The Configuration in `user.conf` and the data flow are as follows.

`iniopt = 3`	The initial data is a 3-dimensional GPV. Time series of the data is necessary if the nudging is performed.
`trnopt = 2`	External terrain data is used.
`exbopt = 0`	No external boundary forcing is performed.

(4) Using both external lateral boundary data and terrain data

The Configuration in `user.conf` and the data flow are as follows.

`iniopt = 3`	The initial data is a 3-dimensional GPV. Time series of the data is necessary if the nudging is performed.
`trnopt = 2`	External terrain data is used.(Boundary smoothing is performed optionally.)
`exbopt = 1`	External lateral boundary forcing is performed.

Restarting from restart file

The Configuration in `user.conf` and the data flow are as follows.

`iniopt = 2`	Begin from the restart file. Time series of GPV is necessary if nudging is performed.
`trnopt =` *anything*	Terrain condition is already determined in the previous run. (No data is necessary.)
`exbopt =` *anything*	External lateral boundary data is necessary if the external lateral forcing is performed.

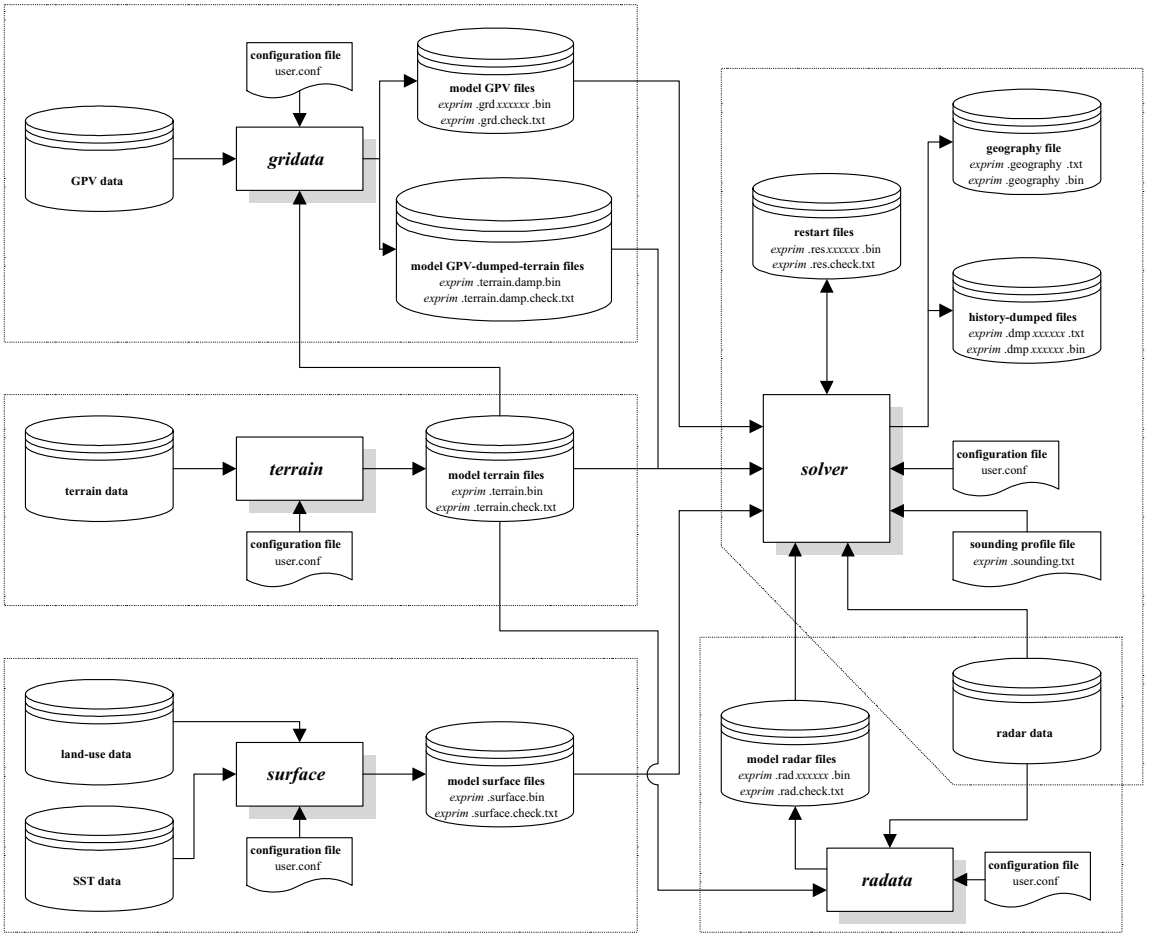

Figure 4.3. The outline of data flow in the single version of CReSS.

4.3 Formats of I/O data

In this section, formats of I/O data which controlled by users will be described: sounding data (*exprim*`.sounding.txt`), history dumped data (*exprim*`.dmp`*xxxxxx*`.txt` or *exprim*`.dmp`*xxxxxx*`.bin`) and geography data (*exprim*`.geography.txt` or *exprim*`.geography.bin`).

4.3.1 Format of the sounding data

The sounding data is a text data composed of five columns of variables. The data types of the first, second and fifth columns are defined by users. All types of combination are allowed. Therefore, there are eight combinations of variables.

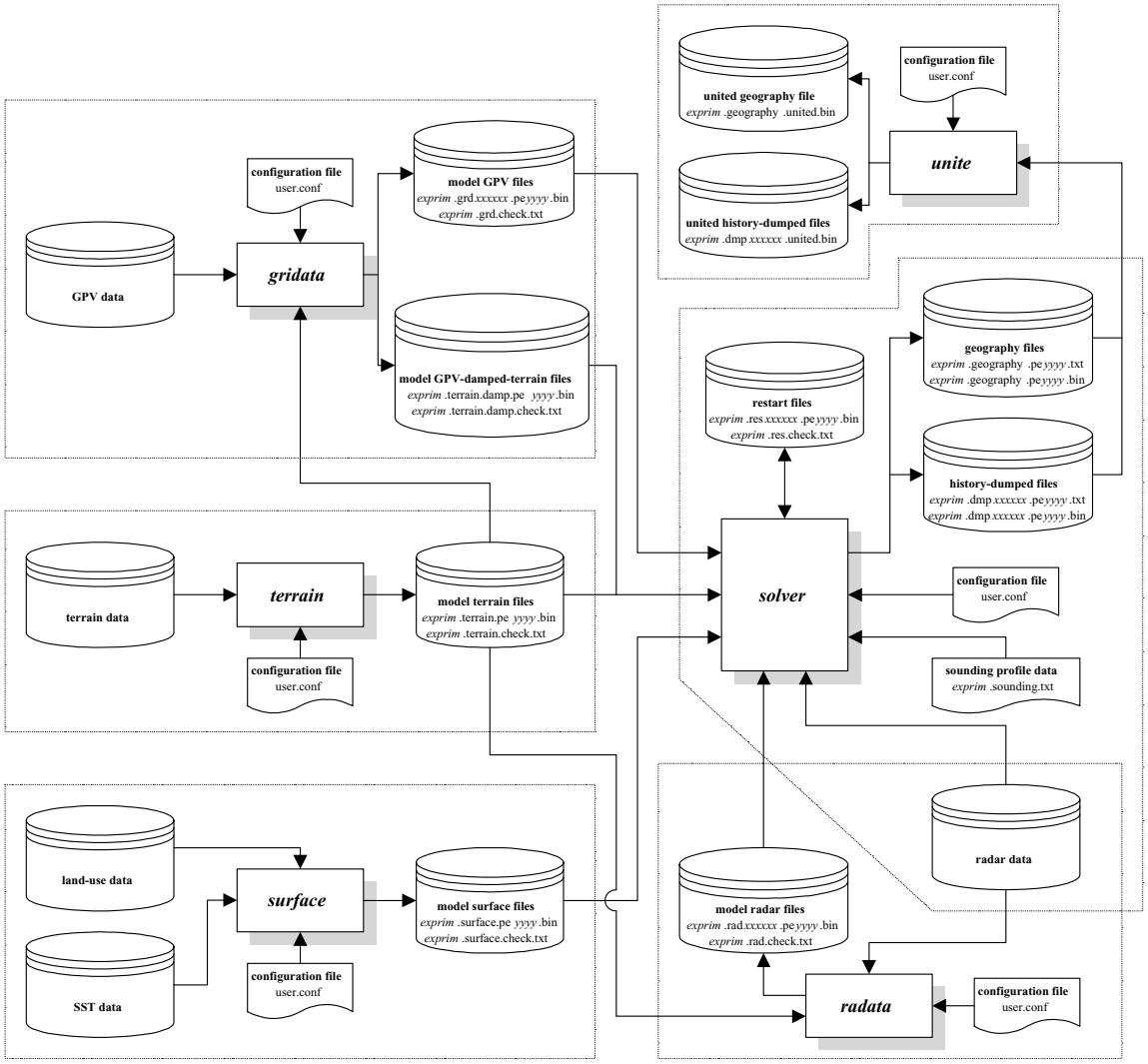

Figure 4.4. The outline of data flow in the parallel version of CReSS.

First column	height [m] or pressure [Pa]
Second column	temperature [K] or potential temperature [K]
Third column	x-component of horizontal velocity [m s^{-1}]
fourth column	y-component of horizontal velocity [m s^{-1}]
fifth column	water vapor mixing ratio [kg kg^{-1}] or relative humidity [%]

This is an example of the sounding file. It is easy to make a sounding file following the example. Some samples are found in `Form` directory of CReSS.

```
#################################################################################
#
#     One dimensional sounding input file, sounding.txt.cats.eye.form
#
# This is the cats eye simulation data.
#
#     Author  : SAKAKIBARA Atsushi
#     Date    : 1999/07/23
# Modification : 1999/07/28
#                  : 1999/11/19
#
#     First  column: height [m]                                                 #
#     Second column: tempereture [K]                                            #
#     Third  column: x components of velocity [m/s]                             #
#     Fourth column: y components of velocity [m/s]                             #
#     Fifth  column: water vapor relative humidity [%]                          #
#                                                                               #
#################################################################################
   780.e0 300.200e0  8.0000e0 0.e0 0.e0
   720.e0 300.200e0  7.9999e0 0.e0 0.e0
   660.e0 300.200e0  7.9993e0 0.e0 0.e0
                   :
                   :
   120.e0 299.800e0 -7.9946e0 0.e0 0.e0
    60.e0 299.800e0 -7.9993e0 0.e0 0.e0
     0.e0 299.800e0 -7.9999e0 0.e0 0.e0
```

4.3.2 Formats of the history dumped file and the geographic information file

Formatted text data or unformatted binary data is optionally chosen for the history dumped data and the geographic information data.

In the case of the formatted text data (`dmpfmt = 1`), a variable will be outputted as follows.

```
      do xxx k=2,nk-2
        write(ionum,*,err=errnum) (variable(i,j,k),i=2,nx-2,j=2,ny-2)
  xxx continue
```

If the unformatted binary data is selected (`dmpfmt = 2`), a variable will be outputted as follows.

```
      do xxx k=2,nk-2
        recnum=recnum+1
        write(ionum,rec=recnum,err=errnum)
     .        (variable(i,j,k),i=2,nx-2,j=2,ny-2)
xxx   continue
```

(refer to `outdmp3d.f` and `outdmp2d.f`). where `nx,ny,nz` are numbers of the grid points in the x, y and z directions, respectively. The number of `nz` is unity in the geographic information file.

The region of the output data is indicated by the thick lines in Fig.4.5. The data points are indicated by the crosses which is the scalar points. The vector variables defined at the closed circles are interpolated to the scalar points. The dimensions of the x, y and z directions are respectively smaller than those of the model grid by three.

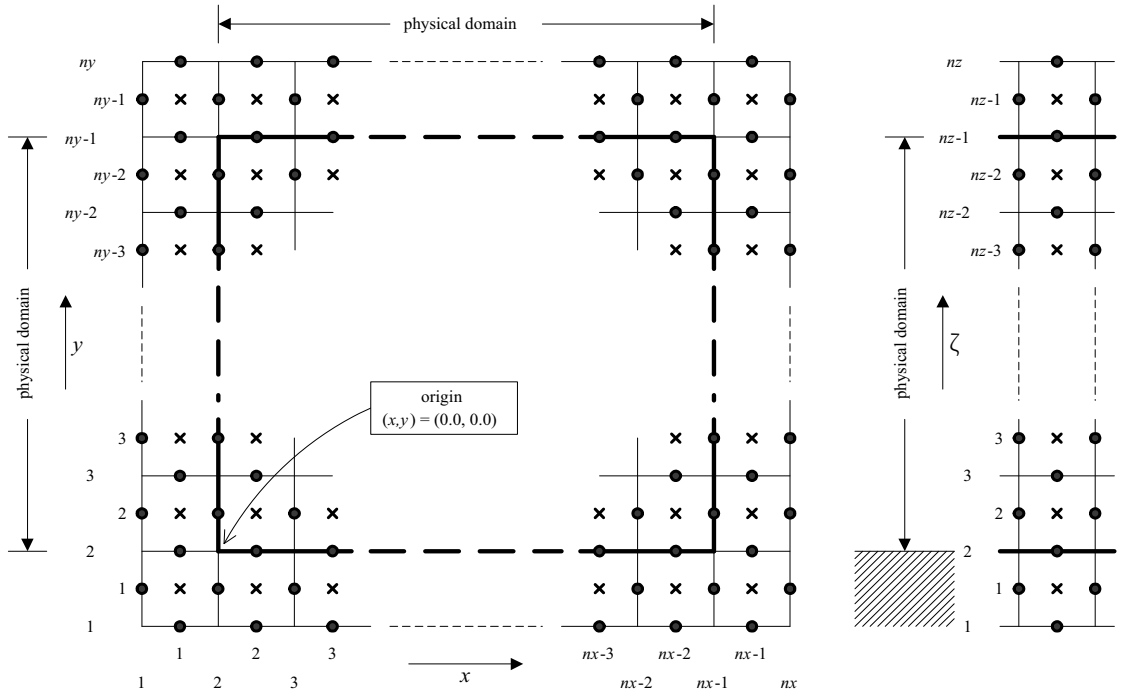

Figure 4.5. Setting of variables in the calculation domain. The closed circles are points of vector variables and the crosses are of scalar variables.

Options for vertical grid of the output data are shown in Fig.4.6.

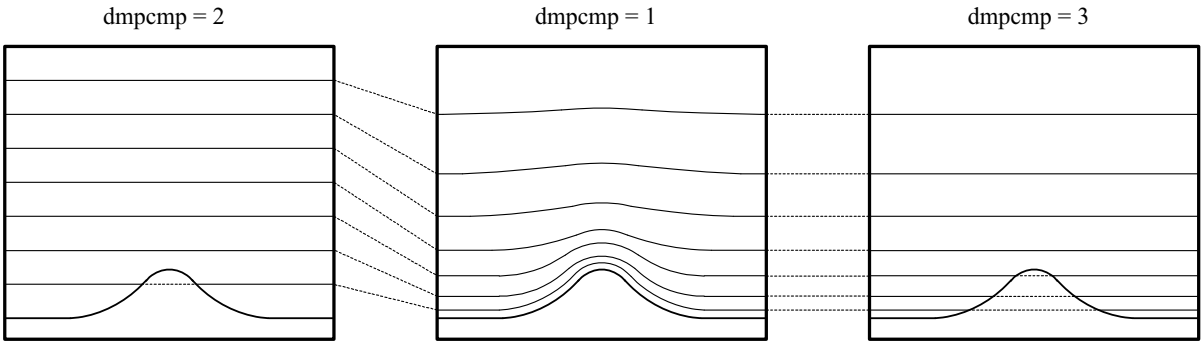

Figure 4.6. Vertical coordinates of the history dumped data.

Output variables in the geographic information data and the history dumped data are summarized in the following tables. The types of output variables of the history dumped data are depend on the configurations.

Geographic information file		
ht	altitude of terrain	m
lat	latitude	°
lon	longitude	°
fc	Coriolis parameter	
mf	map factor	
lu	land use	

History dumped file		
u	x-component of the horizontal velocity	$\mathrm{m\ s^{-1}}$
v	y-component of the horizontal velocity	$\mathrm{m\ s^{-1}}$
w	vertical component of velocity	$\mathrm{m\ s^{-1}}$
pbar	basic state of pressure	Pa
pp	perturbation pressure	Pa
ptbar	basic state of potential temperature	K
ptp	perturbation potential temperature	K
qv	mixing ratio of water vapor	$\mathrm{kg\ kg^{-1}}$
qc	mixing ratio of cloud water	$\mathrm{kg\ kg^{-1}}$
qr	mixing ratio of rain water	$\mathrm{kg\ kg^{-1}}$
qi	mixing ratio of cloud ice	$\mathrm{kg\ kg^{-1}}$
qs	mixing ratio of snow	$\mathrm{kg\ kg^{-1}}$
qg	mixing ratio of graupel	$\mathrm{kg\ kg^{-1}}$
nci	number density of cloud ice	$\mathrm{m^{-3}}$
ncs	number density of snow	$\mathrm{m^{-3}}$
ncg	number density of graupel	$\mathrm{m^{-3}}$
zph	height of grid points	m
prr	rainfall intensity	$\mathrm{m\ s^{-1}}$
par	total rainfall	$\mathrm{m\,s^{-1}}$, m
prs	snow intensity	$\mathrm{m\ s^{-1}}$
pas	total snowfall	$\mathrm{m\,s^{-1}}$, m
prg	graupel intensity	$\mathrm{m\ s^{-1}}$
pag	total graupel fall	$\mathrm{m\,s^{-1}}$, m

4.4 Execution of CReSS

4.4.1 Execution of *solver.exe*

Using an example experiment of the Kelvin-Helmholtz waves, the procedures to execute the solver program *solver* will be explained.

First, users uncompress and expand the compressed archive file of CReSS. (This example is CReSS ver.2.3 while the essence is the same in other version.)

```
% ls
cress2.3m.tar.Z
% uncompress -c cress2.3m.tar.Z | tar xvf -
  messages
     :
% ls
CReSS2.3m        cress2.3m.tar.Z
%
```

The directory structure will be as follows.

```
% cd CReSS2.3m
% ls
Doc             Src             compile.conf
Form            Tmp             compile.csh
%
```

Directory begin with a capital letter while files with a small letter. These are summarized in the following table.

`Doc`	Documents and `Readme` are included.
`Form`	Examples of configuration and the related sounding data are included.
`Src`	All source codes are archived in the directory.
`Tmp`	A temporary directory used in compilation of CReSS.
`compile.conf`	Configuration file of compilation.
`compile.csh`	Shell script to control the compilation.

Some configuration files and the related sounding files are archived in the directory of `Form`. Copy the configuration file and the sounding file of the Kelvin-Helmholtz waves to the directory of `CReSS2.3m` as the following names. We will use `test` as the name of experiment. All I/O data must begin with `test.`.

```
% cp Form/user.conf.cats.eye.form.2 user.conf
% cp Form/sounding.txt.cats.eye.form test.sounding.txt
% ls
Doc                Src                compile.conf       test.sounding.txt
```

```
Form                Tmp                 compile.csh         user.conf
%
```

Second, the configuration file of compilation `compile.conf` will be edited if necessary. If the compiler is different from `f90` or `mpif90` or special options of compilation are used, the file should be edited. Otherwise, the file is not changed.

Third, compile the solver as follows. The dimension is defined in the user configuration file `user.conf` and include files are produced automatically. It is, therefore, not necessary to change the source codes of the solver.

```
% compile.csh solver [compile.conf]
cd Src;  messages
         :
%
```

When the compilation is completed, the executable file `solver.exe` is generated. (Actually, `solver.exe` is a symbolic link to that in the directory of `Src`.)

```
% ls
Doc                 Tmp                 solver.exe
Form                compile.conf        test.sounding.txt
Src                 compile.csh         user.conf
%
```

Forth, execute the solver with reading the user configuration file `user.conf` from the standard input and outputting `test.out` to the standard output. In order to use main frame computer, job script and NQS could be necessary. In that case, ask the system manager how to use NQS.

```
% solver.exe < user.conf > test.log &
%
```

If the calculation is terminated normally, CReSS leaves the message "`This program stopped normally.`" at the end of the standard output `test.out` and the history dumped files (the names include `dmp`), the geographic information file (the names include `geography`) and restart files (the names include `res`) are produced. The standard output includes history of calculation and maximum and minimum of prognostic variables.

```
% ls
Doc                         test.dmp000120.pe0003.bin
Form                        test.dmp000160.pe0000.bin
Src                         test.dmp000160.pe0001.bin
Tmp                         test.dmp000160.pe0002.bin
compile.conf                test.dmp000160.pe0003.bin
compile.csh                 test.dmp000200.pe0000.bin
solver.exe                  test.dmp000200.pe0001.bin
```

```
test.dmp000000.pe0000.bin  test.dmp000200.pe0002.bin
test.dmp000000.pe0001.bin  test.dmp000200.pe0003.bin
test.dmp000000.pe0002.bin  test.dmp000240.pe0000.bin
test.dmp000000.pe0003.bin  test.dmp000240.pe0001.bin
test.dmp000040.pe0000.bin  test.dmp000240.pe0002.bin
test.dmp000040.pe0001.bin  test.dmp000240.pe0003.bin
test.dmp000040.pe0002.bin  test.geography.bin
test.dmp000040.pe0003.bin  test.out
test.dmp000080.pe0000.bin  test.res000240.check.txt
test.dmp000080.pe0001.bin  test.res000240.pe0000.bin
test.dmp000080.pe0002.bin  user.res000240.pe0001.bin
test.dmp000080.pe0003.bin  test.res000240.pe0002.bin
test.dmp000120.pe0000.bin  test.res000240.pe0003.bin
test.dmp000120.pe0001.bin  test.sounding.txt
test.dmp000120.pe0002.bin  user.conf
%
```

4.4.2 Execution of *unite.exe*

In parallel calculation, the history dumped files are outputted separately by each processing element. To combine the files, the post-processor *unite* is used.

The compilation of *unite* is as follows.

```
% compile.csh unite [compile.conf]
cd Src;  messages
          :
%
```

Then the post-processor `unite.exe` is generated. `unite.exe` is an interactive executable. The following example combines the history dumped file at 240 seconds. The name of experiment is `test`.

```
% unite.exe
Now the program, unite start.

    messages
        :
This program stopped normally.
%
```

The combined file is `test.dmp000240.united.bin`. The format of the file is described in Subsection 4.3.2. Types and order of output are also found in the standard output `test.log`.

The file is used in a graphic application such as Grads, then the result will be displayed as in

Fig.4.7 and Fig.4.8.

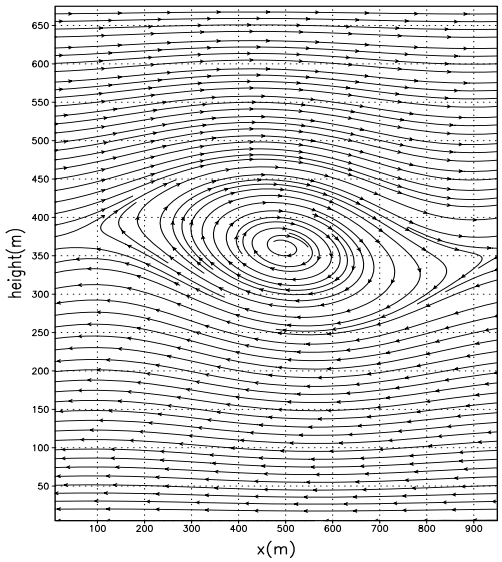

Figure 4.7. Stream function of $(u - w)$ at 240 seconds from the initial.

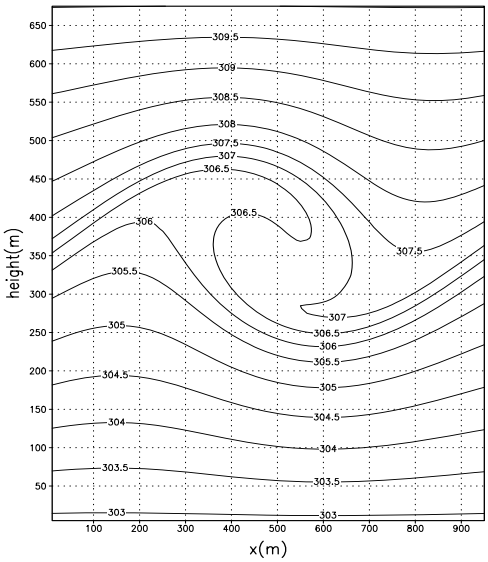

Figure 4.8. Potential temperature [K] at 240 seconds from the initial.

4.4.3 Execution of *terrain.exe*

The pre-processor *terrain* is a program to interpolate a terrain data into the model grids. This is necessary in a prediction experiment as well as a numerical experiment using a non-analytic topography.

While an external terrain data is necessary to produce the model terrain using *terrain*, the format of the external terrain data is not specified in CReSS. In order to read the data, it is necessary to modify the subroutine `rdtrn.f` which will be found in the directory `Src`. User will modify the parts of the `open` statement and `read` statement in `rdtrn.f` which is indicated by `#` to read the external terrain data.

```
* ##### You need to modify the following lines. #####

      siz=nid*njd*wlngth

      write(trnfl(1:16),'(a16)') 'data.terrain.bin'

      open(iotrn,iostat=stat,file=trnfl(1:16),status='old',
     .     access='direct',form='unformatted',recl=siz)

* ####################
```

```
* ##### You need to modify the following lines. #####

      read(iotrn,rec=1,err=100) ((htdat(id,jd),id=1,nid),jd=1,njd)

* ####################
```

The unit of the terrain data is meter. If the unit is not meter, it must be corrected here.

After the namelists of the user configuration file are specified (refer to Section 4.1), compile as follows.

```
% compile.csh terrain [compile.conf]
cd Src;  messages
          :
%
```

When the compilation is completed, the executable program `terrain.exe` is generated. Actually, `terrain.exe` is a symbolic link to that in the directory of `Src`.

The execution of `terrain.exe` is as follows. The input data file must be in the current directory.

```
% terrain.exe < user.conf > test.log.terrain &

    messages
       :
This program stopped normally.
%
```

If the execution succeeded, the terrain data file with the experimental name at the beginning (in the case, it is `test`) will be generated. (For example, `test.terrain.bin`). The program `terrain.exe` of the serial and parallel versions is performed in a single processing element. A single file will be produced by the serial version while multiple files by the parallel version with an extention of `.pe`*yyyy*. Where *yyyy* is the unit number of the processing element. This is necessary to execute the parallel version of `solver.exe`.

4.4.4 Execution of *gridata.exe*

The pre-processor *gridata* produces 3-dimensional grid point data files and the lateral boundary data files.

The format of the external GPV data such as an objective analysis data or a coarse model output is not specified in CReSS. In order to read the data, it is necessary to modify the subroutine `rdgpv.f` which will be found in `Src`. User will change the `open` statement and `read` statement of the subroutine `rdgpv.f`, which are indicated by `#`. The file names of the input data must include date and time of the Gregorian calendar *yyyymmddhhmm* (UTC).

```
* ##### You need to modify the following lines. #####

      write(gpvfl(1:24),'(a8,a12,a4)') 'data.gpv',cdate(1:12),'.bin'

      open(igpv,iostat=stat,file=gpvfl(1:24),status='old',
     .     access='sequential',form='unformatted')

* ####################

* ##### You need to modify the following lines. #####

      read(igpv,end=100,err=100) zdat

      read(igpv,end=100,err=100) udat
      read(igpv,end=100,err=100) vdat

      read(igpv,end=100,err=100) pdat
                  :

* ####################
```

The necessary variables are summarized in the following table.

zdat	height of the each grid point	m
udat	x-component of horizontal velocity	m s^{-1}
vdat	y-component of horizontal velocity	m s^{-1}
pdat	pressure	Pa
ptdat	potential temperature or temperature	K

The following variables are used optionally.

wdat	vertical velocity	m s^{-1}
qvdat	water vapor mixing ratio or relative humidity	kg kg^{-1} or %
qcdat	mixing ratio of cloud water	kg kg^{-1}
qrdat	mixing ratio of rain water	kg kg^{-1}
qidat	mixing ratio of cloud ice	kg kg^{-1}
qsdat	mixing ratio of snow	kg kg^{-1}
qgdat	mixing ratio of graupel	kg kg^{-1}

If the units of the variable are different from the above, correction is necessary in the subroutine.

If data at the surface is present (`refsfc = 1`), the data is allocated at the first level (`kd = 1`). If the altitude of the surface is allocated at the first level of `zdat` and the corresponded data is allocated at the first level of each data, the level which is lower than the altitude will be ignored in the vertical interpolation when `refsfc = 1`. If `refsfc = 0`, the interpolation or extrapolation will be performed in the model grid even if the data is below the ground.

Compilation and execution of `gridata.exe` are essentially the same as `terrain.exe`. After the namelists are configured (refer to Section 4.1), compile the program as follows.

```
% compile.csh gridata [compile.conf]
cd Src;  messages
       :
This program stopped normally.
%
%
```

If the compilation is succeeded, the executable file `gridata.exe` is generated. Actually `gridata.exe` is also a symbolic link to that in the directory `Src`.

The execution of `terrain.exe` is as follows. The input data file must be in the current directory.

```
% gridata.exe < user.conf > test.log.gridata &
cd Src;  messages
         :
%
```

If the execution succeeded, 3-dimensional grid data files (in this case `test.grd`*xxxxxx*`.bin`). If the GPV data include abnormal data, *gridata.exe* will be stopped with messages.

The program `gridata.exe` of the serial and parallel versions is performed in a single processing element. A single file will be produced by the serial version while multiple files by the parallel version with an extension of `.pe`*yyyy*. Where *yyyy* is the unit number of the processing element. This is necessary to execute the parallel version of `solver.exe`.

Chapter 5

Program Lists

In this chapter, structures of solver and other processors of CReSS version 2.3m are summarized as a reference. The tree diagrams of the programs will be presented. Most descriptions are about the parallel version of CReSS. Since the serial version of CReSS is essentially the same as the parallel version, the readers will have no trouble with the descriptions even if they use the serial version.

Each source code of CReSS includes a single subroutine and each file has the same name as subroutine. This is helpful to read the source code as well as the tree diagram.

5.1 Program list

In this section, the programs of CReSS version 2.3, the composing subroutines, include files and the library subroutine of MPI will be described.

5.1.1 Main programs of CReSS

programs		description
solver	`solver.exe`	The solver program is the main program to perform the time integration from an initial value with boundary conditions.
check	`check.exe`	A pre-processor to check the consistency of parameters in the configulation file "user.conf".
gridata	`gridata.exe`	A pre-processor to produce an initial and boundary values for the solver by interpolating grid point values to the grid points of the solver.
terrain	`terrain.exe`	A pre-processor to produce model terrain data by interpolating of topography data.
surface	`surface.exe`	A pre-processor to produce model surface data by interpolating of surface data.
radata	`radata.exe`	A pre-processor to produce radar data on the model grid by interpolating of a topography data.
unite	`unite.exe`	A post-processor to combine multiple-output history files and/or the multiple terrain file in to a single file after parallel calculation.
rstruct	`rstruct.exe`	A post-processor to rearrange the re-start files for different number of processor.

5.1.2 List of subroutines

CReSS is composed of many subroutines. All subroutines are listed below and their functions are described. Each file includes its corresponding subroutine. The names in the list will also refer to the file names.

solver	this is the main program for the solver processer solver.
check	this is the main program for the pre processor ckeck.
terrain	this is the main program for the pre processor terrain.

surface	this is the main program for the pre processor surface.
gridata	this is the main program for the pre processor gridata.
radata	this is the main program for the pre processor radata.
unite	this is the main program for the post processor unite.
rstruct	this is the main program for the post processor rstruct.
abortslv	check the file to abort the solver.
addqt	add the source intensity to the current tracer.
adjstbw	adjust the mean water mass to be between their boundaries.
adjstnc	adjust the concentrations of the cloud ice, snow and graupel for the given mixing ratio.
adjstni	adjust the concentrations of the cloud ice.
adjstnsg	adjust the concentrations of the snow and graupel for the given mixing ratio.
adjstq	force the hydrometeor mixing ratio more than user specified value.
adjstuv	adjust the x and y components of velocity.
advbspe	calculate the base state pressure advection for the horizontally explicit and vertically explicit method.
advbspi	calculate the base state pressure advection for the horizontally explicit and vertically implicit method.
advbspt	calculate the base state potential temperature advection.
advp	calculate the pressure advection.
advs	calculate optional scalar advection.
advuvw	calculate the velocity advection.
aggregat	calculate the aggregation rate for the cloud ice and snow.
allocbuf	allocate the communication buffer array.
allocgrd	allocate the array for gridata.
allocgrp	allocate the grouped domain arrangement array.
allocmph	allocate the array for cloud physics.
allocnio	allocate the unit numbers table.
allocrad	allocate the array for radata.
allocrst	allocate the array for rstruct.
allocsfc	allocate the array for surface.
allocslv	allocate the array for solver.
alloctrn	allocate the array for terrain.
allocuni	allocate the array for unite.
baserho	the base state density is multiplyed by the Jacobian.
bbcs	set the vertical boundary condition for optional scalar variable.
bbcw	set the bottom boundary conditions for the z components of velocity.
bc2d	set the boundary conditions for two dimensional variables.
bc4news	set the boundary conditions at the four corners.
bc8u	set the boundary conditions for optional variable at the u points.
bc8v	set the boundary conditions for optional variable at the v points.
bc8w	set the boundary conditions for optional variable at the w points.

bcbase	set the bottom and the top boundary conditions for the base state variables.
bcgrd	set the boundary conditions for GPV data at current forecast time.
bcgsms	set the boundary conditions for optional scalar variable in GPV data smoothing.
bcgsmu	set the boundary conditions for the x components of velocity in GPV data smoothing.
bcgsmv	set the boundary conditions for the y components of velocity in GPV data smoothing.
bcgsmw	set the boundary conditions for the z components of velocity in GPV data smoothing.
bcini	set the boundary conditions for GPV data at forecast start time.
bcphi	set the boundary conditions for the parabolic pertial differential equation.
bcrad	set the boundary conditions for radar data.
bcradqp	set the boundary conditions for radar data.
bcten	set the boundary conditions for optional tensor.
bcycle	set the periodic boundary conditions.
bcyclex	set the periodic boundary conditions in x direction.
bcycley	set the periodic boundary conditions in y direction.
bruntv	calculate the Brunt-Vaisala frequency squared.
bulksfc	calculate the bulk coefficients of surface flux.
buoytke	calculate the buoyancy production in the T.K.E. equation.
buoywb	calculate the buoyancy in the large time steps integration.
buoywse	calculate the buoyancy in the small time steps integration for the horizontally explicit and vertically explicit method.
buoywsi	calculate the buoyancy in the small time steps integration for the horizontally explicit and vertically implicit method.
castgrp.multi	broadcast the grouped domain arrangemnet array from the root processor element to the others.
castgrp.single	do nothing because this subroutine is dummy.
castname.multi	broadcast the namelist array from the root processor element to the others.
castname.single	do nothing because this subroutine is dummy.
castvar	broadcast optional array from the root processor element to the others.
cg2mkbw	change measurement from [c] [g] to [m] [k] for warm bin cloud physics.
charging	perfom useless calculations, because this routine is imitation.
chkerr.multi	check the error descriptor for calling subroutine, destroy.
chkerr.single	check the error descriptor for calling subroutine, destroy.
chkfall	check the time interval for upwind differential of the sedimentation to hold the common values.
chkfile	check the namelist variables which are read out from the input files.
chkgpv	check the GPV data variables.
chkitr	check the convergence of the iteration.
chkkind	check the kind of the Fortran veriable.
chkmoist	check the air moisture.

chkmxn	read in the maximum and minimum data variable to the standard i/o and check the maximum and minimum data variable.
chkname	read out and set the namelist variables.
chknlgrd	check the namelist variables for gridata.
chknlrad	check the namelist variables for radata.
chknlrst	check the namelist variables for rstruct.
chknlsfc	check the namelist variables for surface.
chknlslv	check the namelist variables for solver.
chknltrn	check the namelist variables for terrain.
chknluni	check the namelist variables for unite.
chkopen.multi	check the common files open between all processor elements.
chkopen.single	check the common files open between all processor elements.
chkrad	check the radar data variables.
chkrain	check the precipitation on the surface.
chkstd.multi	check the control flag of message type for standard i/o.
chkstd.single	do nothing because this subroutine is dummy.
closedmp	close the opening history file.
cloudamt	calculate the cloud amount.
cloudphy	control the inferior subroutines for the cloud micro physics.
cnt2phy	calculate the z components of velocity.
coldblk	calculate the source amounts of the bulk cold cloud micro physics.
collbw	perform the collection processes between water bins.
collc2r	calculate the collection rate between the cloud water and the rain water.
collect	calculate the collection rate.
combin	declare the array for warm bin cloud physics.
comblk	declare the array for bulk cold rain model.
combuf	declare the common used buffer array.
comcap	declare the caption for dumped variables.
comday	declare and define common used and referenced calendar.
comdmp	declare the control variables of history file.
comerr	declare and define common used error list array.
comgrd	declare the array for gridata.
comgrp	declare the grouped domain arrangement array in entire domain.
comindx	declare the unique index for each namelist variable in the namelist array.
comkind	declare and define the kind of veriable for the long integer.
commath	declare and define the mathmatical constants.
commpi	declare the parameters of parallelizing.
comname	declare and define common used namelist array.
comnio	declare the unit numbers table.
comphy	declare and define the physical constants.
comrad	declare the array for radata.

comrst	declare the array for rstruct.
comsave	declare the external referenced variables to avoid save attribute statement.
comsfc	declare the array for surface.
comslv	declare the array and variable for solver.
comstd	declare the control flag of standard i/o.
comtable	declare and define common used and referenced table array.
comtrn	declare the array for terrain.
comuni	declare the array for unite.
convc2r	calculate the auto conversion rate from the cloud water to the rain water.
convers	calculate the conversion rate from the cloud water to the rain water, from the cloud ice to the snow and from the snow to the graupel.
convland	convert veriable type of the land use categories.
copy1d	copy 1-dimensional invar to outvar.
copy2d	copy 2-dimensional invar to outvar.
copy3d	copy 3-dimensional invar to outvar.
copy4d	copy invar to outvar.
coriuv	calculate the Coriolis force in the x and the y components of velocity equation.
coriuvw	calculate the Coriolis force in the x, the y and the z components of velocity equation.
cpondht	correspond and damp the model height to the external data height.
cpondpe.multi	wait for the other processor elements' signals.
cpondpe.single	do nothing because this subroutine is dummy.
cpondsfc	correspond the surface data to the land use data.
culintg	control the inferior subroutines for the time steps integration of the Cubic Lagrange advection.
currpe	set the parameters of parallelizing from user arrangement.
curveuvw	calculate the curvature of earth.
dbz2kg	convert the hydrometeor mesuremement from [dBZe] to [kg m^{-3}]
defdim	declare the dimensions of all array.
defname	declare the namelist variables and blocks.
depsit	calculate the evaporation rate from the rain water to the water vapor and the deposition rate from the water vapor to the ice hydrometeor.
depsitbw	perform deposition for water bin.
destroy	read in the error messages to the standard i/o and stop the program.
dfomssq	calculate the deformation squared.
dfomten	calculate the deformation tensor.
diabat	calculate the diabatic in the pressure equation.
diagnc	get the diagnostic concentrations of the cloud ice, snow and graupel.
diagni	get the diagnostic concentrations of the cloud ice.
diagnsg	get the diagnostic concentrations of the snow and graupel.
diffequa	solve the parabolic pertial differential equation.

disptke	calculate the dissipation in the T.K.E. equation.
distrpg	calculate the distribution ratio at which the collisions between rain water and snow and reset the collection rate.
distrqp	distribute the observed precipitation to the rain, snow and graupel mixing ratio.
diver2d	calculate the negative divergence horizontally.
diver3d	calculate the three dimensional negative divergence.
diverpe	calculate the divergence in the pressure equation with the horizontally explicit and vertically explicit method.
diverpih	calculate the divergence horizontally in the pressure equation with the horizontally explicit and vertically implicit method.
diverpiv	calculate the divergence vertically in the pressure equation with the horizontally explicit and vertically implicit method.
eddydif	calculate the eddy diffusivity.
eddypbl	calculate the eddy viscosity and diffusivity in planetaty boundary layer.
eddyvis	calculate the eddy viscosity and the turbulent length scale.
eddyvisj	the eddy viscosity is devided by Jacobian.
endmpi.multi	read in the final message to the standard i/o and call the function to finalize the MPI processes.
endmpi.single	read in the final message to the standard i/o.
estimsfc	re-estimete the surface value on the water, ice and snow surface.
evapr2v	calculate the evaporation rate from the rain water to the water vapor.
exbcpt	force the lateral boundary values to the external boundary values for potential temperature.
exbcq	force the lateral boundary values to the external boundary values for optional mixing ratio.
exbcss	force the lateral boundary values to the external boundary values for scalar variables.
exbcu	force the lateral boundary values to the external boundary values for the x components of velocity.
exbcv	force the lateral boundary values to the external boundary values for the y components of velocity.
exbcw	force the lateral boundary values to the external boundary values for the z components of velocity.
fallblk	perform the fall out.
fallbw	perform the fall out.
fallqr	perform the fall out of the rain water.
fdmdrv	control the inferior subroutines for the finite difference time steps integration.
fituvwc	fit the x and the y components of velocity and the zeta components of contravariant velocity to the mass consistent equation by the lamb.
forcedrv	control the inferior subroutines for the forcing terms calculation.
forcep	calculate the forcing term in the pressure equation.
forcept	calculate the forcing term in the potential temperature equation.

forceq	calculate the forcing terms in the water hydrometeor and the ice hydrometeor and concentrations equations.
forceqcg	calculate the forcing terms in the charging distribution equation.
forceqt	calculate the forcing term in the tracer equation.
forceqv	calculate the forcing term in the water vapor mixing ratio equation.
forces	control the inferior subroutines for the scalar forcing terms calculation and brige the complicated array, rkh, rkh8u and rkh8v.
forcesfc	get the surface flux to bottom boundary.
forcetke	calculate the forcing terms in the T.K.E. equations.
forceuvw	calculate the forcing terms in the velocity equations.
freezing	calculate the freezing rate from the rain water to the graupel.
gaussel	solve the tridiagonal equation with the Gauss elimination.
get1d	get the horizontally averaged variables.
getarea	get the area of each boundary plane.
getbase	extract the base state variables for the model grid from the three dimensional data variables.
getbufgx	get the receiving buffer in x direction for grouped domain.
getbufgy	get the receiving buffer in y direction for grouped domain.
getbufsx	get the receiving buffer in x direction for sub domain.
getbufsy	get the receiving buffer in y direction for sub domain.
getcname	get the character namelist variable.
getdate	calculate the current date from the time with ssssss format.
getexner	calculate the Exner function.
getiname	get the integer namelist variable.
getindx	get the maximum and minimum indexes of do loops for optional variable or data grid.
getkref	get the index of the base state reference pressure.
getllsw	calculate the latitude and the longitude at south-west corner.
getmxn	get the maximum and minimum prognostic variables in the grouped grid points.
getnews	get the mininum and maximum data index to create the base state variables.
getpt0	set the initial potential temperature perturbation.
getqt0	set the initial tracer.
getrich	calculate the bulk Richardson number on the surface.
getrij	calculate the real indexes at the model grid points in the data region.
getrname	get the real namelist variable.
getta2d	calculate the air temperature.
getta3d	calculate the air temperature.
gettime	calculate the model time in the large time steps integration.
gettrn	get the terrain height.
getunit	get the unit number to open the file.
getvdens	calculate the inverse of base state density.

getvpres	calculate the inverse of the pressure.
getxy	calculate the x and the y coordinates.
getz	calculate the zeta coordinates.
getz11	calculate the zeta coordinates.
getzlow	calculate the z physical coordinates at lowest plane.
getzph	calculate the z physical coordinates.
grddrv	control the inferior subroutines to interpolate the grid data.
grdstep	calculate the number of steps of the main do loop in grddrv.
gseidel	solve the tridiagonal equation with the Gauss-Seidel method.
gsmoos	smooth the interpolated scalar GPV data.
gsmoou	smooth the x components of velocity of interpolated GPV data.
gsmoov	smooth the y components of velocity of interpolated GPV data.
gsmoow	smooth the z components of velocity of interpolated GPV data.
hculs	calculate the horizontal scalar advection by the Cubic Lagrange scheme.
hculuvw	calculate the horizontal velocity advection by the Cubic Lagrange scheme.
heatsfc	calculate the sensible and latent heat on the surface.
heve	control the inferior subroutines for the horizontally explicit and vertically explicit method.
hevi	control the inferior subroutines for the horizontally explicit and vertically implicit method.
hint2d	interpolate the data variables to the model grid horizontally.
hint3d	interpolate the data variables to the model grid horizontally.
hintlnd	interpolate the land use data to the model grid.
hintrad	interpolate the data variables to the model grid horizontally.
ini0name	initialize the namelist variables.
inibinbw	control the inferior subroutines for the initial bin distribution of total water.
inichar	initialize the character variable with space.
inicom	initialize the common used module variables.
inidef	initialize the dimension and namelist variables.
inidisbw	set the initial bin distribution of total water.
inierr	initialize the common used error list array.
inimpi.multi	call the function to initialize the MPI processes and set the parameters of parallelizing.
inimpi.single	set the parameters of parallelizing in the case the program run one after another.
ininame	initialize the common used namelist array.
inisfc	initialize the surface physical parameters.
initke	initialize the T.K.E.
initund	initialize the temperature under the ground and sea.
inivar	set the initial conditions.
instvel	calculate the maximum instantaneous wind velocity.
intgdrv	control the inferior subroutines for the time steps integration.

intrpgpv	control the interior subroutines for interpolating the GPV data to the model grid points.
intrprad	control the interior subroutines for interpolating the radar data to the model grid points.
intrpsnd	extract the base state variables from interpolated sounding data.
jacobian	calculate the transformation Jacobian.
kh8uv	set the horizontal eddy diffusivity at the u and v points.
lbcs	set the lateral boundary conditions for optional scalar variable.
lbcu	set the lateral boundary conditions for the x components of velocity.
lbculs	set the lateral boundary conditions for optional scalar variable.
lbculu	set the lateral boundary conditions for x components of velocity.
lbculv	set the lateral boundary conditions for y components of velocity.
lbculw	set the lateral boundary conditions for w components of velocity.
lbcv	set the lateral boundary conditions for the y components of velocity.
lbcw	set the lateral boundary conditions for the z components of velocity.
lbcwc	set the lateral boundary conditions for the zeta components of contravariant velocity.
ll2xy	calculate the x and the y coordinates from the latitude and the longitude.
llnews	calculate the latitude and the longitude at model corner.
lspdmp	calculate the relaxed lateral sponge damping coefficients.
lspp	calculate the lateral sponge damping for pressure.
lspqv	calculate the lateral sponge damping for the water vapor mixing ratio.
lsps	calculate the lateral sponge damping for optional scalar variable.
lsps0	calculate the lateral sponge damping for optional scalar variable to initial.
lspuvw	calculate the lateral sponge damping for the velocity.
mapfct	calculate the map scale factors.
masscon	fit the x, y and z components of velocity to the mass consistent equation.
melting	calculate the melting rate from the cloud ice to the cloud water and from the snow and graupel to the rain water.
mk2cgbw	change measurement from [m] [k] to [c] [g] for warm bin cloud physics.
more0q	force the mixing ratio to not being less than zero.
move2d	set the grid moving velocity.
mvrst	rename restructed restart files.
ndgstep	get the control flag of analysis nudging.
newblk	solve the new potential temperature perturbation, the mixing ratio and concentrations.
newsindx	get the mininum and maximum data index to create the base state variables.
nlsmqv	perform the non linear smoothing for the water vapor mixing ratio.
nlsms	perform the non linear smoothing for optional scalar variable.
nlsmuvw	perform the non linear smoothing for the velocity.
nuc1stc	calculate the nucleation rate of the condensation, contact and homogeneous.
nuc1stv	calculate the nucleation rate of the deposition or sorption.

nuc2nd	calculate the secondary nucleation rate of the ice crystals.
numchar	count the number of the character variable.
opendmp	open the history file when the current time reaches marked time.
outcheck	open and read in the data to the history data checking file.
outctl	create the GrADS control file.
outdmp	control the inferior subroutines to read in the variables to the history file.
outdmp2d	read in the two dimensional variables to the history file.
outdmp3d	read in the three dimensional variables to the history file.
outgeo	open and read in data to the geographical file.
outgrd	open and read in the data to the three dimensional grided data file.
outllsw	read in the latitude and the longitude at south-west corner to standard i/o.
outmxn	read in the maximum and minimum prognostic variables to the standard i/o when the current time reaches marked time.
outpbl	read in the surface variables to the history file.
outrad	open and read in the data to the radar data file.
outres	open and read in the data to the restart file when the current time reaches marked time.
outrst	open and read in the data to the restructed restart file.
outsfc	open and read in the data to the two dimensional grided surface file.
outstd1	read in the first message to standard i/o.
outstd10	read in the messages to standard i/o.
outstd11	read in the message to standard i/o.
outstd12	read in the messages to standard i/o.
outstd13	read in the messages to standard i/o.
outstd14	read in message to standard i/o.
outstd15	read in message to standard i/o.
outstd16	read in message to standard i/o.
outstd2	read in the final message to standard i/o.
outstd3	read in the message to standard i/o.
outstd4	read in message to standard i/o.
outstd5	read in message to standard i/o.
outstd6	read in the message to standard i/o.
outstd7	read in the messages to standard i/o.
outstd8	read in the messages to standard i/o.
outstd9	read in the messages to standard i/o.
outtrn	open and read in the data to the two dimensional grided terrain file.
p2gpv	perform the analysis nudging to GPV data of pressure.
paractl	set the parameters of GrADS control file.
pbldrv	control the inferior subroutines for the virtical diffusion in the planetary boundary layer.
pblptv	calculate the virtical diffusion for the virtual potential temperature.
pblqv	calculate the virtical diffusion for the water vapor mixing ratio.

pblu	calculate the virtical diffusion for the x components of velocity.
pblv	calculate the virtical diffusion for the y components of velocity.
pc2kg	convert the relative humidity to the water vapor mixing ratio.
pgrad	calculate the pressure gradient force in the velocity equations.
pgradiv	calculate the pressure gradient force vertically with the horizontally explicit and vertically implicit method.
phasev	calculate the phase speed for the open boundary conditions.
phasevbc	calculate the diffrential phase speed term between the external boundary and model grid variables.
phvbcs	calculate the diffrential phase speed term between the external boundary and model grid for scalar variables.
phvbcuvw	calculate the diffrential phase speed term between the external boundary and model grid for velocity variables.
phvs	calculate the scalar phase speed for the open boundary conditions.
phvuvw	calculate the velocity phase speed for the open boundary conditions.
phy2cnt	calculate the zeta components of contravariant velocity.
phycood	calculate the three dimensional z physical coordinates.
prodctwg	calculate the graupel production rate.
putbufgx	put the sending buffer in x direction for grouped domain.
putbufgy	put the sending buffer in y direction for grouped domain.
putbufsx	put the sending buffer in x direction for sub domain.
putbufsy	put the sending buffer in y direction for sub domain.
putunit	return the unit number to close the file.
qp2rad	perform the analysis nudging to radar data of optional precipitation mixing ratio.
raddrv	control the inferior subroutines to interpolate the radar data.
radiat	calculate the short and long wave radiation.
radstep	calculate the number of steps of the main do loop in raddrv.
rbcpt	set the radiative lateral boundary conditions for potential temperature.
rbcq	set the radiative lateral boundary conditions for optional mixing ratio.
rbcqv	set the radiative lateral boundary conditions for the water vapor mixing ratio.
rbcs	set the radiative lateral boundary conditions for optional scalar variable.
rbcs0	set the radiative lateral boundary conditions for optional scalar variable.
rbcss	set the radiative lateral boundary conditions for optional scalar variable.
rbcu	set the radiative lateral boundary conditions for the x components of velocity.
rbcv	set the radiative lateral boundary conditions for the y components of velocity.
rbcw	set the radiative lateral boundary conditions for the z components of velocity.
rdcheck	open and read out the data from the history data checking file.
rdconf	read out the namelist variables.
rdgrd	read out the data from the three dimensional grided data file.

rdgrp	open and read out the data from the grouped domain arrangement file and check them.
rdini	open and read out the data from the three dimensional grided data file at forcast start time.
rdname	read out and set the namelist variables.
rdradmrk	read out the data from the radar data file.
rdradnxt	read out the data from the radar data file.
rdres	open and read out the data from the restart file.
rdsfc	open and read out the data from the two dimensional grided surface file.
rdsnd	open and read out the data from the sounding data file.
rdtrn	open and read out the data from the two dimensional grided terrain file.
rdtund	open and read out the temperature under the ground and sea data from the restart file.
reducelb	reduce the integrated value on lateral boundary.
reducevb	reduce the integrated value on bottom and top boundary.
remapbw	remap the shifted water mass and concentrations for origianl distributed water bins.
repdrv	control the inferior subroutines for the repositon.
repsit2d	reposition the restructed boundary array form original restart boundary array.
repsit3d	reposition the restructed array form original restart array.
resetag	reset the message tag.
rmres	remove the original restart files.
rotuvm2s	rotate the x and the y components of velocity from the projected grid to the latitude and longitude grid.
rotuvs2m	rotate the x and the y components of velocity from the latitude and longitude grid to the projected grid.
roughitr	calculate the roughness length on the sea surface by iteration at forcast start time.
roughnxt	reset the roughness parameter on the sea surface to the next time step.
rstdrv	restruct the restart files.
rstindx	calculate the minimum and maximum and differential do loops indexes to reposition.
rststep	calculate the number of steps of the main do loop in rstdrv.
rstuvwc	the base state density x the Jacobian is multiplyed by u, v and w.
s2gpv	perform the analysis nudging to GPV data of optional scalar variable.
sadjstbw	perform the saturation adjustment for water in the case the generated total water is more than critical value.
satadjst	control the inferior subroutines for the saturation adjustment.
set1d	set the horizontally averaged values from the read out sounding data.
setbase	set the base state variables.
setbin	set the bin parameters.

setblk	calculate the air temperature, saturation mixing ratio, latent heat, viscosity of air, molecular diffusivity of water vapor, mean mass of cloud ice, mean diameters and ventilation factors.
setcap	set the common used and referenced caption for dumped variables.
setcom	set the common used and referenced array.
setcst1d	fill in the 1-dimensional array with the constant value.
setcst2d	fill in the 2-dimensional array with the constant value.
setcst3d	fill in the 3-dimensional array with the constant value.
setcst4d	fill in the 4-dimensional array with the constant value.
setday	set the common used and referenced calendar.
setdim	set the array dimension.
setgpv	set the three dimensional grided point variables.
setgrid	set the model grid.
sethalo	set the parameters of sendding and receiving.
setname	set the namelist variables.
setproj	calculate the map projection parameters.
setrad	control the inferior subroutines to interpolate the radar data.
setradqp	set the three dimensional grided point radar variables.
setsfc	calculate the magnitude of velocity, virtual potential temperature and water vapor mixing ratio on the surface.
setvar1d	set the initial velocity and the scalar perturbations.
setvar3d	set the initial and the boundary conditions.
sfcdrv	control the inferior subroutines to interpolate the surface data.
sfcflx	calculate the bulk and the exchange coefficients of surface flux.
sfcphy	call the surface process driver routine to brige the complicated array.
sheartke	calculate the shear production in the T.K.E. equation.
shedding	calculate the shedding rate from the snow and graupel to the rain water.
shift2nd	exchange the values horizontally in the case the fourth order calculation is performed.
shiftgx	exchange the values in x direction between grouped domain.
shiftgy	exchange the values in y direction between grouped domain.
shiftsx	exchange the values in x direction between sub domain.
shiftsy	exchange the values in y direction between sub domain.
siadjst	perform the saturation adjustment for ice.
slvdrv	control the inferior subroutines for the time steps integration.
slvstep	calculate the number of steps in slvdrv.
smalldrv	control the inferior subroutines for the small time steps integration.
smoo2p	perform the second order smoothing for the pressure.
smoo2qv	perform the second order smoothing for the water vapor mixing ratio.
smoo2s	perform the second order smoothing for optional scalar variable.
smoo2uvw	perform the second order smoothing for the velocity.
smoo4p	perform the fourth order smoothing for the pressure.

smoo4qv	perform the fourth order smoothing for the water vapor mixing ratio.
smoo4s	perform the fourth order smoothing for optional scalar variable.
smoo4uvw	perform the fourth order smoothing for the velocity.
smoogpv	smooth the interpolated GPV data.
sndwave	calculate the base state density x sound wave speed squared.
soildrv	control the inferior subroutines for the surface physics.
sparprt	separate the pressure and the potential temperature to the base state and the perturbation values.
srcblk	control the inferior subroutines for source term of the bulk cold rain cloud phycics.
stepcul	solve the variables to the next time step by the Cubic Lagrange scheme.
steppe	solve the pressure to the next time step with the horizontally explicit and vertically explicit method.
steppi	solve the pressure to the next time step with the horizontally explicit and vertically implicit method.
steppts	solve the potential temperature perturbation to the next time step.
steps	solve the scalar variables to the next time step.
steptund	solve the temperature under the ground and sea to the next time step.
stepuv	solve the x components of velocity to the next time step.
stepwe	solve the z components of velocity to the next time step with the horizontally explicit and vertically explicit method.
stepwi	solve the z components of velocity to the next time step with the horizontally explicit and vertically implicit method.
stretch	calculate the one dimensional physical coordinates.
strsten	calculate the stress tensor.
sumbin	get the total mixing ratio.
swadjst	perform the saturation adjustment for water.
swp2nxt	swap the prognostic variables to the next time step.
t2pt	convert the temperature to the potential temperature.
termblk	calculate the terminal velocity of the rain water, the snow and the graupel.
termbw2d	calculate the terminal velocity for water bin.
termbw3d	calculate the terminal velocity for water bin.
termqr	calculate the terminal velocity of the rain water.
timeflt	perform the Asselin time filter for the prognostic variables.
tkeflx	calculate the turbulent fluxes for the T.K.E..
totalbin	get the total water and ice mixing ratio and concentrations for bin categories.
totalq	get the total water and ice mixing ratio.
trilat	calculate the Coriolis parameters x 0.25.
trndamp	calculate the damped terrain height.
trndrv	control the inferior subroutines to interpolate the terrain data.
turbdrv	control the interior subroutines to solve the turbulent mixing.
turbflx	calculate the turbulent fluxes for optional scalar variable.

turbs	calculate optional scalar turbulent mixing.
turbtke	calculate the T.K.E. turbulent mixing.
turbuvw	calculate the velocity turbulent mixing.
undefice	convert the undefined values to the interpolated values in the sea ice distribution data.
undefsst	convert the undefined values to the interpolated values in the sea surface temperature data.
unidmpen	generate the united history file from the dumped files in entire domain.
unidmpgr	generate the united history file from the dumped files in grouped domain.
unidmprd	generate the united history file from the dumped files in reductional entire domain.
unidrv	control the inferior subroutines for uniting.
unigeoen	generate the united geography file from the dumped files in entire domain.
unigeogr	generate the united geography file from the dumped files in grouped domain.
unigeord	generate the united geography file from the dumped files in reductional entire domain.
unistep	calculate the number of steps of the main do loop in unidrv.
upwmbin	calculate the sedimentation and precipitation for optional bin mass.
upwnbin	calculate the sedimentation for optional bin concentrations.
upwnsg	calculate the sedimentation for optional precipitation concentrations.
upwqcg	calculate the sedimentation for optional charging distribution.
upwqp	calculate the sedimentation and precipitation for optional precipitation mixing ratio.
uvw2gpv	perform the analysis nudging to GPV data of the velocity.
uvw2rad	perform the analysis nudging to radar data of the velocity.
var8uvw	optional variable be averaged to u, v and w points.
var8w8s	optional variable at w points be averaged to scalar points.
var8w8u	optional variable at w points be averaged to u points.
var8w8v	optional variable at w points be averaged to v points.
vbcp	set the vertical boundary conditions for the pressure.
vbcqcg	set the vertical boundary condition for the charging distribution.
vbcs	set the vertical boundary condition for optional scalar variable.
vbcu	set the vertical boundary conditions for the x components of velocity.
vbcv	set the vertical boundary conditions for the y components of velocity.
vbcw	set the vertical boundary conditions for the z components of velocity.
vbcwc	set the vertical boundary conditions for the zeta components of contravariant velocity.
vculs	calculate the vertical scalar advection by the Cubic Lagrange scheme.
vculs0	calculate the vertical scalar advection by the Cubic Lagrange scheme.
vculuvw	calculate the horizontal velocity advection by the Cubic Lagrange scheme.
vextlow	extrapolate the base state variables in lowest layer.
vextund	extrapolate the base state variables in undefined layer.

vint11	interpolate the one dimensional data to the fine horizontally averaged levels vertically.
vint113	interpolate the variable to the model or data grid vertically.
vint133g	interpolate the variable to the model grid vertically.
vint133r	interpolate the variable to the model grid vertically.
vint31g	interpolate the three dimensional input variable to the one dimensional flat plane.
vint31r	interpolate the three dimensional input variable to the one dimensional flat plane.
vint31s	interpolate the three dimensional input variable to the one dimensional flat plane.
vintbase	control the interior subroutines for vertical interpolating the GPV data to the flat plane to create the base state variables.
vintgpv	control the interior subroutines for vertical interpolating the GPV data to the flat plane.
vintrad	control the interior subroutines for vertical interpolating the radar data to the flat plane.
vintsnd	interpolate the sounding data to fine interval levels vartically.
vspdmp	calculate the relaxed vertical sponge damping coefficients.
vspp	calculate the vertical sponge damping for pressure.
vspqv	calculate the vertical sponge damping for the water vapor mixing ratio.
vsps	calculate the vertical sponge damping for optional scalar veriables.
vsps0	calculate the vertical sponge damping for optional scalar variable to initial.
vspuvw	calculate the vertical sponge damping for the velocity.
warmbin	control the inferior subroutines for the warm bin cloud physics.
warmblk	calculate the source amounts of the bulk warm cloud micro physics.
xy2ij	calculate the real indexes from the two dimensional x and the y coordinates at the model grid points.
xy2ll	calculate the latitude and the longitude from the x and the y coordinates.

5.1.3 MPI subroutines list

The program codes of CReSS have MPI subroutines as well as the above subroutines. MPI subroutines used in CReSS are listed in this subsection. The detailed description of MPI routine is found in the manuals of MPI.

subroutine	**description**
`mpi_abort`	Halt MPI processes.
`mpi_allreduce`	Perform the same process in all nodes.
`mpi_barrier`	Adjust timing between nodes.
`mpi_bcast`	Broadcast from a node to all nodes.

continued to the next page

continued

subroutine	description
`mpi_comm_rank`	Get the number of node.
`mpi_comm_size`	Get a number of all nodes.
`mpi_comm_split`	Split communicator.
`mpi_finalize`	Terminate MPI.
`mpi_gather`	Gather data to a node from other nodes.
`mpi_init`	Initialize MPI.
`mpi_sendrecv`	Send and receive data simultaneously.

Part III

Fundamental Theory and Formulations

Chapter 6

The Basic Equations of CReSS

The basic equations of **CReSS** are composed of the Navier-Stokes equation including the effect of the earth rotation, the equation of thermodynamics, the continuity equation, the conservation equations of water vapor mixing ratio, equations of mixing ratios of hydrometeors, and equations of number density of solid hydrometeors. The CReSS model includes the formulation of various physical processes and the formulation of boundary value. In this chapter, we describe the formulation of the basic equations in the constitution.

CReSS can consider the effect of terrain using the terrain-following coordinate. For understanding the basic dynamics of the model, firstly, we describe the basic equations without terrain-following coordinates. Secondly, we describe those in the terrain-following coordinates.

6.1 The basic equations without terrain

The independent variables of the model are the distance coordinates, x, y, z and time, t. The dependent variables are horizontal velocity, u, v, vertical velocity, w, perturbation of potential temperature, θ', perturbation pressure, p', mixing ratio of water vapor, q_v, mixing ratios of hydrometeors, q_x, and number density of solid hydrometrors N_x. Here, The basic state of the potential temperature, pressure and density considering water contents are constant with time and satisfy the hydrostatic equilibrium,

$$\frac{\partial \bar{p}}{\partial z} = -\bar{\rho} g \tag{6.1}$$

The relation between the basic states and the deviation from them are given as,

$$\theta = \bar{\theta} + \theta' \tag{6.2}$$
$$p = \bar{p} + p' \tag{6.3}$$
$$\rho = \bar{\rho} + \rho' \tag{6.4}$$

The density is given by the equation of state diagnostically,

$$\rho = \frac{p}{R_d T}\left(1 - \frac{q_v}{\epsilon + q_v}\right)\left(1 + q_v + \sum q_x\right) \tag{6.5}$$

Here, g is the gravity acceleration, ϵ is the ratio of molecule number between water vapor and dry air and R_d is gas constant of dry air.

All of the dependent variables except density are described in time-developing equations. In the coordinates without terrain, the equations of the dependent variables are given as follows. The basic equations with terrain will be shown in Section 6.2.

Equation of motion

$$\frac{\partial \bar{\rho} u}{\partial t} = -\bar{\rho}\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) - \frac{\partial p'}{\partial x} + \bar{\rho}\left(f_s v - f_c w\right) + \mathrm{Turb}.u \tag{6.6}$$

$$\frac{\partial \bar{\rho} v}{\partial t} = -\bar{\rho}\left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) - \frac{\partial p'}{\partial y} - f_s \bar{\rho} u + \mathrm{Turb}.v \tag{6.7}$$

$$\frac{\partial \bar{\rho} w}{\partial t} = -\bar{\rho}\left(u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) - \frac{\partial p'}{\partial z} - \bar{\rho}\mathrm{Buoy}.w + f_c u + \mathrm{Turb}.w \tag{6.8}$$

Here, f_s, f_c is Coriolis-parameter and Buoy.w is the term of buoyancy.

Equation of pressure

$$\frac{\partial p'}{\partial t} = -\left(u\frac{\partial p'}{\partial x} + v\frac{\partial p'}{\partial y} + w\frac{\partial p'}{\partial z}\right) + \bar{\rho}gw$$
$$-\bar{\rho}c_s^2\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + \bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right) \tag{6.9}$$

Here, c_s is the speed of sound in atmosphere and $Q = 1 + 0.61q_v + \sum q_x$.

Equation of potential temperature

$$\frac{\partial \bar{\rho}\theta'}{\partial t} = -\bar{\rho}\left(u\frac{\partial \theta'}{\partial x} + v\frac{\partial \theta'}{\partial y} + w\frac{\partial \theta'}{\partial z}\right) - \bar{\rho}w\frac{\partial \bar{\theta}}{\partial z} + \text{Turb.}\theta + \bar{\rho}\text{Src.}\theta \tag{6.10}$$

Equations of water vapor and mixing ratio of wear contents

$$\frac{\partial \bar{\rho}q_v}{\partial t} = -\bar{\rho}\left(u\frac{\partial q_v}{\partial x} + v\frac{\partial q_v}{\partial y} + w\frac{\partial q_v}{\partial z}\right) + \text{Turb.}q_v + \bar{\rho}\text{Src.}q_v \tag{6.11}$$

$$\frac{\partial \bar{\rho}q_x}{\partial t} = -\bar{\rho}\left(u\frac{\partial q_x}{\partial x} + v\frac{\partial q_x}{\partial y} + w\frac{\partial q_x}{\partial z}\right) + \text{Turb.}q_x + \bar{\rho}\text{Src.}q_x + \bar{\rho}\text{Fall.}q_x \tag{6.12}$$

Equation of number density of water contents

$$\frac{\partial N_x}{\partial t} = -\bar{\rho}\left[u\frac{\partial}{\partial x}\left(\frac{N_x}{\bar{\rho}}\right) + v\frac{\partial}{\partial y}\left(\frac{N_x}{\bar{\rho}}\right) + w\frac{\partial}{\partial z}\left(\frac{N_x}{\bar{\rho}}\right)\right]$$
$$+\text{Turb.}\frac{N_x}{\bar{\rho}} + \bar{\rho}\text{Src.}\frac{N_x}{\bar{\rho}} + \bar{\rho}\text{Fall.}\frac{N_x}{\bar{\rho}} \tag{6.13}$$

The detail of other dependent variables are described in Section 6.2. The term of diffusion in sub-grid scale Turb.ϕ which appears except in equation of pressure is described in Chapter 7, and the source terms of cloud and precipitation Src.ϕ and falling of precipitation Fall.ϕ which appear in the equation of potential temperature and water contents are described in Chapter 8.

6.2 The basic equations system in terrain-following coordinates

6.2.1 General curvilinear coordinates

CReSS adopts the terrain-following coordinate to consider terrain effect. The basis of this coordinate system doesn't necessarily become to orthonormal, while the vector in Cartesian coordinate is expressed by orthogonal basis. This is categorized in curvilinear coordinate of liner algebra. Here, the basic background is summarized.

Contravariant components and covariant components

Introducing general linear independent bases $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, which are neither regular nor orthogonal, facultative vector $\mathbf{A}$ is expressed as the linear combination,

$$\mathbf{A} = A^1\mathbf{e}_1 + A^2\mathbf{e}_2 + A^3\mathbf{e}_3 \tag{6.14}$$

On the other hand, since the inverse bases $(\mathbf{f}^1, \mathbf{f}^2, \mathbf{f}^3)$, is defined as

$$\mathbf{f}^i \cdot \mathbf{e}_j = \delta^i_j \tag{6.15}$$

then, the components are,

$$\mathbf{f}^i \cdot \mathbf{A} = A^i \tag{6.16}$$

Here, Kronecker's delta is defined as follows,

$$\delta^i_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{6.17}$$

In the case of orthonormal basis, coordinate components are given by scalar product of basis vector and facultative vector. In the case of general basis, however, the components must be given by scalar product of inverse basis vector and facultative vector. Thus components of coordinates for original basis are named contravariant components and expressed in a superscript. On the other hand, components of coordinates for inverse basis are named covariant components, expressed in a subscript.

The facultative vector $\mathbf{B}$ is expressed by inverse basic vector as

$$\mathbf{B} = B_1\mathbf{f}^1 + B_2\mathbf{f}^2 + B_3\mathbf{f}^3 \tag{6.18}$$

The solution of scalar product of them becomes

$$\mathbf{A} \cdot \mathbf{B} = A^i B_i \tag{6.19}$$

Thus, scalar product is expressed easily by the sum of product of components in the relative coordinate when both contravariant and covariant components are used. In the case of orthonormal basis, as inverse basis becomes same to original basis, there are no difference between contravariant and covariant components.

$$\mathbf{A} \cdot \mathbf{B} = A^i B^i \tag{6.20}$$

Scalar product of facultative vector $\mathbf{A}$ and bases $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ gives covariant components for bases of vector $\mathbf{A}$, $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$,

$$A_i = \mathbf{A} \cdot \mathbf{e}_i \tag{6.21}$$

Next, for bases $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, 9 scalar product can be made,

$$\mathrm{G}_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j \tag{6.22}$$

This matrix is named metric matrix. Moreover, this is quadratic tensor which is named metric tensor. Using this, the relationship between contravariant and covariant components is expressed as follows.

$$A^i = \mathrm{G}_{ji} A_j \tag{6.23}$$

curvilinear coordinates

As a function of Cartesian coordinate, three function $F^i(x, y, z)$ which defined at a certain region of a space is considered. The differentiation of them for (x, y, z) is possible for any number of times.

$$\xi = F^1(x, y, z) \tag{6.24}$$
$$\eta = F^2(x, y, z) \tag{6.25}$$
$$\zeta = F^3(x, y, z) \tag{6.26}$$

(ξ, η, ζ) correlates to the each point in a region $P(x, y, z)$. When this correlation is 1 to 1, it can be said that curvilinear coordinates is made in a region. Furthermore, the following condition is assumed.

$$\frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \zeta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \xi} & \dfrac{\partial z}{\partial \eta} & \dfrac{\partial z}{\partial \zeta} \end{vmatrix} \neq 0 \tag{6.27}$$

For Cartesian coordinate, vector $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is

$$\mathbf{e}_1 = \begin{pmatrix} \dfrac{\partial x}{\partial \xi} \\ \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial z}{\partial \xi} \end{pmatrix}, \qquad \mathbf{e}_2 = \begin{pmatrix} \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial z}{\partial \eta} \end{pmatrix}, \qquad \mathbf{e}_3 = \begin{pmatrix} \dfrac{\partial x}{\partial \zeta} \\ \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \zeta} \end{pmatrix} \tag{6.28}$$

Here, vector $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is named 'basis' or 'fundamental vector' of curvilinear coordinates (ξ, η, ζ). Using above, facultative vector field $\mathbf{A}$ is expressed as the linear combinatio of them.

$$\mathbf{A} = A^1\mathbf{e}_1 + A^2\mathbf{e}_2 + A^3\mathbf{e}_3 \tag{6.29}$$

Then, A^i is named 'contravariant components' of curvilinear coordinates (ξ, η, ζ) of vector field $\mathbf{A}$. The covariant component is mentioned in (6.21). Samely, for the bases $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, 9 scalar product can be made.

$$\mathrm{G}_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j \tag{6.30}$$

Matrix G_{ij} is named metric matrix of curvilinear coordinates (ξ, η, ζ). The relationship between contravariant and covariant components is given by (6.23).

6.2.2 Terrain-following coordinates

Many cloud models adopt terrain-following coordinate system to consider the effect of terrain. Such coordinate system are oriented on the special relationship of curvilinear coordinates.

CReSS adopts the terrain-following coordinate as well as NHM and ARPS.

$$\xi = x \tag{6.31}$$
$$\eta = y \tag{6.32}$$
$$\zeta = \zeta\,(x, y, z) \tag{6.33}$$

In this case, velocity vector of Cartesian coordinate can be expressed as well as (6.29) by components of velocity vector (contravariant components) (U, V, W) in the terrain-following coordinate.(Ordinally, (U, V, W) are contravariant components and they should be expressed in the superscript (u^1, u^2, u^3). However, since it is easy, it is written in this way.)

$$u = U\frac{\partial x}{\partial \xi} + V\frac{\partial x}{\partial \eta} + W\frac{\partial x}{\partial \zeta} \tag{6.34}$$

$$v = U\frac{\partial y}{\partial \xi} + V\frac{\partial y}{\partial \eta} + W\frac{\partial y}{\partial \zeta} \tag{6.35}$$

$$w = U\frac{\partial z}{\partial \xi} + V\frac{\partial z}{\partial \eta} + W\frac{\partial z}{\partial \zeta} \tag{6.36}$$

Since it is assumed that condition (6.27) is realized, the inverse of the velocity vector can be asked by solving (6.34)∼(6.36) about (U, V, W).

$$UG^{\frac{1}{2}} = uJ^{yz}_{\eta\zeta} + vJ^{zx}_{\eta\zeta} + wJ^{xy}_{\eta\zeta} \tag{6.37}$$

$$VG^{\frac{1}{2}} = uJ^{yz}_{\zeta\xi} + vJ^{zx}_{\zeta\xi} + wJ^{xy}_{\zeta\xi} \tag{6.38}$$

$$WG^{\frac{1}{2}} = uJ^{yz}_{\xi\eta} + vJ^{zx}_{\xi\eta} + wJ^{xy}_{\xi\eta} \tag{6.39}$$

Here, J is Jacobian. For example, it is defined as

$$J^{yz}_{\eta\zeta} \equiv \frac{\partial\,(y,z)}{\partial\,(\eta,\zeta)} = \begin{vmatrix} \dfrac{\partial y}{\partial \eta} & \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \eta} & \dfrac{\partial z}{\partial \zeta} \end{vmatrix} \tag{6.40}$$

$G^{\frac{1}{2}}$ is defined as Jacobian of the coordinate conversion between (ξ, η, ζ) and (x, y, z) and expressed as

$$G^{\frac{1}{2}} \equiv \frac{\partial\,(x,y,z)}{\partial\,(\xi,\eta,\zeta)} = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \zeta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \xi} & \dfrac{\partial z}{\partial \eta} & \dfrac{\partial z}{\partial \zeta} \end{vmatrix} \tag{6.41}$$

In the case of terrain-following coordinate defined by (6.31)∼(6.33), Jacobians which appear in (6.37)∼(6.39) are as follows.

$$
\begin{aligned}
&J_{\eta\zeta}^{yz} = \frac{\partial z}{\partial \zeta}, \qquad J_{\eta\zeta}^{zx} = 0, \qquad J_{\eta\zeta}^{xy} = 0, \\
&J_{\zeta\xi}^{yz} = 0, \qquad J_{\zeta\xi}^{zx} = \frac{\partial z}{\partial \zeta}, \qquad J_{\zeta\xi}^{xy} = 0, \\
&J_{\xi\eta}^{yz} = -\frac{\partial z}{\partial \xi}, \qquad J_{\xi\eta}^{zx} = -\frac{\partial z}{\partial \eta}, \qquad J_{\xi\eta}^{xy} = 1
\end{aligned}
\tag{6.42}
$$

In the case of three-dimension, Jacobian of the coordinate conversion between (ξ, η, ζ) and (x, y, z) is

$$
G^{\frac{1}{2}} = \left| \frac{\partial z}{\partial \zeta} \right| \tag{6.43}
$$

Here, variable components of Jacobian's components are defined as

$$
J_{31} \equiv J_{\xi\eta}^{yz} = -\frac{\partial z}{\partial \xi} \tag{6.44}
$$

$$
J_{32} \equiv J_{\xi\eta}^{zx} = -\frac{\partial z}{\partial \eta} \tag{6.45}
$$

$$
J_d \equiv J_{\eta\zeta}^{yz} = J_{\zeta\xi}^{zx} = \frac{\partial z}{\partial \zeta} \tag{6.46}
$$

and ζ is defined by using the altitude of surface $z_{sfc}(x, y)$ and the height of the region of model z_{top},

$$
\zeta(x, y, z) = \frac{z_{top}[z - z_{sfc}(x, y)]}{z_{top} - z_{sfc}(x, y)} \tag{6.47}
$$

or

$$
z(\xi, \eta, \zeta) = z_{sfc}(\xi, \eta) + \zeta \left[1 - \frac{z_{sfc}(\xi, \eta)}{z_{top}} \right] \tag{6.48}
$$

In this case, Jacobian's various components are

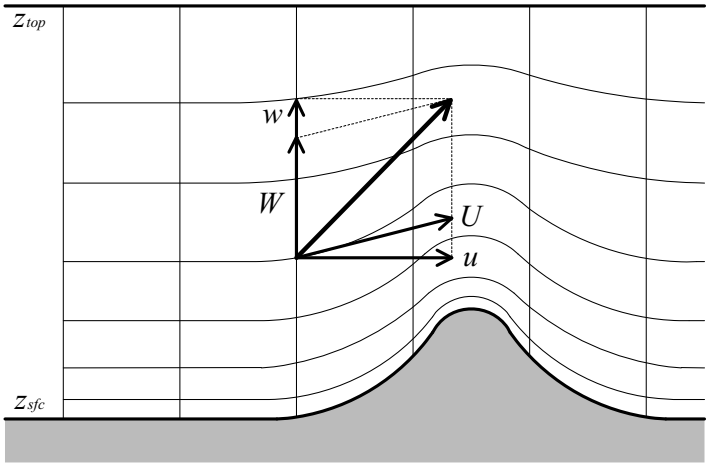

Figure 6.1. Terrain-following coordinates and direction of the vector.

$$J_{31} = -\frac{\partial z}{\partial \xi} = \left(\frac{\zeta}{z_{top}} - 1\right)\frac{\partial z_{sfc}(\xi,\eta)}{\partial \xi} \tag{6.49}$$

$$J_{32} = -\frac{\partial z}{\partial \eta} = \left(\frac{\zeta}{z_{top}} - 1\right)\frac{\partial z_{sfc}(\xi,\eta)}{\partial \eta} \tag{6.50}$$

$$J_d = \frac{\partial z}{\partial \zeta} = 1 - \frac{z_{sfc}(\xi,\eta)}{z_{top}} \tag{6.51}$$

Like in this case, when ζ is a monotonically increasing function about z, it is expressed that

$$G^{\frac{1}{2}} = |J_d| = J_d \tag{6.52}$$

Velocity of the terrain-following coordinate (contravariant velocity) which is given in (6.37)∼(6.39) is transformed as follows,

$$U = u \tag{6.53}$$

$$V = v \tag{6.54}$$

$$W = (uJ_{31} + vJ_{32} + w)\Big/G^{\frac{1}{2}} \tag{6.55}$$

When the transformation from Cartesian coordinate to the coordinate along terrain is performed using above relation, the space differential of a various value ϕ is transformed as follows,

$$\frac{\partial \phi}{\partial x} = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}(J_d\phi) + \frac{\partial}{\partial \zeta}(J_{31}\phi)\right] \tag{6.56}$$

$$\frac{\partial \phi}{\partial y} = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}(J_d\phi) + \frac{\partial}{\partial \zeta}(J_{32}\phi)\right] \tag{6.57}$$

$$\frac{\partial \phi}{\partial z} = \frac{1}{G^{\frac{1}{2}}} \frac{\partial \phi}{\partial \zeta} \tag{6.58}$$

6.2.3 The basic equations in terrain-following coordinates

In the coordinates of terrain-following, three dependent variables are separated into the values of base state and deviations from the base state: potential temperature, pressure and density which considered effects of water contents and water vapor. The values of base state are defined as those are in hydrostatic balance taken account of effects of terrain as

$$\frac{\partial \bar{\rho}}{\partial \zeta} = -G^{\frac{1}{2}} \bar{\rho} g \tag{6.59}$$

To simplify those representation, we use

$$\rho^* = G^{\frac{1}{2}} \bar{\rho} \tag{6.60}$$

and transform each predicted variable as follows:

$$u^* = \rho^* u \tag{6.61}$$
$$v^* = \rho^* v \tag{6.62}$$
$$w^* = \rho^* w \tag{6.63}$$
$$W^* = \rho^* W \tag{6.64}$$
$$\theta^* = \rho^* \theta' \tag{6.65}$$
$$q_v^* = \rho^* q_v \tag{6.66}$$
$$q_x^* = \rho^* q_x \tag{6.67}$$

Using the above, the basic equations system in terrain-excluding coordinates shown in Section 6.1 of this chapter are transformed in terrain-following coordinates as follows.

Equation of motion

$$\begin{aligned} \frac{\partial u^*}{\partial t} = & -\left(u^* \frac{\partial u}{\partial \xi} + v^* \frac{\partial u}{\partial \eta} + W^* \frac{\partial u}{\partial \zeta} \right) \\ & - \left[\frac{\partial}{\partial \xi} \left\{ J_d \left(p' - \alpha Div^* \right) \right\} + \frac{\partial}{\partial \zeta} \left\{ J_{31} \left(p' - \alpha Div^* \right) \right\} \right] + \left(f_s v^* - f_c w^* \right) + G^{\frac{1}{2}} \mathrm{Turb.}u \end{aligned} \tag{6.68}$$

$$\begin{aligned} \frac{\partial v^*}{\partial t} = & -\left(u^* \frac{\partial v}{\partial \xi} + v^* \frac{\partial v}{\partial \eta} + W^* \frac{\partial v}{\partial \zeta} \right) \\ & - \left[\frac{\partial}{\partial \eta} \left\{ J_d \left(p' - \alpha Div^* \right) \right\} + \frac{\partial}{\partial \zeta} \left\{ J_{32} \left(p' - \alpha Div^* \right) \right\} \right] - f_s u^* + G^{\frac{1}{2}} \mathrm{Turb.}v \end{aligned} \tag{6.69}$$

$$\frac{\partial w^*}{\partial t} = -\left(u^*\frac{\partial w}{\partial \xi} + v^*\frac{\partial w}{\partial \eta} + W^*\frac{\partial w}{\partial \zeta}\right) - \frac{\partial}{\partial \zeta}\left(p' - \alpha Div^*\right) - \rho^* \text{Buoy}.w + f_c u^* + G^{\frac{1}{2}}\text{Turb}.w \tag{6.70}$$

where the buoyancy term Buoy.w is expressed as

$$\text{Buoy}.w = -g\frac{\rho'}{\bar{\rho}} = g\left(\frac{\theta'}{\bar{\theta}} - \frac{p'}{\bar{\rho}c_s^2} + \frac{q_v'}{\epsilon + \bar{q}_v} - \frac{q_v' + \sum q_x}{1 + \bar{q}_v}\right) \tag{6.71}$$

Here q_v' is not the deviation from the value of base state but the deviation from the initial value, ϵ is the ratio of molecular weight to water vapor and dry air. c_s is the speed of sound in air given by

$$c_s = \sqrt{\gamma R_d \bar{T}}\ , \qquad \gamma \equiv C_p\,/C_v \tag{6.72}$$

where g is the gravity acceleration、$\bar{T}$ is the temperature of base state and R_d is the gas constant for dry air. C_p, C_v are the specific heat at constant pressure and the specific heat at constant volume for dry air, respectively. f_s, f_c are the Coriolis coefficients:

$$f_s = 2\omega \sin\varphi \tag{6.73}$$

$$f_c = 2\omega \cos\varphi \tag{6.74}$$

where ω is the angular velocity of the earth and φ is the latitude. Furthermore, αDiv^* shown in the pressure term is the divergence damping to suppress soundwaves, which is given by

$$Div^* = \frac{1}{G^{\frac{1}{2}}}\left(\frac{\partial u^*}{\partial \xi} + \frac{\partial v^*}{\partial \eta} + \frac{\partial W^*}{\partial \zeta}\right) \tag{6.75}$$

Equation of pressure

$$\frac{\partial G^{\frac{1}{2}}p'}{\partial t} = -\left(G^{\frac{1}{2}}u\frac{\partial p'}{\partial \xi} + G^{\frac{1}{2}}v\frac{\partial p'}{\partial \eta} + G^{\frac{1}{2}}W\frac{\partial p'}{\partial \zeta}\right) + G^{\frac{1}{2}}\bar{\rho}gw - \bar{\rho}c_s^2\left(\frac{\partial G^{\frac{1}{2}}u}{\partial \xi} + \frac{\partial G^{\frac{1}{2}}v}{\partial \eta} + \frac{\partial G^{\frac{1}{2}}W}{\partial \zeta}\right) + G^{\frac{1}{2}}\bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right) \tag{6.76}$$

where $Q = 1 + 0.61q_v + \sum q_x$ was used.

Equation of potential temperature

$$\frac{\partial \theta^*}{\partial t} = -\left(u^*\frac{\partial \theta'}{\partial \xi} + v^*\frac{\partial \theta'}{\partial \eta} + W^*\frac{\partial \theta'}{\partial \zeta}\right) - \bar{\rho}w\frac{\partial \bar{\theta}}{\partial \zeta} + G^{\frac{1}{2}}\mathrm{Turb}.\theta + \rho^*\mathrm{Src}.\theta \tag{6.77}$$

Equations of mixing ratio of water vapor and water contents

$$\frac{\partial q_v^*}{\partial t} = -\left(u^*\frac{\partial q_v}{\partial \xi} + v^*\frac{\partial q_v}{\partial \eta} + W^*\frac{\partial q_v}{\partial \zeta}\right) + G^{\frac{1}{2}}\mathrm{Turb}.q_v + \rho^*\mathrm{Src}.q_v \tag{6.78}$$

$$\frac{\partial q_x^*}{\partial t} = -\left(u^*\frac{\partial q_x}{\partial \xi} + v^*\frac{\partial q_x}{\partial \eta} + W^*\frac{\partial q_x}{\partial \zeta}\right) + G^{\frac{1}{2}}\mathrm{Turb}.q_x + \rho^*\mathrm{Src}.q_x + \rho^*\mathrm{Fall}.q_x \tag{6.79}$$

Equations of number concentration per unit volume

$$\begin{aligned}\frac{\partial G^{\frac{1}{2}}N_x}{\partial t} = &-\left[u^*\frac{\partial}{\partial \xi}\left(\frac{N_x}{\bar{\rho}}\right) + v^*\frac{\partial}{\partial \eta}\left(\frac{N_x}{\bar{\rho}}\right) + W^*\frac{\partial}{\partial \zeta}\left(\frac{N_x}{\bar{\rho}}\right)\right] \\ &+G^{\frac{1}{2}}\mathrm{Turb}.\frac{N_x}{\bar{\rho}} + \rho^*\mathrm{Src}.\frac{N_x}{\bar{\rho}} + \rho^*\mathrm{Fall}.\frac{N_x}{\bar{\rho}}\end{aligned} \tag{6.80}$$

Similar to the no terrain, there is the prognostic equation of turbulence kinetic energy E in addition to these equations. In these equations, Turb.ϕ is the diffusion term of sub-grid scale. Src.ϕ is the production or loss term and Fall.ϕ is the sedimentation term in the equations of potential temperature and water contents.

6.3 Basic Equations — Map Projection

The computational domain of a cloud-resolving model has been small enough to ignore the earth curvature. Map projection, therefore, was not necessary for a simulation using the cloud-resolving model. Recently, parallel computers rapidly developed and their performance and memory have increased. Computational dimensions of the cloud-resolving model became large and the earth curvature is not small enough to ignore. For instance, a computational domain of a typhoon simulation is large and the effect of the earth curvature should be taken into the computation. To consider the earth curvature, map projections are necessary. In this section, we introduce **the map factor** into the basic equations and map projections will enable to use in the CReSS model.

Longitude-latitude coordinates and conformal map projections are orthogonal curvilinear coordinates, if no topography is considered. When topography is considered, the vertical coordinate is no longer orthogonal to the horizontal coordinates. They are general curvilinear coordinates and tensor transformation is necessary. However, we take the following two steps to introduce map projection and terrain-following coordinates for simplicity. First, the equation system including map projection will be derived without topography. Then, the terrain-following coordinates will be introduced.

6.3.1 Basic equations with map projection — no topography

The longitude-latitude coordinates and conformal map projections with a vertical coordinate are 3-dimensional orthogonal curvilinear coordinates if no topography is considered. Assuming the horizontal coordinates are (ξ, η) and the vertical coordinate is z, the basic equations on a map projection will be derived.

Metric factors (stretching factors) are introduced as

$$h_1 = \frac{1}{m} \tag{6.81}$$

$$h_2 = \frac{1}{n} \tag{6.82}$$

$$h_3 = 1 \tag{6.83}$$

The metric factors m, n will be used as map factors. In general, they are called as the metric factors. For example, metric factors in the spherical coordinate are

$$m = \frac{1}{a\cos\phi} \tag{6.84}$$

$$n = \frac{1}{a} \tag{6.85}$$

where a is the radius of the earth and ϕ is latitude. If the metric factors are these forms, the basic equations which will be derived later are the equations on the spherical coordinate.

Now, we take a point P and a neighbor point. Their coordinates are (ξ, η, z) and $(\xi + d\xi, \eta + d\eta, z + dz)$. Then, the distances between the two points along the each coordinate ds_1, ds_2, ds_3 are

$$ds_1 = \frac{d\xi}{m} \tag{6.86}$$

$$ds_2 = \frac{d\eta}{n} \tag{6.87}$$

$$ds_3 = dz \tag{6.88}$$

Derivatives of tangential unit vector along the each coordinate are

$$\begin{aligned}
&\frac{\partial \mathbf{e}_1}{\partial \xi} = -n\frac{\partial}{\partial \eta}\left(\frac{1}{m}\right)\mathbf{e}_2 - \frac{1}{am}\mathbf{e}_3, && \frac{\partial \mathbf{e}_1}{\partial \eta} = m\frac{\partial}{\partial \xi}\left(\frac{1}{n}\right)\mathbf{e}_2, && \frac{\partial \mathbf{e}_1}{\partial \zeta} = 0,\\
&\frac{\partial \mathbf{e}_2}{\partial \xi} = n\frac{\partial}{\partial \eta}\left(\frac{1}{m}\right)\mathbf{e}_1, && \frac{\partial \mathbf{e}_2}{\partial \eta} = -\frac{1}{an}\mathbf{e}_3 - m\frac{\partial}{\partial \xi}\left(\frac{1}{n}\right)\mathbf{e}_1, && \frac{\partial \mathbf{e}_2}{\partial \zeta} = 0,\\
&\frac{\partial \mathbf{e}_3}{\partial \xi} = \frac{1}{am}\mathbf{e}_1, && \frac{\partial \mathbf{e}_3}{\partial \eta} = \frac{1}{an}\mathbf{e}_2, && \frac{\partial \mathbf{e}_3}{\partial \zeta} = 0
\end{aligned} \tag{6.89}$$

Using these formulas, the equations of momentum, the equation of thermodynamics, the equation of continuity of compressible fluid, the equations of mixing ratios of water vapor and hydrometeors, the equations of number density of solid hydrometeors are rewritten in the following part of this section.

The momentum equations with no topography (6.6)〜(6.8) will be derived using the metric factors of the orthogonal coordinates(ξ, η, z).

The velocity vectors $\mathbf{u}$ on the orthogonal coordinate (ξ, η, z) are represented as

$$\mathbf{u} = u\mathbf{e}_1 + v\mathbf{e}_2 + w\mathbf{e}_3 \tag{6.90}$$

Considering the relationships between distance and coordinate (6.86)〜(6.88), the each component of velocity will be

$$u = \frac{ds_1}{ddt} = \frac{1}{m}\frac{d\xi}{dt}, \tag{6.91}$$

$$v = \frac{ds_2}{ddt} = \frac{1}{n}\frac{d\eta}{dt}, \tag{6.92}$$

$$w = \frac{ds_3}{ddt} = \frac{dz}{dt}. \tag{6.93}$$

The total derivative on the map projection is

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \frac{d\xi}{dt}\frac{\partial}{\partial \xi} + \frac{d\eta}{dt}\frac{\partial}{\partial \eta} + \frac{dz}{dt}\frac{\partial}{\partial z}. \tag{6.94}$$

Using equations (6.91)〜(6.93), then,

$$\frac{d}{dt} = \frac{\partial}{\partial t} + mu\frac{\partial}{\partial \xi} + nv\frac{\partial}{\partial \eta} + w\frac{\partial}{\partial z}. \tag{6.95}$$

Consequently, time derivative of velocity vector (6.90) on the curvilinear coordinates is

$$\frac{d\mathbf{u}}{dt} = \frac{du}{dt}\mathbf{e}_1 + \frac{dv}{dt}\mathbf{e}_2 + \frac{dw}{dt}\mathbf{e}_3 + u\frac{d\mathbf{e}_1}{dt} + v\frac{d\mathbf{e}_2}{dt} + w\frac{d\mathbf{e}_3}{dt}. \tag{6.96}$$

Using (6.89) and (6.95), the time derivatives of the tangential unit vectors in (6.96) are modified as

$$u\frac{d\mathbf{e}_1}{dt}+v\frac{d\mathbf{e}_2}{dt}+w\frac{d\mathbf{e}_3}{dt}=-\mathbf{e}_1 mnv\left[v\frac{\partial}{\partial\xi}\left(\frac{1}{n}\right)-u\frac{\partial}{\partial\eta}\left(\frac{1}{m}\right)\right]+\mathbf{e}_1\frac{uw}{a}$$
$$+\mathbf{e}_2 mnu\left[v\frac{\partial}{\partial\xi}\left(\frac{1}{n}\right)-u\frac{\partial}{\partial\eta}\left(\frac{1}{m}\right)\right]+\mathbf{e}_1\frac{vw}{a}-\mathbf{e}_3\frac{u^2+v^2}{a}. \quad (6.97)$$

These terms appear since the unit vectors changes with the curvilinear coordinates. They are called as **curvature terms** or **metric terms**.
On the coordinates (ξ,η,z), each component (f_ξ,f_η,f_z) of the Coriolis force $2\boldsymbol{\Omega}$ are represented as

$$2\boldsymbol{\Omega}\times\mathbf{u}=\begin{vmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\\ 2\Omega_\xi & 2\Omega_\eta & 2\Omega_z\\ u & v & w\end{vmatrix}$$
$$=\mathbf{e}_1(f_\eta w-f_z v)+\mathbf{e}_2(f_z u-f_\xi w)+\mathbf{e}_3(f_\xi v-f_\eta u). \quad (6.98)$$

Using the relationship of the gradient of the scalar variable in the orthogonal coordinates, the pressure gradient force is

$$\nabla p'=m\frac{\partial p'}{\partial\xi}\mathbf{e}_1+n\frac{\partial p'}{\partial\eta}\mathbf{e}_2+\frac{\partial p'}{\partial z}\mathbf{e}_3. \quad (6.99)$$

Using the above formulations, the equations of momentum (6.6)∼(6.8) will be rewritten as

The equations of momentum

$$\bar\rho\frac{\partial u}{\partial t}=-\bar\rho\left(mu\frac{\partial u}{\partial\xi}+nv\frac{\partial u}{\partial\eta}+w\frac{\partial u}{\partial z}\right)$$
$$-m\frac{\partial p'}{\partial\xi}+\bar\rho(f_\eta w-f_z v)+\bar\rho mnv\left[v\frac{\partial}{\partial\xi}\left(\frac{1}{n}\right)-u\frac{\partial}{\partial\eta}\left(\frac{1}{m}\right)\right]-\bar\rho\frac{uw}{a}+\mathrm{Turb}.u, \quad (6.100)$$

$$\bar\rho\frac{\partial v}{\partial t}=-\bar\rho\left(mu\frac{\partial v}{\partial\xi}+nv\frac{\partial v}{\partial\eta}+w\frac{\partial v}{\partial z}\right)$$
$$-n\frac{\partial p'}{\partial\eta}+\bar\rho(f_z u-f_\xi w)-\bar\rho mnu\left[v\frac{\partial}{\partial\xi}\left(\frac{1}{n}\right)-u\frac{\partial}{\partial\eta}\left(\frac{1}{m}\right)\right]-\bar\rho\frac{vw}{a}+\mathrm{Turb}.v, \quad (6.101)$$

$$\bar\rho\frac{\partial w}{\partial t}=-\bar\rho\left(mu\frac{\partial w}{\partial\xi}+nv\frac{\partial w}{\partial\eta}+w\frac{\partial w}{\partial z}\right)$$
$$-\frac{\partial p'}{\partial z}-\bar\rho\mathrm{Buoy}.w+\bar\rho(f_\xi v-f_\eta u)+\bar\rho\frac{u^2+v^2}{a}+\mathrm{Turb}.w. \quad (6.102)$$

where $\bar{\rho} = \bar{\rho}(z)$ is density of the basic field and it is a function of only the vertical coordinate. The term Buoy.w is buoyancy term.

In the same manner, the equation of pressure perturbation (6.9), the equation of potential temperature perturbation (6.10), and the equations of water substances (6.11)〜(6.13) are rewritten as follows.

The equation of pressure perturbation

$$\begin{aligned}\frac{\partial p'}{\partial t} = &-\left(mu\frac{\partial p'}{\partial \xi} + nv\frac{\partial p'}{\partial \eta} + w\frac{\partial p'}{\partial z}\right) + \bar{\rho}gw \\ &-\bar{\rho}c_s^2\left[mn\left(\frac{\partial}{\partial \xi}\frac{u}{m} + \frac{\partial}{\partial \eta}\frac{v}{n}\right) + \frac{\partial w}{\partial z}\right] + \bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right)\end{aligned} \tag{6.103}$$

The equation of potential temperature perturbation

$$\bar{\rho}\frac{\partial \theta'}{\partial t} = -\bar{\rho}\left(mu\frac{\partial \theta'}{\partial \xi} + nv\frac{\partial \theta'}{\partial \eta} + w\frac{\partial \theta'}{\partial z}\right) - \bar{\rho}w\frac{\partial \bar{\theta}}{\partial z} + \bar{\rho}\mathrm{Src}.\theta + \mathrm{Turb}.\theta \tag{6.104}$$

The equations of mixing ratios of water vapor and hydrometeors

$$\bar{\rho}\frac{\partial q_v}{\partial t} = -\bar{\rho}\left(mu\frac{\partial q_v}{\partial \xi} + nv\frac{\partial q_v}{\partial \eta} + w\frac{\partial q_v}{\partial z}\right) + \bar{\rho}\mathrm{Src}.q_v + \mathrm{Turb}.q_v \tag{6.105}$$

$$\bar{\rho}\frac{\partial q_x}{\partial t} = -\bar{\rho}\left(mu\frac{\partial q_x}{\partial \xi} + nv\frac{\partial q_x}{\partial \eta} + w\frac{\partial q_x}{\partial z}\right) + \bar{\rho}\mathrm{Fall}.q_x + \bar{\rho}\mathrm{Src}.q_x + \mathrm{Turb}.q_x \tag{6.106}$$

The equations of number density of hydrometeors

$$\begin{aligned}\frac{\partial N_x}{\partial t} = &-\bar{\rho}\left[mu\frac{\partial}{\partial \xi}\left(\frac{N_x}{\bar{\rho}}\right) + nv\frac{\partial}{\partial \eta}\left(\frac{N_x}{\bar{\rho}}\right) + w\frac{\partial}{\partial z}\left(\frac{N_x}{\bar{\rho}}\right)\right] \\ &+\bar{\rho}\text{Src.}\frac{N_x}{\bar{\rho}} + \bar{\rho}\text{Fall.}\frac{N_x}{\bar{\rho}} + \text{Turb.}\frac{N_x}{\bar{\rho}}\end{aligned} \tag{6.107}$$

The equation of the turbulence kinetic energy is necessary to evaluate the turbulence term Turb.ϕ. The equation on the map projection will be shown in Section of 7.

6.3.2 Basic equations with map projection and terrain

In this section, the basic equations using terrain-following coordinate with map projections are described. Introducing the terrain-following coordinate, the coordinate system is not orthogonal coordinate but a general curvilinear coordinate. Because the map factor is not a function of the vertical coordinate, we take the same way to introduce the terrain-following vertical coordinate as shown in Section 6.2.

In the previous section, the metric factors m, n are a stretched factor. In this section, they are the map factors. Most map projections in meteorology are a conformal projection such as the polar stereographic projection, the Lambert conformal projection and Mercator projection. In the conformal projection, direction is correct and two metric factors in the horizontal directions are the same at any point on the map. That is

$$\left(h_1 = \frac{1}{m}\right) = \left(h_2 = \frac{1}{n}\right) \tag{6.108}$$

In general, the map factors are a function of longitude and latitude.

Using (6.91)〜(6.93), the total derivative on the conformal map projections is

$$\begin{aligned}\frac{d}{dt} &= \frac{\partial}{\partial t} + \frac{d\xi}{dt}\frac{\partial}{\partial \xi} + \frac{d\eta}{dt}\frac{\partial}{\partial \eta} + \frac{d\zeta}{dt}\frac{\partial}{\partial z} \\ &= \frac{\partial}{\partial t} + mu\frac{\partial}{\partial \xi} + mv\frac{\partial}{\partial \eta} + W\frac{\partial}{\partial \zeta}\end{aligned} \tag{6.109}$$

Then, the vertical velocity W is

$$\begin{aligned}W = \frac{d\zeta}{dt} &= mu\frac{\partial \zeta}{\partial \xi} + mv\frac{\partial \zeta}{\partial \eta} + w\frac{\partial \zeta}{\partial z} \\ &= \left[mu\left(-\frac{\partial z}{\partial \xi}\right) + mv\left(-\frac{\partial z}{\partial \eta}\right) + w\frac{\partial \zeta}{\partial z}\right]\frac{\partial \zeta}{\partial z} \\ &= \frac{1}{G^{\frac{1}{2}}}\left(muJ_{31} + mvJ_{32} + w\right)\end{aligned} \tag{6.110}$$

Derivatives of a dependent variable ϕ with distances (x, y, z) are

$$m\frac{\partial \phi}{\partial \xi} \rightarrow m\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(J_d\phi\right) + \frac{\partial}{\partial \zeta}\left(J_{31}\phi\right)\right], \tag{6.111}$$

$$m\frac{\partial \phi}{\partial \eta} \rightarrow m\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d\phi\right) + \frac{\partial}{\partial \zeta}\left(J_{32}\phi\right)\right], \tag{6.112}$$

$$\frac{\partial \phi}{\partial z} \rightarrow \frac{1}{G^{\frac{1}{2}}}\frac{\partial \phi}{\partial \zeta}. \tag{6.113}$$

Using these derivatives, the basic equations are rewritten as follows.

Equations of motion

$$\begin{aligned}\bar{\rho}\frac{\partial u}{\partial t} = &-\bar{\rho}\left(mu\frac{\partial u}{\partial \xi} + mv\frac{\partial u}{\partial \eta} + W\frac{\partial u}{\partial \zeta}\right) - m\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(J_d p'\right) + \frac{\partial}{\partial \zeta}\left(J_{31}p'\right)\right] \\ &+\bar{\rho}(f_\eta w - f_z v) + \bar{\rho}m^2 v\left[v\frac{\partial}{\partial \xi}\left(\frac{1}{m}\right) - u\frac{\partial}{\partial \eta}\left(\frac{1}{m}\right)\right] - \bar{\rho}\frac{uw}{a} + \mathrm{Turb}.u,\end{aligned} \tag{6.114}$$

$$\begin{aligned}\bar{\rho}\frac{\partial v}{\partial t} = &-\bar{\rho}\left(mu\frac{\partial v}{\partial \xi} + mv\frac{\partial v}{\partial \eta} + W\frac{\partial v}{\partial \zeta}\right) - m\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d p'\right) + \frac{\partial}{\partial \zeta}\left(J_{32}p'\right)\right] \\ &+\bar{\rho}(f_z u - f_\xi w) - \bar{\rho}m^2 u\left[v\frac{\partial}{\partial \xi}\left(\frac{1}{m}\right) - u\frac{\partial}{\partial \eta}\left(\frac{1}{m}\right)\right] - \bar{\rho}\frac{vw}{a} + \mathrm{Turb}.v,\end{aligned} \tag{6.115}$$

$$\begin{aligned}\bar{\rho}\frac{\partial w}{\partial t} = &-\bar{\rho}\left(mu\frac{\partial w}{\partial \xi} + mv\frac{\partial w}{\partial \eta} + W\frac{\partial w}{\partial \zeta}\right) \\ &-\frac{1}{G^{\frac{1}{2}}}\frac{\partial p'}{\partial \zeta} - \bar{\rho}\mathrm{Buoy}.w + \bar{\rho}(f_\xi v - f_\eta u) + \bar{\rho}\frac{u^2 + v^2}{a} + \mathrm{Turb}.w.\end{aligned} \tag{6.116}$$

Equation of the pressure perturbation

$$\begin{aligned}\frac{\partial p'}{\partial t} = &-\left(mu\frac{\partial p'}{\partial \xi} + mv\frac{\partial p'}{\partial \eta} + W\frac{\partial p'}{\partial \zeta}\right) + \bar{\rho}gw \\ &-\bar{\rho}c_s^2\frac{1}{G^{\frac{1}{2}}}\left[m^2\left(\frac{\partial}{\partial \xi}\frac{G^{\frac{1}{2}}u}{m} + \frac{\partial}{\partial \eta}\frac{G^{\frac{1}{2}}v}{m}\right) + \frac{\partial G^{\frac{1}{2}}w}{\partial \zeta}\right] + \bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right)\end{aligned} \tag{6.117}$$

Equation of the potential temperature perturbation

$$\bar{\rho}\frac{\partial \theta'}{\partial t} = -\bar{\rho}\left(mu\frac{\partial \theta'}{\partial \xi} + mv\frac{\partial \theta'}{\partial \eta} + W\frac{\partial \theta'}{\partial \zeta}\right) - \bar{\rho}w\frac{1}{G^{\frac{1}{2}}}\frac{\partial \bar{\theta}}{\partial \zeta} + \bar{\rho}\mathrm{Src}.\theta + \mathrm{Turb}.\theta. \tag{6.118}$$

Equations of mixing ratios of water substances

$$\bar{\rho}\frac{\partial q_v}{\partial t} = -\bar{\rho}\left(mu\frac{\partial q_v}{\partial \xi} + mv\frac{\partial q_v}{\partial \eta} + W\frac{\partial q_v}{\partial \zeta}\right) + \bar{\rho}\mathrm{Src}.q_v + \mathrm{Turb}.q_v, \tag{6.119}$$

$$\bar{\rho}\frac{\partial q_x}{\partial t} = -\bar{\rho}\left(mu\frac{\partial q_x}{\partial \xi} + mv\frac{\partial q_x}{\partial \eta} + W\frac{\partial q_x}{\partial \zeta}\right) + \bar{\rho}\mathrm{Fall}.q_x + \bar{\rho}\mathrm{Src}.q_x + \mathrm{Turb}.q_x. \tag{6.120}$$

Equations of number density of water substances

$$\begin{aligned}\frac{\partial N_x}{\partial t} = &-\bar{\rho}\left[mu\frac{\partial}{\partial \xi}\left(\frac{N_x}{\bar{\rho}}\right) + mv\frac{\partial}{\partial \eta}\left(\frac{N_x}{\bar{\rho}}\right) + W\frac{\partial}{\partial \zeta}\left(\frac{N_x}{\bar{\rho}}\right)\right] \\ &+\bar{\rho}\mathrm{Src}.\frac{N_x}{\bar{\rho}} + \bar{\rho}\mathrm{Fall}.\frac{N_x}{\bar{\rho}} + \mathrm{Turb}.\frac{N_x}{\bar{\rho}}.\end{aligned} \tag{6.121}$$

The above equations will be rewritten more simply using variables with the asterisk (6.61)〜(6.67) as well as using the following expression.

$$m^2\left[v\frac{\partial}{\partial \xi}\left(\frac{1}{m}\right) - u\frac{\partial}{\partial \eta}\left(\frac{1}{m}\right)\right] = u\frac{\partial m}{\partial \eta} - v\frac{\partial m}{\partial \xi} \tag{6.122}$$

Then, the basic equations are rewritten as follows.

Equations of motion

$$\begin{aligned}\frac{\partial u^*}{\partial t} = &-\left(mu^*\frac{\partial u}{\partial \xi} + mv^*\frac{\partial u}{\partial \eta} + W^*\frac{\partial u}{\partial \zeta}\right) - m\left[\frac{\partial}{\partial \xi}\left(J_d p'\right) + \frac{\partial}{\partial \zeta}\left(J_{31}p'\right)\right] \\ &+(f_\eta w^* - f_z v^*) + v^*\left[u\frac{\partial m}{\partial \eta} - v\frac{\partial m}{\partial \xi}\right] - u^*\frac{w}{a} + G^{\frac{1}{2}}\mathrm{Turb}.u,\end{aligned} \tag{6.123}$$

$$\begin{aligned}\frac{\partial v^*}{\partial t} = &-\left(mu^*\frac{\partial v}{\partial \xi} + mv^*\frac{\partial v}{\partial \eta} + W^*\frac{\partial v}{\partial \zeta}\right) - m\left[\frac{\partial}{\partial \eta}\left(J_d p'\right) + \frac{\partial}{\partial \zeta}\left(J_{32}p'\right)\right] \\ &+(f_z u^* - f_\xi w^*) - u^*\left[u\frac{\partial m}{\partial \eta} - v\frac{\partial m}{\partial \xi}\right] - v^*\frac{w}{a} + G^{\frac{1}{2}}\mathrm{Turb}.v,\end{aligned} \tag{6.124}$$

$$\begin{aligned}\frac{\partial w^*}{\partial t} = &-\left(mu^*\frac{\partial w}{\partial \xi} + mv^*\frac{\partial w}{\partial \eta} + W^*\frac{\partial w}{\partial \zeta}\right) \\ &-\frac{\partial p'}{\partial \zeta} - \rho^*\mathrm{Buoy}.w + (f_\xi v^* - f_\eta u^*) + \frac{u^*u + v^*v}{a} + G^{\frac{1}{2}}\mathrm{Turb}.w.\end{aligned} \tag{6.125}$$

Equation of the pressure perturbation

$$\begin{aligned}\frac{\partial G^{\frac{1}{2}}p'}{\partial t} = &-G^{\frac{1}{2}}\left(mu\frac{\partial p'}{\partial \xi} + mv\frac{\partial p'}{\partial \eta} + W\frac{\partial p'}{\partial \zeta}\right) + gw^* \\ &-\bar{\rho}c_s^2\left[m^2\left(\frac{\partial}{\partial \xi}\frac{G^{\frac{1}{2}}u}{m} + \frac{\partial}{\partial \eta}\frac{G^{\frac{1}{2}}v}{m}\right) + \frac{\partial G^{\frac{1}{2}}w}{\partial \zeta}\right] + G^{\frac{1}{2}}\bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right)\end{aligned} \tag{6.126}$$

Equation of the potential temperature perturbation

$$\frac{\partial \theta^*}{\partial t} = -\left(mu^* \frac{\partial \theta'}{\partial \xi} + mv^* \frac{\partial \theta'}{\partial \eta} + W^* \frac{\partial \theta'}{\partial \zeta} \right) - \bar{\rho} w \frac{\partial \bar{\theta}}{\partial \zeta} + \rho^* \mathrm{Src}.\theta + G^{\frac{1}{2}} \mathrm{Turb}.\theta \tag{6.127}$$

Equations of mixing ratios of water vapor and water substances

$$\frac{\partial q_v^*}{\partial t} = -\left(mu^* \frac{\partial q_v}{\partial \xi} + mv^* \frac{\partial q_v}{\partial \eta} + W^* \frac{\partial q_v}{\partial \zeta} \right) + \rho^* \mathrm{Src}.q_v + G^{\frac{1}{2}} \mathrm{Turb}.q_v, \tag{6.128}$$

$$\frac{\partial q_x^*}{\partial t} = -\left(mu^* \frac{\partial q_x}{\partial \xi} + mv^* \frac{\partial q_x}{\partial \eta} + W^* \frac{\partial q_x}{\partial \zeta} \right) + \rho^* \mathrm{Fall}.q_x + \rho^* \mathrm{Src}.q_x + G^{\frac{1}{2}} \mathrm{Turb}.q_x. \tag{6.129}$$

Equations of number density of solid water substances

$$\begin{aligned} \frac{\partial G^{\frac{1}{2}} N_x}{\partial t} = &-\left[mu^* \frac{\partial}{\partial \xi}\left(\frac{N_x}{\bar{\rho}}\right) + mv^* \frac{\partial}{\partial \eta}\left(\frac{N_x}{\bar{\rho}}\right) + W^* \frac{\partial}{\partial \zeta}\left(\frac{N_x}{\bar{\rho}}\right)\right] \\ &+\rho^* \mathrm{Src}.\frac{N_x}{\bar{\rho}} + \rho^* \mathrm{Fall}.\frac{N_x}{\bar{\rho}} + G^{\frac{1}{2}} \mathrm{Turb}.\frac{N_x}{\bar{\rho}} \end{aligned} \tag{6.130}$$

To suppress computational instability due to the acoustic waves, an artificial term with respect to pressure αDiv^* is used in calculation. The pressur term $p' - \alpha Div^*$ will be rewritten in map projection as

$$Div^* = \frac{1}{G^{\frac{1}{2}}} \left[m^2 \left(\frac{\partial}{\partial \xi} \frac{u^*}{m} + \frac{\partial}{\partial \eta} \frac{v^*}{m} \right) + \frac{\partial W^*}{\partial \zeta} \right]. \tag{6.131}$$

The equation of the turbulent kinetic energy, which is related with the turbulent term Turb.ϕ will be shown in Section 7.

6.3.3 Conformal map projections

Most map projections in meteorology are a conformal map projection. The stretching coefficients m, n are independent of direction in the conformal map projection. This means that $m = n$. The basic equations will be simpler.

CReSS can use the following three conformal map projections.

- **The stereographic projection** for middle and high latitudes.
- **The Lambert conformal projection** for middle latitude.
- **The Mercator projection** for lower latitudes.

Chapter 7

Subgrid Scale Diffusion

The atmosphere is represented by grid point value in a numerical simulation. However, air motions of smaller than the grid spacing always exist in the actual atmosphere. No matter how small grid spacing, such motions can exist. They are called the subgrid scale motions, which act as diffusion in general. In addition, they correspond to turbulence and are often called 'turbulence diffusion'.

Since the subgrid-scale motion exist even if we use very small grid sipaceing, prognostic equations of the subgrid-scale motion is necessary. For example, we can separate velocity into the grid-scale motion and deviations. In the equations of averaged variables, the second-degree correlation of deviation appears as unknown variables which is called Reynolds stress. Then, their prognostic equations are necessary. However, the third-degree correlation appears in the prognostic equations. As a result, the system of equations is not closed. Such problem is due to the non-linearity of turbulent flow. Kellar and Friedmann (1924) got first recognition on this problem, which is called 'closure problem'.

To solve the closure problem, the unknown variables should be expressed by known variables. In this chapter, we discuss the formulation of subgrid-scale motion using the following two manner.

- Modeling of unknown variables in the prognostic equations by dealing with two-degree correlation
- Modeling of the prognostic equations on the two-degree correlation represented by scalar, which can indicate averaged velocity, turbulence kinetic energy and turbulent flow of dissipation ratio through a concept of eddy viscosity

7.1 Parameteriaation of Turbulence Transport

There are various scale motion in atomosphere. The motion, which can be expressed with grid of numerical simulation, is called grid-scale motion, mean motion, subgrid scale motion or eddy motion.

To separate these motions, we assume that the field variables A, velocity, temperature, mixing ratio and so on, can be separated into mean field and deviation components.

$$A = \bar{A} + A'' \tag{7.1}$$

where the corresponding means are indicated by $\bar{}$ and the deviation components by $''$.

There are various ways when we average variables, but we do not show them in detail. For your information, average of deviations and average of products of two variabeles are given by

$$\overline{A''} = 0 \tag{7.2}$$
$$\overline{A\bar{B}} = \bar{A}\bar{B} + \overline{A''B''} \tag{7.3}$$

In other words, if you average products, you do not always get the variable which correspond with products of each averaged, and the second term of above equation appears. We apply them to x components in equations of motion. Here, to simplify we discuss incompressible fluid $\rho = const$, and we can obtain a variable consisting of mean and deviation components.

$$\begin{aligned}\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} + \bar{w}\frac{\partial \bar{u}}{\partial z} - f\bar{v} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x} \\ - \frac{1}{\rho}\left(\frac{\partial}{\partial x}\rho\overline{u''u''} + \frac{\partial}{\partial y}\rho\overline{u''v''} + \frac{\partial}{\partial z}\rho\overline{u''w''}\right) + \nu\nabla^2\bar{u}\end{aligned} \tag{7.4}$$

The terms $-\rho\overline{u''u''}, -\rho\overline{u''v''}, -\rho\overline{u''w''}$ in this equation represent stress by turbulent flow, which is called eddy stress or Reynolds stress. We can regard them as transport of momentum, so stress are made by trasport of momentum by eddy.

Similarly, the equations on potential temperature and mixing ratio are given by

$$\frac{\partial \bar{\theta}}{\partial t} + \bar{u}\frac{\partial \bar{\theta}}{\partial x} + \bar{v}\frac{\partial \bar{\theta}}{\partial y} + \bar{w}\frac{\partial \bar{\theta}}{\partial z} = -\frac{\partial}{\partial x}\overline{u''\theta''} - \frac{\partial}{\partial y}\overline{v''\theta''} - \frac{\partial}{\partial z}\overline{w''\theta''} \tag{7.5}$$

Even if you use grid, the variables indicated by prime cannot be expressed. If you use them in numerical simulation, you have to represent them with the variables indicated by overbar (mean

components). If you not, you cannot get effects of transport by eddy motion which has smaller scale than grid distance. The problem of turbulent parametarization is how we have to express such deviation by eddy, which is indicated by prime, by using limited grid. In addition, you need to pay attention to a difference between the definition of these variables indicated by $^-$ and the definition of variables indicated by $^-$ shown at the second chapter.

7.2 Eddy Viscosity Model

7.2.1 Formulation of Diffusion Term

In this section, we formulate diffusion term (the term of turbulence mixing) $G^{\frac{1}{2}}\mathrm{Turb}.\phi$. The diffusion term appears in Equations of motion (6.123)〜(6.125), Equation of potential temperature (6.127), Equations of mixing ratio of water vapor and hydrometeor (6.128) and Equations of number concentration per unit volume (6.130), which shown in the basic dynamical equations in terrain-following coordinates in the section 6.3.2.

The diffusion term in the equation of motions is expressed using stress tensor τ_{ij} as follows:

$$\begin{aligned} G^{\frac{1}{2}}\mathrm{Turb}.u &= G^{\frac{1}{2}}\left(\frac{\partial \tau_{11}}{\partial x}+\frac{\partial \tau_{12}}{\partial y}+\frac{\partial \tau_{13}}{\partial z}\right) \\ &= \frac{\partial}{\partial \xi}\left(J_d\tau_{11}\right)+\frac{\partial}{\partial \eta}\left(J_d\tau_{12}\right)+\frac{\partial}{\partial \zeta}\left(\tau_{13}+J_{31}\tau_{11}+J_{32}\tau_{12}\right) \end{aligned} \tag{7.6}$$

$$\begin{aligned} G^{\frac{1}{2}}\mathrm{Turb}.v &= G^{\frac{1}{2}}\left(\frac{\partial \tau_{21}}{\partial x}+\frac{\partial \tau_{22}}{\partial y}+\frac{\partial \tau_{23}}{\partial z}\right) \\ &= \frac{\partial}{\partial \xi}\left(J_d\tau_{21}\right)+\frac{\partial}{\partial \eta}\left(J_d\tau_{22}\right)+\frac{\partial}{\partial \zeta}\left(\tau_{23}+J_{31}\tau_{21}+J_{32}\tau_{22}\right) \end{aligned} \tag{7.7}$$

$$\begin{aligned} G^{\frac{1}{2}}\mathrm{Turb}.w &= G^{\frac{1}{2}}\left(\frac{\partial \tau_{31}}{\partial x}+\frac{\partial \tau_{32}}{\partial y}+\frac{\partial \tau_{33}}{\partial z}\right) \\ &= \frac{\partial}{\partial \xi}\left(J_d\tau_{31}\right)+\frac{\partial}{\partial \eta}\left(J_d\tau_{32}\right)+\frac{\partial}{\partial \zeta}\left(\tau_{33}+J_{31}\tau_{31}+J_{32}\tau_{32}\right) \end{aligned} \tag{7.8}$$

where stress tensor τ_{ij} consists of shear stress and Reynolds stress. Reynolds stress consists of fluctuation from averaged variables, so we need modeling in any way to the form with averaged variables. Reynolds stress can express in the form of gradient diffusion using viscosity coefficient from an analogy of shear stress.

$$\tau_{11} = \bar{\rho}\nu_{\tau h}\left(S_{11}-\frac{2}{3}Div\right) \tag{7.9}$$

$$\tau_{12} = \bar{\rho}\nu_{\tau h}S_{12} \tag{7.10}$$

$$\tau_{13} = \bar{\rho}\nu_{\tau v}S_{13} \tag{7.11}$$

$$\tau_{21} = \bar{\rho}\nu_{\tau h}S_{12} \tag{7.12}$$

$$\tau_{22} = \bar{\rho}\nu_{\tau h}\left(S_{22} - \frac{2}{3}Div\right) \tag{7.13}$$

$$\tau_{23} = \bar{\rho}\nu_{\tau v}S_{23} \tag{7.14}$$

$$\tau_{31} = \bar{\rho}\nu_{\tau h}S_{13} \tag{7.15}$$

$$\tau_{32} = \bar{\rho}\nu_{\tau h}S_{23} \tag{7.16}$$

$$\tau_{33} = \bar{\rho}\nu_{\tau v}\left(S_{33} - \frac{2}{3}Div\right) \tag{7.17}$$

where $\nu_{\tau h} and \nu_{\tau v}$ are horizontal and vertical eddy viscosity coefficients regarding kinetic momentum, respectively. The molecular viscosity coefficient of shear stress is so small to eddy viscosity coefficient that it can be neglected. S_{ij} is deformation rate tensor. In curvilinear coordinate system, it is given by

$$S_{11} = 2\frac{\partial u}{\partial x} = \frac{2}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(J_d u\right) + \frac{\partial}{\partial \zeta}\left(J_{31}u\right)\right] \tag{7.18}$$

$$S_{22} = 2\frac{\partial v}{\partial y} = \frac{2}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d v\right) + \frac{\partial}{\partial \zeta}\left(J_{32}v\right)\right] \tag{7.19}$$

$$S_{33} = 2\frac{\partial w}{\partial z} = \frac{2}{G^{\frac{1}{2}}}\frac{\partial w}{\partial \zeta} \tag{7.20}$$

$$S_{12} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d u\right) + \frac{\partial}{\partial \xi}\left(J_d v\right) + \frac{\partial}{\partial \zeta}\left(J_{32}u + J_{31}v\right)\right] \tag{7.21}$$

$$S_{13} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(J_d w\right) + \frac{\partial}{\partial \zeta}\left(u + J_{31}w\right)\right] \tag{7.22}$$

$$S_{23} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d w\right) + \frac{\partial}{\partial \zeta}\left(v + J_{32}w\right)\right] \tag{7.23}$$

and Div is divergence.

$$Div = \frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(G^{\frac{1}{2}}u\right) + \frac{\partial}{\partial \eta}\left(G^{\frac{1}{2}}v\right) + \frac{\partial}{\partial \zeta}\left(G^{\frac{1}{2}}W\right)\right] \tag{7.24}$$

The diffusion terms of potential temperature, mixing ration of hydrometeor and water vapor and the number concentration per unit volume are formalized by using ϕ as follows:

$$\begin{aligned} G^{\frac{1}{2}}\mathrm{Turb}.\phi &= G^{\frac{1}{2}}\left(\frac{\partial H_{\phi 1}}{\partial x} + \frac{\partial H_{\phi 2}}{\partial y} + \frac{\partial H_{\phi 3}}{\partial z}\right) \\ &= \frac{\partial}{\partial \xi}\left(J_d H_{\phi 1}\right) + \frac{\partial}{\partial \eta}\left(J_d H_{\phi 2}\right) + \frac{\partial}{\partial \zeta}\left(H_{\phi 3} + J_{31}H_{\phi 1} + J_{32}H_{\phi 2}\right) \end{aligned} \tag{7.25}$$

where $H_{\phi 1}, H_{\phi 2}$ and $H_{\phi 3}$ are the molecular diffusions of the corresponding scalar ϕ and turbulent fluxes in the x, y and z directions, respectively. They have forms of the gradient diffusion similar to velocity

$$H_{\phi 1} = \bar{\rho}\nu_{Hh}\frac{\partial \phi}{\partial x} = \bar{\rho}\nu_{Hh}\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \xi}\left(J_d\phi\right) + \frac{\partial}{\partial \zeta}\left(J_{31}\phi\right)\right] \tag{7.26}$$

$$H_{\phi 2} = \bar{\rho}\nu_{Hh}\frac{\partial \phi}{\partial y} = \bar{\rho}\nu_{Hh}\frac{1}{G^{\frac{1}{2}}}\left[\frac{\partial}{\partial \eta}\left(J_d\phi\right) + \frac{\partial}{\partial \zeta}\left(J_{32}\phi\right)\right] \tag{7.27}$$

$$H_{\phi 3} = \bar{\rho}\nu_{Hv}\frac{\partial \phi}{\partial z} = \bar{\rho}\nu_{Hv}\frac{1}{G^{\frac{1}{2}}}\frac{\partial \phi}{\partial \zeta} \tag{7.28}$$

where $\nu_{Hh} and \nu_{Hv}$ are horizontal and vertical eddy viscosity coefficients about scalar, respectively. The molecular diffusion coefficient is so small that it is neglected.

We have the modeling of Reynolds to the form with eddy viscosity coefficient. The manner is called eddy viscosity model, with which we can appreciate the eddy viscosity coefficients $\nu_{\tau h} and \nu_{\tau v} and the essy diffusion coefficients \nu_{Hh} and \nu_{Hv}$ appeard in avobe equations.

In following section, two eddy viscosity model are explained, which are actually used in ***CReSS***.

- One order closure of Smagorinsky
- One and a half order closure with turbulence kinetic energy

7.2.2 One order closure of Smagorinsky

Smagorinsky (1963) and Lilly (1962) give the eddy viscosity coefficient in the case it is isotropic vertically and horizontally, using $\nu_{\tau h} = \nu_{\tau v} = \nu_\tau$

$$\nu_\tau = \begin{cases} \left(C_S\Delta\right)^2\left(Def^2 - \dfrac{N^2}{Pr}\right), & \nu_\tau > 0 \\ 0, & \nu_\tau \leq 0 \end{cases} \tag{7.29}$$

where C_S is smagorinsky constant, $C_S = 0.21$ by Deardorff (1972a). Δ is averaged grid interval of numerical simulation.

$$\Delta = \left(\Delta x \Delta y \Delta z\right)^{\frac{1}{3}} \tag{7.30}$$

Def, which is measurement of transformation, can be obtained

$$Def^2 = \frac{1}{2}\left(S_{11}^2 + S_{22}^2 + S_{33}^2\right) + S_{12}^2 + S_{13}^2 + S_{23}^2 - \frac{2}{3}Div^2 \tag{7.31}$$

and

$$N^2 = \begin{cases} \dfrac{g}{G^{\frac{1}{2}}} \dfrac{\partial \ln \theta}{\partial \zeta}, & q_v < q_{vsw} \\ \dfrac{g}{G^{\frac{1}{2}}} \left[\dfrac{1 + \dfrac{\mathcal{L}_v q_{vsw}}{R_d T}}{1 + \dfrac{\mathcal{L}_v^2 q_{vsw}}{C_p R_v T^2}} \left(\dfrac{\partial \ln \theta}{\partial \zeta} + \dfrac{\mathcal{L}_v}{C_p T} \dfrac{\partial q_{vsw}}{\partial \zeta} \right) - \dfrac{\partial q_w}{\partial \zeta} \right], & q_v \geq q_{vsw} \end{cases} \tag{7.32}$$

where N is a constant Brunt-Väisälä frequency, and

$$Pr = \frac{\nu_\tau}{\nu_H}, \qquad \nu_{Hh} = \nu_{Hv} = \nu_H \tag{7.33}$$

where Pr is Turbulent Prandtl number. So we can obtain the eddy viscosity coefficient regarding scalar ϕ. g is gravity acceleration, T is temperature, $R_d and R_v$ are gas constants for dry air and wet air, respectively. C_p is specific heat at constant pressure and q_w is a sum of mixing ratio of molecular weight for water vapor, cloud liquid water and rainwater. With the equation of Tetens, mixing ratio of water saturation q_{vsw} is given by

$$q_{vsw} = \epsilon \frac{610.78}{p} \exp\left(17.269 \frac{T - 273.16}{T - 35.86}\right) \tag{7.34}$$

and latent heat for water evaporation $\mathcal{L}_v$ is given by

$$\mathcal{L}_v = 2.50078 \times 10^6 \left(\frac{273.16}{T}\right)^{\left(0.167 + 3.67 \times 10^{-4} T\right)} \tag{7.35}$$

where ϵ is the ratio of molecular weight of water vapor and of dry air.

Next, in the case it is anisotropic vertically and horizontally, the eddy viscosity coefficients of each direction are represented as

$$\nu_{\tau h} = \begin{cases} (C_S \Delta_h)^2 \left(Def^2 - \dfrac{N^2}{Pr}\right), & \nu_{\tau h} > 0 \\ 0, & \nu_{\tau h} \leq 0 \end{cases} \tag{7.36}$$

$$\nu_{\tau v} = \begin{cases} (C_S \Delta_v)^2 \left(Def^2 - \dfrac{N^2}{Pr}\right), & \nu_{\tau v} > 0 \\ 0, & \nu_{\tau v} \leq 0 \end{cases} \tag{7.37}$$

where Δ_h and Δ_v are given by

$$\Delta_h = (\Delta x \Delta y)^{\frac{1}{2}} \tag{7.38}$$

$$\Delta_v = \Delta z \tag{7.39}$$

Furthermore,

$$Pr = \frac{\nu_{\tau h}}{\nu_{Hh}} = \frac{\nu_{\tau v}}{\nu_{Hv}} \tag{7.40}$$

Thus, we obtain the eddy viscosity coefficients ν_{Hh}, ν_{Hv} of each direction regarding scalar ϕ, using the Turbulent Prandtl number Pr.

7.2.3 One and a half order closure with turbulence kinetic energy

With regard to one and a half order closure, we use prognostic equations on turbulence kinetic energy. We mark deviation from averaged flow each velosity component with $''$, so this turbulence kinetic energy is represented as

$$E = \frac{1}{2}\left(\overline{u''^2 + v''^2 + w''^2}\right) \tag{7.41}$$

and, the prognostic equations are given by

$$\begin{aligned}\frac{\partial \rho^* E}{\partial t} = &-\left(u^*\frac{\partial E}{\partial \xi} + v^*\frac{\partial E}{\partial \eta} + W^*\frac{\partial E}{\partial \zeta}\right) + \text{Buoy.}E + \rho^*\left(\frac{1}{2}\nu_E Def^2 - \frac{2}{3}EDiv\right) - \rho^*\frac{C_e}{l_h}E^{\frac{3}{2}} \\ &+ \left[\frac{\partial}{\partial \xi}\left(J_d H_{E1}\right) + \frac{\partial}{\partial \eta}\left(J_d H_{E2}\right) + \frac{\partial}{\partial \zeta}\left(H_{E3} + J_{31}H_{E1} + J_{32}H_{E2}\right)\right]\end{aligned} \tag{7.42}$$

Buoy.E which appeard in equations (7.42), is given as follows with the converting term of potential energy and kinetic energy.

$$\text{Buoy.}E = \begin{cases} -g\rho^*\nu_{Hv}\left(A\dfrac{\partial \theta_e}{\partial \zeta} - \dfrac{\partial q_{all}}{\partial \zeta}\right), & q_v \geq q_{vsw} \;\; or \;\; q_c + q_i > 0 \\ -g\rho^*\nu_{Hv}\left(\dfrac{1}{\theta}\dfrac{\partial \theta}{\partial \zeta} + 0.61\dfrac{\partial q_v}{\partial \zeta}\right), & q_v < q_{vsw} \;\; or \;\; q_c + q_i = 0 \end{cases} \tag{7.43}$$

where, A is expressed as

$$A = \frac{1}{\theta}\frac{1 + \dfrac{1.61\epsilon\mathcal{L}_v q_v}{R_d T}}{1 + \dfrac{\epsilon\mathcal{L}_v^2 q_v}{C_p R_d T^2}} \tag{7.44}$$

where, g is gravity acceleration, T is temperature, ϵ is the ratio of molecular weight for water vapor and dry air, q_{all} is a sum of mixing ratio of molecular weight for water vapor, cloud liquid

water and cloud ice, and θ_e is equivalent potential temperature. C_pandR_d are gas constants for dry air and wet air, respectively. $\mathcal{L}_v$ is latent heat for water evaporation.

Next, the third term of right hand Def, Div is shown at the section 7.2.2. The coefficient C_e of the forth term, the dissipating term is represented as

$$C_e = \begin{cases} 3.9, & thelowestlayer \\ 0.93, & theotherlayer \end{cases} \tag{7.45}$$

In addition, the last term of right hand, flux of turbulence kinetic energy is given by

$$H_{E1} = \bar{\rho}\nu_E \frac{\partial E}{\partial x} = \bar{\rho}\nu_E \frac{1}{G^{\frac{1}{2}}} \left[\frac{\partial}{\partial \xi} (J_d E) + \frac{\partial}{\partial \zeta} (J_{31} E) \right] \tag{7.46}$$

$$H_{E2} = \bar{\rho}\nu_E \frac{\partial E}{\partial y} = \bar{\rho}\nu_E \frac{1}{G^{\frac{1}{2}}} \left[\frac{\partial}{\partial \eta} (J_d E) + \frac{\partial}{\partial \zeta} (J_{32} E) \right] \tag{7.47}$$

$$H_{E3} = \bar{\rho}\nu_E \frac{\partial E}{\partial z} = \bar{\rho}\nu_E \frac{1}{G^{\frac{1}{2}}} \frac{\partial E}{\partial \zeta} \tag{7.48}$$

where ν_E is the eddy viscosity coefficient for turbulence kinetic energy.

Thus, the eddy viscosity coefficients $\nu_{\tau h}, \nu_{\tau v}$ are represented as a function of turbulent kinetic energy E,

$$\nu_{\tau h} = 0.1 E^{\frac{1}{2}} l_h \tag{7.49}$$

$$\nu_{\tau v} = 0.1 E^{\frac{1}{2}} l_v \tag{7.50}$$

where l_handl_v are horisontal and vertical mixing length scales. In the case where the difference is small between horisontal and vertical grid interval,

$$l = l_h = l_v = \begin{cases} \Delta s, & unstableorneutral \\ min\left(\Delta s, l_s\right), & stable \end{cases} \tag{7.51}$$

where $\$Deltas$ and l_s are expressed as

$$\Delta s = \Delta s_h = \Delta s_v = (\Delta x \Delta y \Delta z)^{\frac{1}{3}} \tag{7.52}$$

$$l_s = 0.76 E^{\frac{1}{2}} \left| \frac{g}{\bar{\theta}} \frac{\partial \bar{\theta}}{\partial z} \right|^{-\frac{1}{2}} \tag{7.53}$$

On the other hand, in the case where the difference is large between horisontal and vertical grid interval,

$$l_h = \Delta s_h \tag{7.54}$$

$$l_v = \begin{cases} \Delta s_v, & unstable or neutral \\ min\left(\Delta s_v, l_s\right), & stable \end{cases} \tag{7.55}$$

where Δs_h and Δs_v are expressed as

$$\Delta s_h = (\Delta x \Delta y)^{\frac{1}{2}} \tag{7.56}$$

$$\Delta s_v = \Delta z \tag{7.57}$$

Finally, the eddy viscosity coefficients $\nu_{\tau h} and \nu_{\tau v}$ are represented as a function of turbulence kinetic energy E,

$$\nu_{\tau h} = max\left(0.1E^{\frac{1}{2}} l_h, \alpha \Delta s_h^2\right) \tag{7.58}$$

$$\nu_{\tau v} = max\left(0.1E^{\frac{1}{2}} l_v, \alpha \Delta s_v^2\right) \tag{7.59}$$

where α is smaller number like $\alpha = 10^{-6}$. The eddy viscosity coefficient ν_{Hv}, ν_{Hh} for scalar ϕ and ν_E for the turbulence kinetic energy E can be obtained as follows:

In the case of the small difference between horisontal and vertical grid interval,

$$\frac{\nu_{\tau h}}{\nu_{Hh}} = \frac{\nu_{\tau v}}{\nu_{Hv}} = \frac{1}{1 + 2l/\Delta s} \ \left(\ = Pr \ \right) \tag{7.60}$$

$$\nu_E = 2\nu_{\tau h} = 2\nu_{\tau v} \tag{7.61}$$

In the case of the large difference between horisontal and vertical grid interval,

$$\frac{\nu_{\tau h}}{\nu_{Hh}} = \frac{1}{1 + 2l_h/\Delta s_h} = \frac{1}{3} \tag{7.62}$$

$$\frac{\nu_{\tau v}}{\nu_{Hv}} = \frac{1}{1 + 2l_v/\Delta s_v} \ \left(\ = Pr \ \right) \tag{7.63}$$

$$\nu_E = 2\nu_{\tau h} \tag{7.64}$$

Chapter 8

Physical Processes of Cloud and Precipitation

Cloud physics is largely divided into cloud microphysics and cloud dynamics. Because they are closely related each other, knowledge of cloud physics is necessary for understanding cloud dynamics.

Ascending a humid air parcel in atmosphere, clouds generate in association with a conversion from water vapor to cloud and precipitation particles and precipitation is induced. We generally classify the formation processes of precipitation into "warm rain" and "cold rain (ice-phase rain)." The "warm rain" is rainfall which has no ice-phase process, and clouds form below the altitude of 0°C. Such clouds is called "warm clouds." The "cold rain" is rainfall of which ice-phase processes considerably contribute for the growth of precipitation particles. Such clouds is called "cold clouds." The clouds form not only below the altitude of 0°C but also above the altitude. The clouds have both a liquid phase and an ice-phase. There are the following categories in modeling of these clouds.

- Parametarization of "warm rain" with a bulk method
- Parametarization containing an ice-phase with a bulk method
- A method predicting time development of size distribution with dividing the sizes of liquid-phase particles into several bins (a bin method)
- A bin method as same as the above but containing an ice-phase
- A bin method containing categorized aerosols
- A method of which a bin method is used for a liquid phase and a bulk method for an ice-phase (A hybrid type)

Bulk method parametarizations of "warm rain" and "cold rain" are used in the ***CReSS***. Hereafter, modelings of these two methods are described.

8.1 Parametarization of "Warm Rain" with a Bulk Method

8.1.1 Equations for "Warm Rain" Process of Cloud and Precipitation

The parametarization of "warm rain" with a bulk method has the following three categories of water substance[1].

Sign	Meaning	Contents
q_v	Mixing ratio of water vapor	Water which exists in the atmosphere in the state of gas
q_c	Mixing ratio of cloud water	Its fall velocity can be regarded as zero. It corresponds to cloud drops in the actual atmosphere. The cloud drop is ordinary a small particle of which its diameter is less than 100 μm.
q_r	Mixing ratio of rain water	It corresponds to raindrops, which have a meaningful fall velocity, in the actual atmosphere.

Modeling variables involved in the "warm rain" process are three variables for water substance [kg kg^{-1}] and potential temperature (temperature) [K]. Using equations shown in section 6.1(we can explain by using the equations in which orography is not considered), the equations for the "warm rain" process are simply written as follows.

$$\frac{\partial \bar{\rho}\theta}{\partial t} = \mathrm{Adv}.\theta + \mathrm{Turb}.\theta - \bar{\rho}w\frac{\partial \bar{\theta}}{\partial z} + \frac{\bar{\rho}\mathcal{L}_v}{C_p\Pi}\left(CN_{vc} - EV_{cv} - EV_{rv}\right) \tag{8.1}$$

$$\frac{\partial \bar{\rho}q_v}{\partial t} = \mathrm{Adv}.q_v + \mathrm{Turb}.q_v - \bar{\rho}\left(CN_{vc} - EV_{cv} - EV_{rv}\right) \tag{8.2}$$

$$\frac{\partial \bar{\rho}q_c}{\partial t} = \mathrm{Adv}.q_c + \mathrm{Turb}.q_c + \bar{\rho}\left(CN_{vc} - EV_{cv} - CN_{cr} - CL_{cr}\right) \tag{8.3}$$

$$\frac{\partial \bar{\rho}q_r}{\partial t} = \mathrm{Adv}.q_r + \mathrm{Turb}.q_r + \bar{\rho}\left(CN_{cr} + CL_{cr} - EV_{rv}\right) + \frac{\partial}{\partial z}\left(\bar{\rho}U_r q_r\right) \tag{8.4}$$

where Adv.ϕ and Turb.ϕ represent the terms of advection and subgrid-scale turbulance. $\mathcal{L}_v$, C_p and Π are latent heat of evaporation of water [J kg^{-1}], specific heat at constant pressure in a dry air [J K kg^{-1}] and the Exner function, respectively. The last term of the right side of Eq. (8.4) represents a flux divegence of q_r in association with falling of rainwater. Microphisical processes considered here are as follows.

Sign	Contents
CN_{vc}	Conversion from water vapor to cloud water by condensation (condensation)

[1]Rain water which has a diameter of 0.1-0.5 mm is sometimes called drizzle. Here, drizzle is contained in a category of rain.

EV_{cv}	Conversion from cloud water to water vapor by evaporation (evaporation)
EV_{rv}	Conversion from rain water to water vapor by evaporation (evaporation)
CN_{cr}	Conversion from cloud water to rain water by coalescence growth. It corresponds to growing from a cloud droplet to the size of a raindrop by coalescence or diffusion of water vapor (autoconversion).
CL_{cr}	Conversion from cloud water to rain water by collision. A process of which a large waterdrop captures a small waterdrop (collection).

All of these amounts are defined as a positive value. A process of condensation from water vapor to rain water is neglected. Calculations of those processes are shown in the next section.

8.1.2 Microphysical Processes

Conversion between Water Vapor and Cloud Water: $-CN_{vc} + EV_{cv}$

As the same as Klemp and Wilhelmson (1978), we use the method of the moist saturation adjustment by Soong and Ogura (1973). This method will be explained in Section 8.2.5.

Saturation Mixing Ratio (Tetens Equation): q_{vsw}

Saturation mixing ratio over water q_{vsw} is represented by using the Tetens equation.

$$q_{vsw} = \epsilon \frac{610.78}{p} \exp\left(17.269 \frac{\Pi\theta - 273.16}{\Pi\theta - 35.86}\right) \tag{8.5}$$

Here, ϵ is the ratio with the molecular weight of water vapor and the molecular weight of dry air.

Conversion from Cloud Water to the Rain Water: CN_{cr}, CL_{cr}

The conversion from cloud water to rain water with coalescence growth (CN_{cr}) and conversion from cloud water to rain water (CL_{cr}) is calculated by using the Kessler (1969)'s parameterization.

$$CN_{cr} = k_1 (q_c - a) \tag{8.6}$$
$$CL_{cr} = k_2 q_c q_r^{0.875} \tag{8.7}$$

where

$$k_1 = 0.001 \quad [\mathrm{s}^{-1}] \tag{8.8}$$
$$a = 0.001 \quad [\mathrm{kg\ kg}^{-1}] \tag{8.9}$$
$$k_2 = 2.2 \quad [\mathrm{s}^{-1}]. \tag{8.10}$$

Evaporation of Rain Water: EV_{rv}

In the same way as Ogura and Takahashi (1971) and Klemp and Wilhelmson (1978), evaporation of rain water is represented as a following.

$$EV_{rv} = \frac{1}{\bar{\rho}} \frac{\left(1 - q_v / q_{vsw}\right) C \left(\bar{\rho} q_r\right)^{0.525}}{5.4 \times 10^5 + 2.55 \times 10^6 / \left(p q_{vsw}\right)} \tag{8.11}$$

where C is a ventilation factor given by

$$C = 1.6 + 124.9 \left(\bar{\rho} q_r\right)^{0.2046} . \tag{8.12}$$

Fall Velocity of Rain: U_r

Adding change of density to the equation of Soong and Ogura (1973), fall velocity U_r of the last term in the right side of the equation (8.4) is given by

$$U_r = 36.34 \left(\bar{\rho} q_r\right)^{0.1346} \left(\frac{\rho_0}{\bar{\rho}}\right) . \tag{8.13}$$

where ρ_0 is density [kg m^{-3}] at the surface in the basic state and the unit of U_r is [m s^{-1}]. Precipitation at the surface is calculated by using this fall velocity. Note that a difference in z^* cordinate system (ζ cordinate system) is multiplied by a matrix, as shown in equations (6.58).

8.2 Parameterization of the bulk cold rain

8.2.1 Parameterization of the cold rain

We formulate the parameterization of the cloud and precipitation processes containing ice-phase with a bulk method which are used in a cloud model. Here, conversions of water substance and changes of temperature and mixing ratio of water vapor by the conversions are considered. When we use a bulk method, the water in the atmosphere is divided into some categories (e.g. rain, snow and graupel) The categories are formulized with the typical variables (usually, mixing ratio or mixing ratio and number concentration), and their time developments are calculated. Therefore, when we use a bulk method, the definition of each variable must be clear. The way of categorization and variables are different in each model.

The following physical processes are considered in formulization of parameterization containing an ice-phase. In the formulization, the type of a particle must be considered.

- Primary and secondary nucleation of ice crystals
- Growth and decline of the particle with water vapor diffusion.
- Growth with collision between particles

- Breakup of a particle (breakup of a raindrop)
- Conversion to another category (Conversion from the cloud water to rain water, and from cloud water to snow, and from snow to graupel. etc.)
- Freezing and melting
- Shedding of un-freezing water
- Gravity falling

Although the definitions of the cloud physical valiables and its treatment are different in each model, we show two types of models containing an ice-phase in the following.

- A model which calculates only the time development equation of the mixing ratio of each category
- A model which calculates the graupel, cloud ice, snow, and their number concentrations adding the above equation

Hereafter, we summarize the formulizations by Murakami (1999), Murakami et al. (1994) and Murakami (1990).

We consider the six categories for the physical processes of cloud and precipitation: water vapor, cloud water, rain water, cloud ice, snow and graupel. In addition to the six categories, there is a model which makes fog water and hail another categories.

Sign	Meaning	Contents
θ	Potential temperture	$\theta = \bar{\theta} + \theta'$
q_v	Mixing ratio of water vapor	Water which exists in the atmosphere in the state of gas.
q_c	Mixing ratio of cloud water	Liquid water of which a particle diameter is small and fall velocity is negligible. It can move with atmospheric motion.
q_r	Mixing ratio of rain water	Usually, the particle of a liquid water with a diameter more than 100 μm is called rain. It is expressed as "rain water" in a model. Fall velocity depending on the size of a particle is meaningful. Although it moves in association with the atmospheric motion in horizontal, it falls and drops out of an air mass in vertical.
q_i	Mixing ratio of cloud ice	A minute crystal of ice which is called "ice crystal" in cloud physics. Usually, a diameter is less than 100 μm . Fall velocity is so small that it is negligible.

q_s	Mixing ratio of snow	In a model, it means a solid precipitation particle with density of about 0.1 g cm^{-3} and fall velocity of about 1 m s^{-1} . It corresponds to a snow crystal or snow flakes, etc.
q_g	Mixing ratio of graupel	In a model, it means a solid precipitation particle with density of about 0.4 g cm^{-3} and fall velocity of about 1∼4 m s^{-1} . It corresponds to a snow crystal with cloud droplets, snow flakes with cloud droplets, graupel, etc.
q_h	Mixing ratio of hail	In a model, it means a solid precipitation particle with density of about 0.9 g cm^{-3} and fall velocity of about 10 m s^{-1} fall speed.In an actual cloud particle, frozen rain, hail, etc. correspond. hail is contained in graupel in CReSS.
N_i	number concentration of cloud ice	Expressing cloud ice in a model, number concentration is sometimes regarded as a variable.
N_s	number concentration of snow	The same in case of snow.
N_g	number concentration of graupel	The same in case of graupel.

Here, the units of potential temperature, mixing ratio and number concentration are [K], [kg kg^{-1}] and [m^{-3}], respectively. We often use [g kg^{-1}] as the unit of mixing ratio.

8.2.2 The Equation System for Cloud and Precipitation Processes

The time development equations of potential temperature, water vapor and particles of cloud and precipitation are used here. Considering number concentrations of particles, the time development equations of number concentration of cloud ice, snow, and graupel are needed. These describe briefly the equation system shown in Section 6.1.

The time development equation of a potential temperature and a mixing ratio of water is expressed as

$$\frac{\partial \bar{\rho}\theta}{\partial t} = \mathrm{Adv}.\theta + \mathrm{Turb}.\theta - \bar{\rho} w \frac{\partial \bar{\theta}}{\partial z} + \bar{\rho}\left(\mathrm{Src}.\theta_V + \mathrm{Src}.\theta_S + \mathrm{Src}.\theta_F\right) \tag{8.14}$$

$$\frac{\partial \bar{\rho} q_v}{\partial t} = \text{Adv.}q_v + \text{Turb.}q_v + \bar{\rho}\text{Src.}q_v \tag{8.15}$$

$$\frac{\partial \bar{\rho} q_c}{\partial t} = \text{Adv.}q_c + \text{Turb.}q_c + \bar{\rho}\text{Src.}q_c + \bar{\rho}\text{Fall.}q_c \tag{8.16}$$

$$\frac{\partial \bar{\rho} q_r}{\partial t} = \text{Adv.}q_r + \text{Turb.}q_r + \bar{\rho}\text{Src.}q_r + \bar{\rho}\text{Fall.}q_r \tag{8.17}$$

$$\frac{\partial \bar{\rho} q_i}{\partial t} = \text{Adv.}q_i + \text{Turb.}q_i + \bar{\rho}\text{Src.}q_i + \bar{\rho}\text{Fall.}q_i \tag{8.18}$$

$$\frac{\partial \bar{\rho} q_s}{\partial t} = \text{Adv.}q_s + \text{Turb.}q_s + \bar{\rho}\text{Src.}q_s + \bar{\rho}\text{Fall.}q_s \tag{8.19}$$

$$\frac{\partial \bar{\rho} q_g}{\partial t} = \text{Adv.}q_g + \text{Turb.}q_g + \bar{\rho}\text{Src.}q_g + \bar{\rho}\text{Fall.}q_g \tag{8.20}$$

where characters of $v, c, r, i, s,$ and g, which are attached at the bottom, express cloud water, rain water, cloud ice, snow, and graupel, respectively. Hereafter, we sometimes use x or y as representative characters.

The meaning of each term is shown below.

Adv.ϕ	Advection term of potential temperature or mixing ratio of water substance
Turb.ϕ	Diffusion term of potential temperature or mixing ratio of water substance by a subgrid-scale turbulance
Src.θ_V	Source term of potential temperature in association with condensation and evaporation
Src.θ_S	Source term of potential temperature in association with sublimation
Src.θ_F	Source term of potential temperature in association with freezeing and melting
Src.q_x	Source term of the mixing ratio of water
Fall.q_x	Term of falling water substance (precipitation)

The time development equations of number concentrations of cloud ice, snow, and graupel are

$$\frac{\partial N_i}{\partial t} = \text{Adv.}\frac{N_i}{\bar{\rho}} + \text{Turb.}\frac{N_i}{\bar{\rho}} + \bar{\rho}\text{Src.}\frac{N_i}{\bar{\rho}} + \bar{\rho}\text{Fall.}\frac{N_i}{\bar{\rho}} \tag{8.21}$$

$$\frac{\partial N_s}{\partial t} = \text{Adv.}\frac{N_s}{\bar{\rho}} + \text{Turb.}\frac{N_s}{\bar{\rho}} + \bar{\rho}\text{Src.}\frac{N_s}{\bar{\rho}} + \bar{\rho}\text{Fall.}\frac{N_s}{\bar{\rho}} \tag{8.22}$$

$$\frac{\partial N_g}{\partial t} = \text{Adv.}\frac{N_g}{\bar{\rho}} + \text{Turb.}\frac{N_g}{\bar{\rho}} + \bar{\rho}\text{Src.}\frac{N_g}{\bar{\rho}} + \bar{\rho}\text{Fall.}\frac{N_g}{\bar{\rho}} \tag{8.23}$$

. where characters of i, s, g, which are attached at the bottom, express water vapor, cloud ice, snow, and graupel, respectively. Hereafter, we sometimes use x or y as representative characters.

The meaning of each term is shown below.

Adv. $N_x/\bar{\rho}$	Advection term of number concentration of water in the solid state
Turb. $N_x/\bar{\rho}$	Diffusion term of number concentration of water in the solid state by a subgrid-scale turbulence
Src. $N_x/\bar{\rho}$	Source term of the number concentration of water in the solid state
Fall. $N_x/\bar{\rho}$	Term of change of number concentration of water in the solid state in association with precipitation

Source terms of those equations are as follows.

Source term of the equation (8.14) of potential temperature θ: $\mathrm{Src}.\theta_V + \mathrm{Src}.\theta_S + \mathrm{Src}.\theta_F$

$$\mathrm{Src}.\theta_V = \frac{\mathcal{L}_v}{C_p \Pi} \, VD_{vr} \tag{8.24}$$

$$\mathrm{Src}.\theta_S = \frac{\mathcal{L}_s}{C_p \Pi} \left(NUA_{vi} + VD_{vi} + VD_{vs} + VD_{vg} \right) \tag{8.25}$$

$$\begin{aligned} \mathrm{Src}.\theta_F = \frac{\mathcal{L}_f}{C_p \Pi} \, & (NUF_{ci} + NUC_{ci} + NUH_{ci} + CL_{cs} + CL_{cg} + CL_{ri} + CL_{rs} + CL_{rg} \\ & - ML_{ic} - ML_{sr} - ML_{gr} + FR_{rg} - SH_{sr} - SH_{gr}) \end{aligned} \tag{8.26}$$

Source term of the equation for mixing ratio of water vapor q_v (8.15):

$$\mathrm{Src}.q_v = -NUA_{vi} - VD_{vr} - VD_{vi} - VD_{vs} - VD_{vg} \tag{8.27}$$

Source term of the equation for mixing ratio of cloud water q_c (8.16):

$$\mathrm{Src}.q_c = -NUF_{ci} - NUC_{ci} - NUH_{ci} - CL_{cr} - CL_{cs} - CL_{cg} - CN_{cr} + ML_{ic} \tag{8.28}$$

Source term of the equation for mixing ratio of rain water q_r (8.17):

$$\begin{aligned} \mathrm{Src}.q_r = & \, VD_{vr} + CL_{cr} - CL_{ri} - CL_{rs} - CL_{rg} + CN_{cr} \\ & + ML_{sr} + ML_{gr} - FR_{rg} + SH_{sr} + SH_{gr} \end{aligned} \tag{8.29}$$

Source term of the equation for mixing ratio of cloud ice q_i (8.18):

$$\begin{aligned} \mathrm{Src}.q_i = & \, NUA_{vi} + NUF_{ci} + NUC_{ci} + NUH_{ci} \\ & + VD_{vi} - CL_{ir} - CL_{is} - CL_{ig} - CN_{is} - ML_{ic} + SP_{si} + SP_{gi} \end{aligned} \tag{8.30}$$

Source term of the equation for mixing ratio of snow q_s (8.19):

$$\begin{aligned} \mathrm{Src}.q_s = & -SP_{si} + VD_{vs} + CL_{cs} + CL_{rs}\alpha_{rs} + CL_{is} - CL_{sr}\left(1 - \alpha_{rs}\right) - CL_{sg} \\ & + CN_{is} - CN_{sg} - ML_{sr} - SH_{sr} \end{aligned} \tag{8.31}$$

Source term of the equation for mixing ratio of graupel q_g (8.20):

$$\begin{aligned} \mathrm{Src}.q_g = & -SP_{gi} + VD_{vg} + PG_g + CL_{ri} + CL_{ir} + \left(CL_{rs} + CL_{sr}\right)\left(1 - \alpha_{rs}\right) \\ & + CN_{sg} - ML_{gr} + FR_{rg} - SH_{gr} \end{aligned} \tag{8.32}$$

Source term of the equation for number concentration of cloud ice $\frac{N_i}{\bar{\rho}}$ (8.21):

$$\mathrm{Src.}\frac{N_i}{\bar{\rho}} = \frac{1}{m_{i0}} NUA_{vi} + \frac{N_c}{\bar{\rho} q_c} \left(NUF_{ci} + NUC_{ci} + NUH_{ci} \right) + SP_{si}^N + SP_{gi}^N + \frac{N_i}{\bar{\rho} q_i} \left(VD_{vi} - CL_{ir} - CL_{is} - CL_{ig} - ML_{ic} \right) - AG_i^N - \frac{1}{m_{s0}} CN_{is} \quad (8.33)$$

Source term of the equation for number concentration of snow $\frac{N_s}{\bar{\rho}}$ (8.22):

$$\mathrm{Src.}\frac{N_s}{\bar{\rho}} = \frac{N_s}{\bar{\rho} q_s} \left(VD_{vs} - ML_{sr} \right) - CL_{sr}^N \left(1 - \alpha_{rs} \right) - CL_{sg}^N - AG_s^N + \frac{1}{m_{s0}} CN_{is} - CN_{sg}^N \quad (8.34)$$

Source term of the equation for number concentration of graupel $\frac{N_g}{\bar{\rho}}$ (8.23):

$$\mathrm{Src.}\frac{N_g}{\bar{\rho}} = \frac{N_g}{\bar{\rho} q_g} \left(VD_{vg} - ML_{gr} \right) + CL_{ri}^N + CL_{rs}^N \left(1 - \alpha_{rs} \right) + CN_{sg}^N + FR_{rg}^N \quad (8.35)$$

were $\mathcal{L}_v, \mathcal{L}_s$, and $\mathcal{L}_f$ are latent heat with evaporation, sublimation, and melting [J kg^{-1}], respectively. C_p is specific heat at the constant pressure [J K kg^{-1}] in dry air. Π is the Exner function. m_{i0} and m_{s0} are the minimum masses of cloud ice and snow [kg], respectively. The next table shows the meaning of each term in the above equations. Figure 8.1 shows the correlation between categories. In Section 8.2.4 conversion terms, which consist of these source terms, are formulated.

Sign	Contents
NUA_{vi}	Deposition or sorption nucleation
NUF_{ci}	Condensation-freezing nucleation
NUC_{ci}	Contact-freezing nucleation
NUH_{ci}	Homogeneous-freezing nucleation
SP	Secondary nucleation of ice crystals
VD	Vapor deposition, evaporation and sublimation
CL	Collision collection
PG	Growth of graupel with collisional collection （graupel produciton）
AG	Aggregation
CN	Conversion from a certain category to the other categories （conversion）
ML	Melting
FR	Freezing
SH	Shedding of liquid water
SP^N	Secondary nucleation of ice crystals for number concentration
CL^N	Collisional collection for number concentration （collection）
AG^N	Aggregation for number concentration （aggregation）

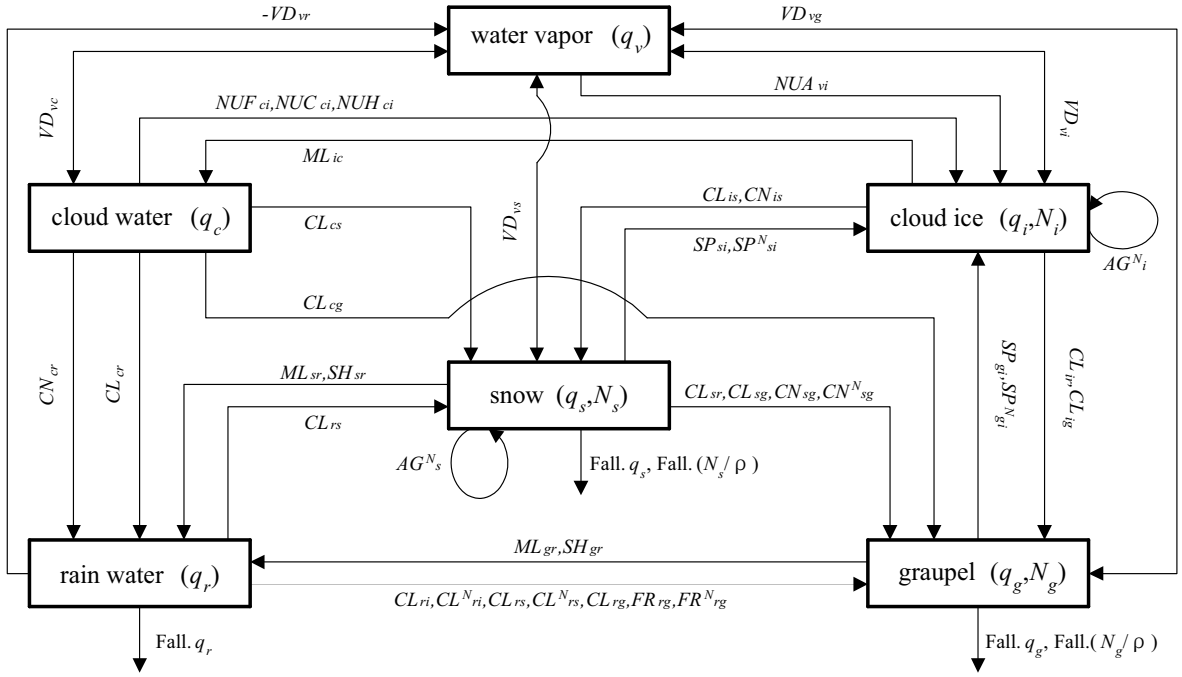

Figure 8.1. Correlation of cloud microphysical processes in a bulk method

CN^N	Conversion from a certain category to the other categories for number concentration (conversion)
FR^N	Freezing for number concentration (freezing)
α_{rs}	$1-\alpha_{rs}$: Production rate of graupel with the collision of raindrops and snow

There is no exchange term VD_{vc} between water vapor and an cloud water in (8.24), (8.27), and (8.28). The exchange term is calculated with the moist saturation adjustment method. Section 8.2.5 describes about the calculation.

8.2.3 Expression of Particles of Cloud and Precipitation

Particle-size Distribution

When we use a bulk method, we need to give particle-size distribution with a certain suitable function because only mixing ratio, or mixing ratio and number concentration can be directly calculated. A particle-size distribution is needed for the calculations of averaged mass and averaged fall velocity.

The particle-size distribution in the bulk method is often given with the exponential function. This is given by

$$\underbrace{n_x(D_x)}_{[m^{-4}]} = \underbrace{n_{x0}}_{[m^{-4}]} \exp(\underbrace{-\lambda_x}_{[m^{-1}]} D_x) \tag{8.36}$$

using the parameters of an inclination in an exponential function for particle size distribution λ_x and y-section density n_{x0}. The parameters are based on Marshall and Palmer (1948). This distribution is sometimes called the Marshall-Palmer distribution. The distribution is sometimes expressed by using a gamma function[2]
A gamma function distribution[3] expressing particle size distribution of a category is

$$\underbrace{f_x\left(D_x\right)}_{[m^{-1}]} = \frac{1}{\Gamma(\nu_x)}\left(\frac{D_x}{D_{nx}}\right)^{\nu_x-1}\frac{1}{D_{nx}}\exp\left(-\frac{D_x}{D_{nx}}\right) \tag{8.43}$$

where D_x is the diameter of particle [m] and $\Gamma\left(\nu_x\right)$ is a standardization (Integral from 0 to ∞ is made to be set to 1.) constant, ν_x is the form parameter of a gamma function and D_{nx} is a characteristic diameter. Number concentrations of particles of cloud and precipitation are expressed as

[2]A gamma function is one of the special functions which extended Factorial $n!$ even to the complex number. Using this function, we can mechanically solve a certain kind of definite integral. Especially, we often integrated using the exponential function distribution (8.36) in cloud physics. In such a case, definite integral can be solved mechanically. Although there are some definitions for the gamma function $\Gamma\left(x\right)$, it is defined with integration as follows.

$$\Gamma\left(x\right) = \int_0^\infty \exp\left(-t\right)\ t^{x-1}\ dt \tag{8.37}$$

$$\Gamma\left(x+1\right) = x\Gamma\left(x\right) \tag{8.38}$$

$$\Gamma\left(1\right) = 1 \tag{8.39}$$

where especially x is the positive integer NNN, it is set to

$$\Gamma\left(n+1\right) = n\left(n-1\right)\left(n-2\right)\cdots 2\cdot 1\cdot\Gamma\left(1\right) = n! \tag{8.40}$$

and is

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}. \tag{8.41}$$

For example, if this is used instead of carrying out integration by parts 3 times, in the definite integral of cloud physics, the following is mechanically calculable like

$$\int_0^\infty D_x^3\exp\left(-\lambda_x D_x\right)\ dD_x = \frac{1}{\lambda_x^4}\Gamma\left(4\right) = \frac{6}{\lambda_x^4} \tag{8.42}$$

[3]Murakami (1999)

$$\underbrace{n_x\left(D_x\right)}_{[m^{-4}]} = \underbrace{n_{xt}}_{[m^{-3}]}\underbrace{f_x\left(D_x\right)}_{[m^{-1}]}. \tag{8.44}$$

The averaged diameter of a particle $\bar{D}_x$ in this distribution is

$$\bar{D}_x = \int_0^\infty D_x\ f_x\left(D_x\right)\ dD_x = \frac{\Gamma\left(\nu_x+1\right)}{\Gamma\left(\nu_x\right)} D_{nx} = \nu_x D_{nx}. \tag{8.45}$$

Here, the relation of (8.38) is used for the last conversion. Generally, the P'th moment of a gamma function distribution is expressed as follows (a complex number is sufficient as P).

$$\int_0^\infty D_x^P\ f_x\left(D_x\right)\ dD_x = \frac{\Gamma\left(\nu_x+P\right)}{\Gamma\left(\nu_x\right)} D_{nx}^P. \tag{8.46}$$

The exponential function distribution (8.36) is one of the special form of a gamma function distribution (8.43). The distribution (8.43) is the exponential function distribution (8.36) when we set as follows.

$$\nu_x = 1 \tag{8.47}$$

$$D_{nx} = \frac{1}{\lambda_x} \tag{8.48}$$

In this case, (8.46) can be expressed as

$$\int_0^\infty D_x^P\ \lambda_x \exp\left(-\lambda_x D_x\right)\ dD_x = \frac{1}{\lambda_x^P}\Gamma\left(P+1\right). \tag{8.49}$$

It is an availability of a gamma function that definite integral is made simply. The average radius of a particle $\bar{D}_x$ is

$$\bar{D}_x = \frac{1}{\lambda_x}. \tag{8.50}$$

A single distribution is used for an cloud water and cloud ice, and the exponential function distribution[4] is used for precipitation particles of rain water, snow, and graupel. Here, the averaged diameters of cloud water and cloud ice are

[4]The exponential function distribution is used by Murakami (1999), Ikawa and Saito (1991), Murakami et al. (1994), Ikawa et al. (1991), Lin et al. (1983). Ferrier (1994) uses the exponential function distribution for cloud water and the gamma function distribution for the other precipitation particles.

$$\bar{D}_c = \left(\frac{6\bar{\rho} q_c}{\pi \rho_w N_c} \right)^{\frac{1}{3}} \tag{8.51}$$

$$\bar{D}_i = \left(\frac{6\bar{\rho} q_i}{\pi \rho_i N_i} \right)^{\frac{1}{3}} . \tag{8.52}$$

where ρ_w is density of cloud water [kg m^{-3}] and ρ_i is density of cloud-ice [kg m^{-3}]. The constant value 1×10^8 m^{-3} is used for the number concentration of cloud water N_c.

Particle size distributions of rain water, snow, and graupel are given as

$$n_r (D_r) = n_{r0} \exp (-\lambda_r D_r) \tag{8.53}$$
$$n_s (D_s) = n_{s0} \exp (-\lambda_s D_s) \tag{8.54}$$
$$n_g (D_g) = n_{g0} \exp (-\lambda_g D_g) . \tag{8.55}$$

Note that the unit of n_x is [m^{-4}].

Averaged Mass and Averaged Fall Velocity

The mass of a particle of cloud or precipitation, which depends on the form of the particle, is given by the empirical equation of an exponentiation of a particle diameter.

$$m_x (D_x) = \alpha_{ux} D_x^{\beta_{ux}} \tag{8.56}$$

$\beta_{ux} = 3$ in the case of a spherical particle. Using a gamma function distribution, the averaged mass is

$$\bar{m}_x = \alpha_{ux} D_{nx}^{\beta_{ux}} \frac{\Gamma (\nu_x + \beta_{ux})}{\Gamma(\nu_x)} . \tag{8.57}$$

The fall velocity of a particle is also given by the empirical equation of the exponentiation of a diameter.

$$U_x (D_x) = \alpha_{ux} D_x^{\beta_{ux}} \left(\frac{\rho_0}{\bar{\rho}} \right)^{\gamma_{ux}} \tag{8.58}$$

In the same way, using a gamma function distribution, the averaged fall velocity of which the weight with the particle number concentration per unit diameter [m^{-4}] is multiplied is

$$\bar{U}_{xN} = \alpha_{ux} D_{nx}^{\beta_{ux}} \frac{\Gamma\left(\nu_x + \beta_{ux}\right)}{\Gamma(\nu_x)} \left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}}. \tag{8.59}$$

The averaged fall velocity of which the weight of mass is multiplied is

$$\bar{U}_{xq} = \alpha_{ux} D_{nx}^{\beta_{ux}} \frac{\Gamma\left(\nu_x + 3 + \beta_{ux}\right)}{\Gamma\left(\nu_x + 3\right)} \left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}}. \tag{8.60}$$

These relations include an exponential function distribution as a special case. Using the relational expression with a gamma function in the cases of (8.47), (8.60) and (8.59) are

$$\bar{U}_{xN} = \alpha_{ux} \frac{\Gamma\left(1 + \beta_{ux}\right)}{\lambda_x^{\beta_{ux}}} \left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}} \tag{8.61}$$

$$\bar{U}_{xq} = \alpha_{ux} \frac{\Gamma\left(4 + \beta_{ux}\right)}{6\lambda_x^{\beta_{ux}}} \left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}}. \tag{8.62}$$

where ρ_0 is the air density at the surface in the basic state [kg m^{-3}].

In the exponential function distribution[5] (8.53)$\sim$(8.55), the total particle number for categories $x = r, s, g$: N_x is

$$N_x = \int_0^\infty n_{x0} \exp\left(-\lambda_x D_x\right)\ dD_x = \frac{n_{x0}}{\lambda_x}. \tag{8.63}$$

The total mass of water substance x per unit volume is

$$\bar{\rho} q_x = \int_0^\infty \frac{\pi}{6} \rho_x D_x^3 n_{x0} \exp\left(-\lambda_x D_x\right)\ dD_x = \frac{\pi \rho_x n_{x0}}{\lambda_x^4}. \tag{8.64}$$

From these two equations, an inclination parameter in an exponential function paritlce size distribution λ_x andthe y-section density n_{x0} are

$$\lambda_x = \left(\frac{\pi \rho_x N_x}{\bar{\rho} q_x}\right)^{\frac{1}{3}} \tag{8.65}$$

$$n_{x0} = N_x \left(\frac{\pi \rho_x N_x}{\bar{\rho} q_x}\right)^{\frac{1}{3}}. \tag{8.66}$$

[5] Ikawa and Saito (1991)

Then, the terminal fall velocity of particle in category x which is weighted by number concentration is

$$\begin{aligned}\bar{U}_{xN} &= \frac{1}{N_x}\int_0^\infty U_x\left(D_x\right) n_{x0}\exp\left(-\lambda_x D_x\right)\, dD_x \\ &= \alpha_{ux}\frac{\Gamma\left(1+\beta_{ux}\right)}{\lambda_x^{\beta_{ux}}}\left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}}\end{aligned} \tag{8.67}$$

and this is consistent with (8.61). In the same way, the mass-weighted terminal fall speed of category x is

$$\begin{aligned}\bar{U}_{xq} &= \frac{1}{\bar{\rho}q_x}\int_0^\infty \frac{\pi}{6}U_x\left(D_x\right) D_x^3\rho_x n_{x0}\exp\left(-\lambda_x D_x\right)\, dD_x \\ &= \alpha_{ux}\frac{\Gamma\left(4+\beta_{ux}\right)}{6\lambda_x^{\beta_{ux}}}\left(\frac{\rho_0}{\bar{\rho}}\right)^{\gamma_{ux}}.\end{aligned} \tag{8.68}$$

and this is consistent with (8.62). These averaged fall velocities are used for calculations of change of mixing ratio and number concentration in association with a falling, as described in Section 8.2.6.

Finally, the form parameter of each category is summarized below.

variable	the density of y-section[m^{-4}]	the form parameter of fall speed	density[kg m^{-3}]
q_c	———	$(\alpha_{uc}=2.98\times10^7,\ \beta_{uc}=2.0,\ \gamma_{uc}=1.0)$	$\rho_w=1.0\times10^3$
q_r	$n_{r0}=8.0\times10^6$	$\alpha_{ur}=842,\ \beta_{ur}=0.8,\ \gamma_{ur}=0.5$	$\rho_w=1.0\times10^3$
q_i	———	$(\alpha_{ui}=700,\ \beta_{ui}=1.0,\ \gamma_{ui}=0.33)$	$\rho_i=5.0\times10^2$
q_s	$(n_{s0}=1.8\times10^6)$	$\alpha_{us}=17,\ \beta_{us}=0.5,\ \gamma_{us}=0.5$	$\rho_s=8.4\times10^1$
q_g	$(n_{g0}=1.1\times10^6)$	$\alpha_{ug}=124,\ \beta_{ug}=0.64,\ \gamma_{ug}=0.5$	$\rho_g=3.0\times10^2$

8.2.4 Formulizations of Physical Processes of Source terms

Hereafter, we will explain physical processes of source terms. Signs are summarized into table in each section (Sections 8.2.5 and 8.2.7 are the same as this section).

Nuclear Formation of Primary Ice ä $NUA_{vi}, NUF_{ci}, NUC_{ci}, NUH_{ci}$

The primary ice nucleation are shown as below.

homogeneous nucleation	sublimation nucleation		water vapor → ice crystals	×:not occur
	freezing nucleation		supercooled water drops → ice crystals	◯:NUH_{ci}
heterogeneous nucleation	sublimation nucleation	sublimation nuclei	water vapor → ice crystals	◯:NUA_{vi}
	freezing nucleation	sorption nuclei		×:unknown
		contact-freezing nuclei	supercooled water drops → ice crystals	◯:NUC_{ci}
		immersion-freezing nuclei	supercooled water drops → ice crystals	◯:NUF_{ci}

Our model adopts $NUA_{vi}, NUF_{ci}, NUC_{ci}, NUH_{ci}$.

(1)Sublimation Nucleation: NUA_{vi}

(a)Number Concentration of Sublimation Nuclei as a Function of Supercooling Temperature[6]

Number concentration of sublimation nuclei as a function of supercooled temperature T_s is written by

$$NUA_{vi} = \frac{m_{i0}}{\bar{\rho}} \beta_2 N_{i0} \exp\left(\beta_2 T_s\right) \left(\frac{S_i - 1}{S_{wi} - 1}\right)^B \frac{\partial T_s}{\partial z} w \tag{8.69}$$

$$NUA_{vi}^N = \frac{NUA_{vi}}{m_{i0}}. \tag{8.70}$$

Note that a difference in z^* coordinate system (ζ coordinate system) is multiplied by a matrix, as shown in equations (6.58) since the vertical difference is a difference in a real space.

(b)Number Concentration of Sublimation Nuclei as a Function of Supersaturation[7]

Number concentration of sublimation nuclei as a function of supersaturation is written by

$$NUA_{vi} = \frac{m_{i0}}{\bar{\rho}} 15.25 \exp\left(5.17 + 15.25 SS_i\right) \frac{\partial SS_i}{\partial z} w \tag{8.71}$$

$$NUA_{vi}^N = \frac{NUA_{vi}}{m_{i0}} \tag{8.72}$$

[6]Ikawa and Saito (1991), Cotton et al. (1986), Murakami (1990), Ikawa et al. (1991), Murakami et al. (1994), 村上 (1999)

[7]Meyers et al. (1992), 村上 (1999)

Note that a difference in z^* coordinate system (ζ cordinate system) is multiplied by a matrix, as well as the same way in (a).

(c)Considering both Supercooling Temperature and Supersaturation[8]

Ferrier (1994) adopted a method that uses two equations owing to temperature for heterogeneous sublimation nucleation. In the method, the equation of Murakami (1990), Cotton et al. (1986) is used when temperature is more than -5 °C , and that of Meyers et al. (1992) is used when temperature is less than -5 °C .

$$NUA_{vi} = \frac{m_{i0}}{\bar{\rho}} w \frac{\partial N_i}{\partial z} \tag{8.73}$$

$$NUA_{vi}^N = \frac{NUA_{vi}}{m_{i0}} \tag{8.74}$$

where N_i is divided by -5 °C as follows.

$$N_i = \begin{cases} N_{i01} \exp\left(\beta_2 T_s\right) \left(\dfrac{S_i - 1}{S_{wi} - 1}\right)^B, & T \geq -5 \ [^\circ\mathrm{C}] \\ N_{i02} \exp\left(a_1 SS_i - b_1\right), & T < -5 \ [^\circ\mathrm{C}] \end{cases} \tag{8.75}$$

Note that a difference in z^* coordinate system (ζ coordinate system) is multiplied by a matrix, as well as the same way in (a) and (b).

Meanings of the signs used in (a)∼(c) are as follows.

a_1	Coefficient of the Ferrier equation in below -5 °C	12.96	
b_1	Coefficient of the Ferrier equation in below -5 °C	0.639	
B	Coefficient of the Huffmann and Vail equation	4.5	
m_{i0}	Minimum mass of cloud ice	10^{-12}	kg
N_{i0}	Coefficient of Fletcher equation	10^{-2}	m^{-3}
N_{i01}	Number of particles of the Ferrier equation in more than -5 °C	10^3	m^{-3}
N_{i02}	Number of particles of the Ferrier equation in below -5 °C	50	m^{-3}
q_{vsi}	Saturation mixing ratio over ice		kg kg^{-1}
q_{vsw}	Saturation mixing ratio over water		kg kg^{-1}
T	Temperature		K
T_0	Melting point of ice	273.16	K
T_s	Supercooling temperature $(T_0 - T)$		K

[8]Ferrier (1994)

S_i	Ratio of mixing ratio of water vapor and saturation mixing ratio of ice in an air parcel		
S_{wi}	Ratio of the saturation mixing ratios of water and ice		
SS_i	Ice supersaturation $(S_i - 1)$		
w	Vertical motion in z coordinate system		m s^{-1}
β_2	Coefficient of the Fletcher equation	0.6	K^{-1}
$\bar{\rho}$	Air density in the basic state		kg m^{-3}

(2)Immesion Freezing Nucleation: NUF_{ci}

Heterogeneous nucleation of cloud droplets depends on a size, physical and chemical features of freezing nucleus and temperature and size of a cloud drop. We use the nucleation that is extrapolated outside to the size of a cloud droplet with the Bigg (1953) empirical equation[9].

$$NUF_{ci} = B' \left[\exp\left(A'T_s\right) - 1\right] \frac{\bar{\rho} q_c^2}{\rho_w N_c} \tag{8.76}$$

$$NUF_{ci}^N = B' \left[\exp\left(A'T_s\right) - 1\right] \frac{q_c}{\rho_w} \tag{8.77}$$

Meanings of the signs used here are as follows.

A'	Coefficient of the Bigg empirical equation	0.66	K^{-1}
B'	Coefficient of the Bigg empirical equation	100.0	m^{-3} s^{-1}
N_c	Number concentration of cloud droplets	1×10^8	m^{-3}
T	Temperature		K
T_0	Melting point of ice	273.16	K
T_s	Supercooled temperature $(T_0 - T)$		K
$\bar{\rho}$	Air density in the basic state		kg m^{-3}
ρ_w	Density of water	1×10^3	kg m^{-3}

(3)Contact Freezing Nucleation ä NUC_{ci}

The contact freezing nucleation is occurred with a collision of an ice particle core (freezing core) and a supercooled cloud droplet. The following three processes are considered to it.

- Brownian diffusion
- diffusiophoresis
- thermophoresis

The generation rate of ice particles by these processes is given by

[9]村上 (1999), Ikawa and Saito (1991)

$$\left[\frac{dN_c}{dt}\right]_b = F_1 \mathcal{D}_{ar} \tag{8.78}$$

$$\left[\frac{dN_c}{dt}\right]_v = F_1 F_2 \frac{R_v T}{\mathcal{L}_v} \tag{8.79}$$

$$\left[\frac{dN_c}{dt}\right]_t = F_1 F_2 f_t \tag{8.80}$$

where

$$F_1 = 2\pi D_c N_c N_{ar} \tag{8.81}$$

$$F_2 = \frac{\kappa_a}{p}\left(T - T_{cl}\right) \tag{8.82}$$

$$f_t = \frac{0.4\left[1 + 1.45K_n + 0.4\exp\left(-1/K_n\right)\right]\left(\kappa + 2.5K_n\kappa_a\right)}{\left(1 + 3K_n\right)\left(2\kappa + 5\kappa_a K_n + \kappa_a\right)}. \tag{8.83}$$

K_n in the equations is the Knudsen number which is defined as

$$K_n = \lambda_{a0}\frac{p_{00}}{T_{00}R_a}\frac{T}{p}. \tag{8.84}$$

The diffusion coefficient of an aerosol particle is

$$\mathcal{D}_{ar} = \frac{kT_{cl}}{6\pi R_a \mu}\left(1 + K_n\right). \tag{8.85}$$

The number concentration N_{ar} of the contact freezing nuclei activated by temperature T_{cl}[10] is

$$N_{ar} = N_{a0}\left(270.16 - T_{cl}\right)^{1.3}. \tag{8.86}$$

The formation of an ice particle with the contact freezing nuclei is finally represented by the sum of the three processes, as shown in follows.

$$NUC_{ci}^N = \frac{1}{\bar{\rho}}\left(\left[\frac{dN_c}{dt}\right]_b + \left[\frac{dN_c}{dt}\right]_v + \left[\frac{dN_c}{dt}\right]_t\right) \tag{8.87}$$

$$NUC_{ci} = \frac{\bar{\rho}q_c}{N_c}NUC_{ci}^N \tag{8.88}$$

[10] Cotton et al. (1986)

Meanings of the signs used here are as follows.

D_c	Diameter of a cloud droplet		m
k	The Boltzmann number	1.380658×10^{-23}	J K^{-1}
$\mathcal{L}_v$	Latent heat of evaporation of water		J kg^{-1}
N_{a0}	Coefficient in the number concentration of contact freezing nuclei	2×10^5	m^{-3}
N_c	Number concentration of cloud droplets	1×10^8	m^{-3}
p	Pressure		Pa
p_{00}	Standard pressure	101325	Pa
R_a	Radius of an aerosol particle	3×10^{-7}	m
R_v	The gas constant of water vapor	461.0	J K^{-1} kg^{-1}
T	Temperature		K
T_{00}	Temperature in the basic state	293.15	K
T_{cl}	Temperature of a cloud droplet		K
κ	Thermal conductivity of air	2.4×10^{-2}	J m^{-1} s^{-1} K^{-1}
κ_a	Thermal conductivity of an aerosol particle		J m^{-1} s^{-1} K^{-1}
λ_{a0}	Mean free path at p_{00}, T_{00}	6.6×10^{-8}	m
μ	Coefficient of viscosity of air		kg m^{-1} s^{-1}
$\bar{\rho}$	Air density in the basic state		kg m^{-3}

(4)Homogeneous Freezing Nuclei ä NUH_{ci}

We assume that cloud water freezes momentarily when temperature is below -40 °C [11]. Rate of the homogeneous freezing nuclei is written by

$$NUH_{ci}^N = \frac{1}{\bar{\rho}}\frac{N_c}{2\Delta t} \tag{8.89}$$

$$NUH_{ci} = \frac{q_c}{2\Delta t} \tag{8.90}$$

where Δt is the time step in the leap frog method. Note that generation of cloud water must be given by the moist saturation adjustment before this calculation (Ferrier, 1994).

Secondary Nucleation of Ice Crystals: SP

Although there are still many points which are not known about the nucleation of secondary ice crystals, the following processes are mainly known.

[11] Ikawa and Saito (1991), Ferrier (1994)

- While snow and graupel grow up with collection of cloud droplets, minute particles of ice are generated (Hallett and Mossop, 1974).
- Snow and graupel collide each other while they are falling, and the fragment of small ice is generated (Vardiman, 1978).
- When the large supercooled water droplet, which is generated near the cloud top, freezes, ice crystals of high-concentration are generated (Hobbs and Rangno, 1985).

We consider only the Hallett-Mossop rime splintering mechanism[12]. Generation rates of the secondary nucleation of ice crystals are

$$SP_{si}^N = \frac{1}{\bar{\rho}} \times 3.5 \times 10^8 f\left(T_s\right) CL_{cs} \tag{8.91}$$

$$SP_{si} = m_{i0} SP_{si}^N \tag{8.92}$$

$$SP_{gi}^N = \frac{1}{\bar{\rho}} \times 3.5 \times 10^8 f\left(T_g\right) CL_{cg} \tag{8.93}$$

$$SP_{gi} = m_{i0} SP_{gi}^N. \tag{8.94}$$

Note that the secondary nucleation of ice crystals has never occurred in the wet growth of graupel. Function of the temperature of a particle $f\left(T_x\right)$ (T_x is T_s or T_g) is defined as follows.

$$f\left(T_x\right) = \begin{cases} 0, & T_x > 270.16\ [K] \\ \dfrac{T_x - 268.16}{2}, & 268.16 \leq T_x \leq 270.16\ [K] \\ \dfrac{268.16 - T_x}{3}, & 265.16 \leq T_x \leq 268.16\ [K] \\ 0, & T_x < 265.16\ [K] \end{cases} \tag{8.95}$$

Although $f\left(T_x\right)$ will become a negative value in (72) of Cotton et al. (1986), if an absolute value is taken, it will become the same as the equation (8.95). Although (8.95) is a function that it may be $f\left(T_x\right) = 0$ at $T_x = 268.16$ [K] , Ikawa et al. (1991), Ikawa and Saito (1991) use a function that it may be $f\left(T_x\right) = 1$ at $T_x = 268.16$[K] as follows.

$$f\left(T_x\right) = \begin{cases} 0, & T_x \geq 270.16\ [K] \\ \dfrac{270.16 - T_x}{2}, & 268.16 < T_x < 270.16\ [K] \\ 1, & T_x = 268.16\ [K] \\ \dfrac{T_x - 265.16}{3}, & 265.16 \leq T_x < 268.16\ [K] \\ 0, & T_x < 265.16\ [K] \end{cases} \tag{8.96}$$

[12]村上 (1999), Ikawa and Saito (1991), Cotton et al. (1986)

This function has the maximum value at -5 °C and is considered to be more reasonable.

Meanings of the signs used here are as follows.

CL_{cg}	Growth rate of graupel in the collision with cloud droplets		s^{-1}
CL_{cs}	Growth rate of snow in the collision with cloud droplets		s^{-1}
m_{i0}	The mass of the minimum cloud ice	10^{-12}	kg
T_s	Temperature of snow		K
T_g	Temperature of graupel		K
T_x	Temperature of snow or graupel		K
$\bar{\rho}$	Air density in the basic state		kg m^{-3}

Water Vapor Diffusion Growth ä VD

Diffusion growth is the generation, the growth, disappearance, and consumption by exchange of the water molecule between moisture and a particle. There are the following processes and some of them are considered in our model.

gaseous phase — liquid phase	condensation	water vapor → cloud water	calculated by the moist saturation adjustment
		water vapor → rain water	neglected because of small amount
	evaporation	cloud water → water vapor	calculated by the moist saturation adjustment
		rain water → water vapor	$VD_{vr} < 0$
gaseous phase — solid phase	sublimation	water vapor → cloud ice	$VD_{vi} > 0$
		water vapor → snow	$VD_{vs} > 0$
		water vapor → graupel	$VD_{vg} > 0$
		water vapor → hail	$VD_{vh} > 0$
	sublimation evaporation	cloud ice → water vapor	$VD_{vi} < 0$
		snow → water vapor	$VD_{vs} < 0$
		graupel → water vapor	$VD_{vg} < 0$
		hail → water vapor	$VD_{vh} < 0$

(1)Evaporation of Rain Water ä VD_{rv}

Variations of mixing ratio and number concentration of rain water [13] are represented by

$$VD_{vr} = -VD_{rv} = \frac{2\pi}{\bar{\rho}} \left(S_w - 1\right) G_w \left(T, p\right) VENT_r \tag{8.97}$$

where

$$G_w \left(T, p\right) = \left(\frac{\mathcal{L}_v^2}{\kappa R_v T^2} + \frac{1}{\bar{\rho} q_{vsw} \mathcal{D}_v} \right)^{-1}. \tag{8.98}$$

The term of the ventilation coefficient of rain water is represented by

$$VENT_r = n_{r0} \left[0.78 \lambda_r^{-2} + 0.31 S_c^{\frac{1}{3}} \nu^{-\frac{1}{2}} \alpha_{\bar{u}r}^{\frac{1}{2}} \Gamma \left(\frac{5 + \beta_{ur}}{2} \right) \lambda_r^{-\frac{(5+\beta_{ur})}{2}} \left(\frac{\rho_0}{\bar{\rho}} \right)^{\frac{1}{4}} \right]. \tag{8.99}$$

(2)Sublimation Condensation of Snow and Graupel (Sublimation Evaporation)ä VD_{vs}, VD_{vg} [14]

Sublimation condensation (sublimation evaporation) rates of snow and graupel are formulized like rain water. However, they are divided in the cases more and less than the melting point of ice. Here, snow and graupel are simultaneously formulized as $x = s, g$. That is,

considering the latent heat of freezing in a collecting supercooled cloud droplets at $T < T_0$,

$$VD_{vx} = \frac{2\pi}{\bar{\rho}} \left(S_i - 1\right) G_i \left(T, p\right) VENT_x - \frac{\mathcal{L}_s \mathcal{L}_f}{\kappa R_v T^2} G_i \left(T, p\right) CL_{cx}. \tag{8.100}$$

At $T > T_0$,

when a melting process occurs,

$$VD_{vx} = 2\pi \mathcal{D}_v \left(q_v - q_{vs} \left(T_0\right)\right) VENT_x \tag{8.101}$$

when a melting process does not occur,

$$VD_{vx} = \frac{2\pi}{\bar{\rho}} \left(S_w - 1\right) G_w \left(T, p\right) VENT_x. \tag{8.102}$$

[13]村上 (1999)
[14]村上 (1999), Ikawa and Saito (1991), Lin et al (1983)

where

$$G_i(T,p) = \left(\frac{\mathcal{L}_s^2}{\kappa R_v T^2} + \frac{1}{\bar{\rho} q_{vsi} \mathcal{D}_v} \right)^{-1} \tag{8.103}$$

where $G_w(T,p)$ is given by the equation (8.98). A ventilation coefficient at $x = s, g$ is as follows.

$$VENT_x = n_{x0} \left[0.78 \lambda_x^{-2} + 0.31 S_c^{\frac{1}{3}} \nu^{-\frac{1}{2}} \alpha_{ux}^{\frac{1}{2}} \Gamma \left(\frac{5+\beta_{ux}}{2} \right) \lambda_x^{-\frac{(5+\beta_{ux})}{2}} \left(\frac{\rho_0}{\bar{\rho}} \right)^{\frac{1}{4}} \right] \tag{8.104}$$

(3)Growth of Cloud Ice in Gaseous Phase ä VD_{vi} [15]

Variation of growth of cloud ice in gaseous phase is represented as follows.

$$VD_{vi} = \frac{q_v - q_{vsi}}{q_{vsw} - q_{vsi}} a_1 (\bar{m}_i)^{a_2} \frac{N_i}{\bar{\rho}} \tag{8.105}$$

where a_1, a_2 is a parameter depending on temperature (Köenig (1971)) and $\bar{m}_i$ is mean mass of cloud ice given as following equation.

$$\bar{m}_i = \frac{q_i \bar{\rho}}{N_i} \tag{8.106}$$

Meanings of the signs used in (1)-(3) are as follows.

CL_{cg}	Growth rate of graupel in coalescence with cloud droplets		s^{-1}
CL_{cs}	Growth rate of snow in coalescence with cloud droplets		s^{-1}
$\mathcal{D}_v$	Diffusion coefficient of water vapor		$m^2\ s^{-1}$
$\mathcal{L}_f$	Latent heat of melting of water		J kg^{-1}
$\mathcal{L}_s$	Latent heat of sublimation of water		J kg^{-1}
$\mathcal{L}_v$	Latent heat of evaporation of water		J kg^{-1}
n_{g0}	Density of graupel in the y section		m^{-4}
n_{r0}	Density of rain water in the y section	8.0×10^6	m^{-4}
n_{s0}	Density of snow in the y section		m^{-4}
$q_{vs}(T_0)$	Saturation mixing ratio of the melting point of water		kg kg^{-1}

[15] Ikawa and Saito (1991), Ikawa et al. (1991)

q_{vsi}	Ice saturation mixing ratio		kg kg^{-1}
q_{vsw}	Water saturation mixing ratio		kg kg^{-1}
R_v	The gas constant of water vapor	461.0	J K^{-1} kg^{-1}
S_c	Schmidt number	0.6	
$S_i - 1$	Ice supersaturation of an air parcel		
$S_w - 1$	Water supersaturation of an air parcel		
T	Temperature		K
T_0	Melting point of ice	273.16	K
α_{ug}	Coefficient in the relational expression of fall velocity and diameter of graupel	124	m$^{1-\beta_{ug}}$ s^{-1}
α_{ur}	Coefficient in the relational expression of fall velocity and diameter of rain water	842	m$^{1-\beta_{ur}}$ s^{-1}
α_{us}	Coefficient in the relational expression of fall velocity and diameter of snow	17	m$^{1-\beta_{us}}$ s^{-1}
β_{ug}	Coefficient in the relational expression of fall velocity and diameter of graupel	0.64	
β_{ur}	Coefficient in the relational expression of fall velocity and diameter of rain water	0.8	
β_{us}	Coefficient in the relational expression of fall velocity and diameter of snow	0.5	
λ_g	Inclination in a reversed exponential function expressing particle size distribution of graupel		m^{-1}
λ_r	Inclination in a reversed exponential function expressing particle size distribution of rain water		m^{-1}
λ_s	Inclination in a reversed exponential function expressing particle size distribution of snow		m^{-1}
κ	Thermal conductivity of air	2.4×10^{-2}	J m^{-1} s^{-1} K^{-1}
ν	Coefficient of kinematic viscosity of air		m^2 s^{-1}
$\bar{\rho}$	Air density in the basic state		kg m^{-3}
ρ_0	Density at the surface in the basic state		kg m^{-3}

Particle Collection: CL

In this section, we formulize the growth process of the particle which captures a certain particle by the collision between various particles. For the particle collision, the following processes can be respectively considered to the categories of the particles which are defined before.

process	sink	source	sign	meaning	model
Collection of cloud water	q_c	q_i	CL_{ci}	Growth of cloud ice by collection with cloud water	×
	q_c	q_s	CL_{cs}	Growth of snow by collection with cloud water	◯
	q_c	q_g	CL_{cg}	Growth of graupel by collection with cloud water	◯
	q_c	q_h	CL_{ch}	Growth of hail by collection with cloud water	△

	q_c	q_r	CL_{cr}	Growth of rain water by collection with cloud water	◯
Collection of rain water	q_r	q_i	CL_{ri}	Growth of cloud ice by collection with rain water	◯
	q_r	q_s	CL_{rs}	Growth of snow by collection with rain water	◯
	q_r	q_g	CL_{rg}	Growth of graupel by collection with rain water	◯
	q_r	q_h	CL_{rh}	Growth of hail by collection with rain water	△
Collection of cloud ice	q_i	q_r	CL_{ir}	Growth of rain water by collection with cloud ice	◯
	q_i	q_s	CL_{is}	Growth of snow by collection with cloud ice	◯
	q_i	q_g	CL_{ig}	Growth of graupel by collection with cloud ice	◯
	q_i	q_h	CL_{ih}	Growth of hail by collection with cloud ice	△
Collection of snow	q_s	q_r	CL_{sr}	Growth of rain water by collection with snow	◯
	q_s	q_g	CL_{sg}	Growth of graupel by collection with snow	◯
	q_s	q_h	CL_{sh}	Growth of hail by collection with snow	△

In this table, "◯" and "×" denote the signs of which it is included or is not included in our model, respectively, and "△" is the sign of suspending for including in our model at present.

The number concentration will change for the collection of snow particles (aggregation) and that of rain water (accretion) but the mixing ratio do not change. These are described in the other section.

In the following section, we formulize the coalescences between large particles (rain water, snow, and graupel), which have a sufficient value of fall velocity, a large particle and a small particle (cloud water and cloud ice) of which its fall velocity can be neglected, and the growth of graupel by collection of supercooled rain water and ice crystals[16].

(1)Coalescence between Rain Water, Snow and Graupel ä CL_{xy} $(x, y = r, s, g; x \neq y)$

Variations of mixing ratio and number concentration by the collection between precipitation particles (rain water, snow, and hail) are formulized as follows.

$$CL_{xy} = \pi^2 \frac{\rho_x}{\bar{\rho}} E_{xy} \sqrt{\left(\bar{U}_x - \bar{U}_y\right)^2 + \alpha \bar{U}_x \bar{U}_y} n_{x0} n_{y0} \left(\frac{5}{\lambda_x^6 \lambda_y} + \frac{2}{\lambda_x^5 \lambda_y^2} + \frac{0.5}{\lambda_x^4 \lambda_y^3} \right) \tag{8.107}$$

$$CL_{xy}^N = \frac{\pi}{2\bar{\rho}} E_{xy} \sqrt{\left(\bar{U}_x - \bar{U}_y\right)^2 + \alpha \bar{U}_x \bar{U}_y} n_{x0} n_{y0} \left(\frac{1}{\lambda_x^3 \lambda_y} + \frac{1}{\lambda_x^2 \lambda_y^2} + \frac{1}{\lambda_x \lambda_y^3} \right) \tag{8.108}$$

where $x, y = r, s, g; x \neq y$. Meanings of the signs used are as follows.

[16]Murakami (1999), Lin et al. (1983), Murakami (1990), Ikawa and Saito (1991)

E_{xy}	Collection efficiency of a particle		
n_{x0}	Density of a category x in the y section		m^{-4}
n_{y0}	Density of a category y in the y section		m^{-4}
$\bar{U}_x$	Mean fall velocity weighted by a mass of a category x		$\mathrm{m\ s}^{-1}$
$\bar{U}_y$	Mean fall velocity weighted by mass of a category y		$\mathrm{m\ s}^{-1}$
α	Coefficient of an adjustment term	0.04	
λ_x	Inclination in a reversed exponential function expressing particle size distribution of a category x		m^{-1}
λ_y	Inclination in a reversed exponential function expressing particle size distribution of a category y		m^{-1}
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_x	Density of a particle of a category x		$\mathrm{kg\ m}^{-3}$

(2)Collection of Rain water, Snow, and Graupel with Cloud Water and Cloud Ice ä CL_{cy}, CL_{iy} $(y = r, s, g)$

Because fall velocities of cloud water and cloud ice are relatively small to those of rain water, snow and graupel, the collection is represented as follows.

$$CL_{xy} = \frac{\pi}{4} \bar{E}_{xy}\, n_{y0}\, q_x\, \alpha_{uy}\, \Gamma\left(3 + \beta_{uy}\right)\, \lambda_y^{-(3+\beta_{uy})}\, \left(\frac{\rho_0}{\bar{\rho}}\right)^{\frac{1}{2}} \tag{8.109}$$

The mean collection efficiency for coalescence with cloud ice $\bar{E}_{iy}$ is constant, and that of coalescence with cloud water $\bar{E}_{cy}$ is given by

$$\bar{E}_{cy} = \frac{Stk^2}{\left(Stk + 0.5\right)^2} \tag{8.110}$$

where Stk is the Stokes number which is given with the average radius of cloud water, cloud ice, and precipitation particle. Ikawa and Saito (1991) calculates as follows.

$$Stk = \bar{D}_c^2 \rho_w \frac{\bar{U}_y}{9\mu \bar{D}_y} \tag{8.111}$$

Meanings of the signs used here are as follows.

D_c	Diameter of a particle of cloud water		m
D_y	Diameter of a particle of a category y		m
n_{y0}	Density of a category y in the y section		m^{-4}
$\bar{U}_y$	Mean fall velocity multiplied by an weight coefficient of mass of a category y		$\mathrm{m\ s}^{-1}$
α_{uy}	Coefficient in the relational expression of fall velocity and diameter of a category y		$\mathrm{m}^{1-\beta_{uy}}\ s^{-1}$
β_{uy}	Coefficient in the relational expression of fall velocity and diameter of a category y		
λ_y	Inclination in a reversed exponential function expressing particle size distribution of a category y		m^{-1}
μ	Coefficient of viscosity of air		$\mathrm{kg\ m}^{-1}\ \mathrm{s}^{-1}$
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_0	Density at the surface in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_w	Density of water	1×10^3	$\mathrm{kg\ m}^{-3}$

(3)Formation Process of Graupel by Collection of a Supercooled Raindrop and Ice

Crystals ä CL_{ri}

When it is assumed that a raindrop freezes instantaneously, variations of mixing ratio and number concentration of graupel with its formation by the collision of a supercooled raindrop and ice crystals are given as follows[17].

$$CL_{ri} = \frac{\pi^2}{24} E_{ir}\ N_i\ n_{r0}\ \alpha_{ur}\ \Gamma\left(6+\beta_{ur}\right)\ \lambda_r^{-(6+\beta_{ur})}\ \left(\frac{\rho_0}{\bar{\rho}}\right)^{\frac{1}{2}} \tag{8.112}$$

$$CL_{ri}^{N} = \frac{\pi}{4\bar{\rho}} E_{ir}\ N_i\ n_{r0}\ \alpha_{ur}\ \Gamma\left(3+\beta_{ur}\right)\ \lambda_r^{-(3+\beta_{ur})}\ \left(\frac{\rho_0}{\bar{\rho}}\right)^{\frac{1}{2}} \tag{8.113}$$

Meanings of the signs used here are as follows.

E_{ir}	Collection efficiency of a particle	1.0	
n_{r0}	Density of rain water in the y section	8.0×10^6	m^{-4}
α_{ur}	Coefficient in the relational expression of fall velocity and diameter of rain water	842	$\mathrm{m}^{1-\beta_{ur}}\ \mathrm{s}^{-1}$
β_{ur}	Coefficient in the relational expression of fall velocity and diameter of rain water	0.8	
λ_r	Inclination in a reversed exponential function expressing particle size distribution of rain water		m^{-1}
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_0	Density at the surface in the basic state		$\mathrm{kg\ m}^{-3}$

(4)Summary of Collection Efficiencies for Coalescences ä E_{xy} [18]

Collection efficiencies E_{xy} for coalescences between particles shown in (1)-(3) are summarized in the following table.

E_{cr}	Collection efficiency for the coalescence of rain water with cloud water	$Stk^2 \Big/ (Stk+0.5)^2$
E_{cs}	Collection efficiency for the coalescence of snow with cloud water	$Stk^2 \Big/ (Stk+0.5)^2$
E_{cg}	Collection efficiency for the coalescence of graupel with cloud water	$Stk^2 \Big/ (Stk+0.5)^2$
E_{rs}	Collection efficiency for the coalescence of snow with rain water	1.0
E_{rg}	Collection efficiency for the coalescence of graupel with rain water	1.0
E_{ir}	Collection efficiency for the coalescence of rain water with cloud ice	1.0

[17]村上 (1999)
[18]Ikawa and Saito (1991), Ikawa et al. (1991)

E_{is}	Collection efficiency for the coalescence of snow with cloud ice	1.0
E_{ig}	Collection efficiency for the coalescence of graupel with cloud ice	0.1
E_{sr}	Collection efficiency for the coalescence of rain water with snow	1.0
E_{sg}	Collection efficiency for the coalescence of graupel with snow	0.001

(5)Category Allocation Ratios of Snow and Raindrop after the Coalescence ä α_{rs}

In the coalescence of snow and raindrop, it is difficult that we decide an adequate category a particle formed by the coalescence in the layer less than 0 °C . Here, the allocation ratios of the snow and graupel, which appear in the equations (8.31) and (8.32), is given by using the average masses of rain water $\bar{m}_r$ and snow $\bar{m}_s$ as follows.

$$\alpha_{rs} = \frac{\bar{m}_s^2}{\bar{m}_s^2 + \bar{m}_r^2} \tag{8.114}$$

The average masses of rain water $\bar{m}_r$ and snow $\bar{m}_s$ are given as follows.

$$\bar{m}_r = \rho_r \left(\frac{4}{\lambda_r}\right)^3 \tag{8.115}$$

$$\bar{m}_s = \rho_s \left(\frac{4}{\lambda_s}\right)^3 \tag{8.116}$$

Using these equations, we can find that the generation ratio of graupel by the coalescence of a raindrop and snow is $(1 - \alpha_{rs})$. Note that the ratios in (8.114) are tentative values.

Production of Graupel ä PG

Production of graupel is an important and complicated problem, since graupel is producted by the collision between other particles. In this section, production process of graupel is explained in detail especially, although it should be explained in the section of "Collision between particles".

Production of graupel has two processes: dry growth and wet growth. By the former process, all supercooled cloud droplets freeze in an instant on contact with graupel, and surface of graupel remains dry. By the latter process, not all supercooled cloud droplets freeze by the latent heat which supercooled cloud droplets emit when they collided with graupel. Surface of graupel gets wet.

In dry growth process, although all supercooled cloud droplets contribute to growth, cloud ice and snow hardly contribute to growth because the collection efficiency of cloud ice or snow colliding with graupel is small. On the contrary, in wet growth process, a freezing amount of

supercooled water is decided by the budget of sensible and latent heat. Cloud ice or snow are efficiently captured by graupel.

Dry growth process is defined as following[19].

$$PG_{dry} = CL_{cg} + CL_{rg} + CL_{ig} + CL_{sg} \tag{8.117}$$

On the contrary, wet growth process is defined as following.

$$PG_{wet} = \frac{2\pi \left[\kappa T_s + \mathcal{L}_v \mathcal{D}_v \bar{\rho} \left(q_{vs}\left(T_0\right) - q_v\right)\right]}{\bar{\rho}\left(\mathcal{L}_f - C_w T_s\right)} VENT_g + \left(CL'_{ig} + CL'_{sg}\right)\left(1 + \frac{C_i T_s}{\mathcal{L}_f - C_w T_s}\right) \tag{8.118}$$

Where $VENT_g$ is ventilation efficiency (8.104). Whether graupel grows by dry growth process or wet growth process is decided by following relation of PG_{dry} and PG_{wet}.

$$PG_g = PG_{dry}, \qquad PG_{dry} \leq PG_{wet} \tag{8.119}$$

$$PG_g = PG_{wet}, \qquad PG_{dry} > PG_{wet} \tag{8.120}$$

Meanings of signs used in this section are shown as follows.

CL_{cg}	Growth rate of cloud water by coalescence with cloud water		s^{-1}
CL_{ig}	Growth rate of cloud ice by coalescence with cloud water		s^{-1}
CL'_{ig}	Growth rate of cloud ice by coalescence with cloud water in wet growth process		s^{-1}
CL_{rg}	Growth rate of rain by collecting cloud water		s^{-1}
CL_{sg}	Growth rate of snow by collecting cloud water		s^{-1}
CL'_{sg}	Growth rate of snow by collecting cloud water in wet growth process		s^{-1}
C_i	Specific heat at constant pressure of ice	2.0×10^3	$J\ K^{-1}kg^{-1}$
C_w	Specific heat at constant pressure of water	4.17×10^3	$J\ K^{-1}kg^{-1}$
$\mathcal{D}_v$	Diffusion coefficient of water vapor		$m^2\ s^{-1}$
$\mathcal{L}_f$	Latent heat of fusion of water		$J\ kg^{-1}$
$\mathcal{L}_v$	Latent heat of evaporation of water		$J\ kg^{-1}$
$q_{vs}\left(T_0\right)$	Saturation mixing ratio at melting point of water point of water		$kg\ kg^{-1}$
T	Temperature		K
T_0	Melting point of ice	273.16	K
T_s	Temperature in supercooling $(T_0 - T)$		K
κ	Efficiency of conduction of heat of air	2.4×10^{-2}	$J\ m^{-1}\ s^{-1}\ K^{-1}$
$\bar{\rho}$	Air Density in the basic state		$kg\ m^{-3}$

[19]村上 (1999)

Aggregation ä AG

Aggregation, as well as production of graupel, should be categorized in "Collision between particles," however, we explain it especially in this section.

We consider the following two phenomenons for aggregation. First, the number of cloud ice per unit volume decreases by the collision between cloud ices (ice crystals). Second, the number of snow per unit volume decreases by the collision between snows (snowflakes).

By aggregation process, only the number concentration changes but mixing ratio does not change.

(1)Aggregation between Cloud Ice Particles ä AG_i^N [20]

Decrease of the number concentration of cloud ices (ice crystals) by aggregation process is defined as following.

$$AG_i^N = \left[\frac{d}{dt}\left(\frac{N_i}{\bar{\rho}}\right)\right]_{aggr} = -\frac{c_1}{2\bar{\rho}}N_i \tag{8.121}$$

where c_1 is defined as following.

$$c_1 = \frac{\bar{\rho} q_i \alpha_{ui} E_{ii} X}{\rho_i}\left(\frac{\rho_0}{\bar{\rho}}\right)^{\frac{1}{3}} \tag{8.122}$$

Meanings of signs used in this section are shown as follows.

E_{ii}	Collection efficiency between ice crystals	0.1	
X	Spectrum dispersion of fall velocity of ice crystal	0.25	
α_{ui}	The coefficient which appears in the equation of fall velocity and diameter of ice crystal	700	$\mathrm{m}^{1-\beta_{ui}}\ \mathrm{s}^{-1}$
β_{ui}	The coefficient which appears in the equation of fall velocity and diameter of ice crystal	1.0	
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_0	Air density at the surface in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_i	Density of cloud ice	5.0×10^2	$\mathrm{kg\ m}^{-3}$

(2)Aggregation between Snow Particles ä AG_s^N [21]

A decrease of the number concentration of snow (snowflakes) by aggregation process is defined as following. By this process, only the number concentration of snow (N_s) changes but mixing ratio of snow (q_s) does not change.

[20]村上 (1999), Ikawa and Saito (1991)
[21]村上 (1999), Ikawa and Saito (1991)

$$AG_s^N = \left[\frac{d}{dt}\left(\frac{N_s}{\bar{\rho}}\right)\right]_{aggr} = -\frac{1}{\bar{\rho}}\frac{\alpha_{us}E_{ss}I\left(\beta_{us}\right)}{4\times 720}\pi^{\frac{1-\beta_{us}}{3}}\,\bar{\rho}^{\frac{2+\beta_{us}}{3}}\,\rho_s^{\frac{-2-\beta_{us}}{3}}\,q_s^{\frac{2+\beta_{us}}{3}}\,N_s^{\frac{4-\beta_{us}}{3}} \tag{8.123}$$

where $I\left(\beta_{us}\right)$ is defined as following.

$$I\left(\beta_{us}\right) = \int_0^\infty\int_0^\infty x^3y^3\left(x+y\right)^2\left|x_{us}^\beta - y_{us}^\beta\right|\exp\left[-\left(x+y\right)\right]\,dxdy \tag{8.124}$$

This equation is solved with Gauss's hypergeometric function as following. [22]

$$I\left(\beta_{us}\right) = \Gamma\left(\beta_{us}\right)2^{1-d}\sum_{i=1}^{3}C_i\left[\frac{F\left(1,d;8-i;0.5\right)}{7-i} - \frac{F\left(1,d;4+\beta_{us};0.5\right)}{3+\beta_{us}+i}\right] \tag{8.129}$$

where

$$\begin{aligned} d &= 10+\beta_{us}\\ C_1 &= 1\\ C_2 &= 3\\ C_3 &= 1. \end{aligned}$$

We adopted the following values as typical values of $I\left(\beta_{us}\right)$ (Ikawa and Saito, 1991; Mizuno, 1990).

[22]The hypergeometric function is one of solutions of hypergeometric equation which is two order linear ordinary differential equation, and has regular sigularity at $x = 0, 1, \infty$.

$$F\left(x,a;b;c\right) = 1 + \frac{a\cdot b}{c}\frac{x}{1!} + \frac{a\left(a+1\right)b\left(b+1\right)}{c\left(c+1\right)}\frac{x^2}{2!} + \cdots \tag{8.125}$$

where $c \neq 0, -1, -2, \cdots$. With Pockhimer's sign,

$$\left(a\right)_n = a\left(a+1\right)\left(a+2\right)\cdots\left(a+n-1\right) = \frac{\left(a+n-1\right)!}{\left(a-1\right)!} \tag{8.126}$$

$$\left(a\right)_0 = 1 \tag{8.127}$$

$$F\left(x,a;b;c\right) = \sum_{n=0}^{\infty}\frac{\left(a\right)_n\left(b\right)_n}{\left(c\right)_n}\frac{x^n}{n!} \tag{8.128}$$

As special limit of this hypergeometric function, there is confluent hypergeometric function

β_{us}	0.4	0.5	0.6
$I(\beta_{us})$	1108	1610	2566

Meanings of signs used in these equations are shown as follows.

E_{ss}	Collection efficiency between snows	0.1	
α_{us}	The coefficient which appears in the equation of fall velocity and diameter of snow	17	$\mathrm{m}^{1-\beta_{us}}\ \mathrm{s}^{-1}$
β_{us}	The coefficient which appears in the equation of fall velocity and diameter of snow	0.5	
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_s	Density of snow	8.4×10^{1}	$\mathrm{kg\ m}^{-3}$

Conversion ä CN

Particle conversion from a certain category 'x' into other category 'y' is expressed by CN_{xy}.

For example, Kessler (1969)'s expression of conversion from cloud water into rain water is one of the earliest formulation of category conversion.

We consider the following types of category conversions.

process	consumed	growing	sign	meaning
conversion from cloud water to rain water	q_c	q_r	CN_{cr}	by aggregation
conversion from cloud ice to snow	q_i	q_s	CN_{is}	by aggregation and sublimation
conversion from snow to graupel	q_s	q_g	CN_{sg}	by growth with capturing cloud particle
conversion from snow to hail	q_s	q_h	CN_{sh}	by growth with capturing cloud droplets
conversion from graupel to hail	q_g	q_h	CN_{gh}	by growth with capturing cloud particle
conversion from snow to graupel	q_s	q_g	CN_{sg}	by freezing of rain water
conversion from snow to hail	q_s	q_h	CN_{sh}	by freezing of rain water
conversion from graupel to hail	q_g	q_h	CN_{gh}	by freezing of rain water

Hereafter, we consider about conversion processes of $CN_{cr}, CN_{is}, CN_{sg}$.

(1)Conversion from Cloud to Rain ä CN_{cr} [23]

The conversion from cloud water to rain water is based on coalescence growth of cloud droplets. Kessler (1969) parameterized this process first. This process is investigated in detail by Berry (1968),and Berry and Reinhardt (1974). Recently, the method for the conversion based on this process is used in many cases.

(a)Conversion Based on Berry(1968), Berry and Reinhardt(1974) [24]

The method for the conversion based on Berry (1968), Berry and Reinhardt (1974) is shown as following.

$$CN_{cr} = \begin{cases} \dfrac{0.104 g E_{cc}}{\mu \left(N_c \rho_w\right)^{\frac{1}{3}}} \left(\bar{\rho}^4 q_c^7\right)^{\frac{1}{3}}, & q_c \geq q_{cm} \\ \\ 0, & q_c < q_{cm} \end{cases} \tag{8.130}$$

where the collection efficiency between cloud water is $E_{cc} = 0.55$. The critical value of mixing ratio of cloud water to convert from cloud water to rain water (q_{cm}) is shown as following.

$$q_{cm} = \frac{\rho_w}{6\bar{\rho}} \pi D_{cm}^3 N_c \tag{8.131}$$

where the critical mean value of diameter of cloud is $D_{cm} = 20\ [\mu m]$. We adopt a typical value ($N_c = 10^8\ [\mathrm{m}^{-3}]$) of the number concentration of cloud (N_c). However, it is better for this to be calculated by time integration as a forecast variable of a time development equation essentially.

(b)Kessler ć 1969 Ĉ 's Definition of Coalescence Growth [25]

Kessler's parametarization is the most classic parameterization, and is also used in many models now.

$$CN_{cr} = a \left(q_c - q_{cm}\right) H \left(q_c - q_{cm}\right) \tag{8.132}$$

where H is set function. Usually a is selected as $10^{-3}\ [s^{-1}]$,q_{cm} is selected as 10^{-3}[kg kg^{-1}]. In Cotton and Anthes (1989), q_{cm} and a are given by the function of q_c as following.

$$q_{cm} = \frac{4\pi \rho_w N_c D_{cm}^3}{3\bar{\rho}} = 4 \times 10^{-12} N_c, \qquad D_{cm} = 10^{-5}\ [m] \tag{8.133}$$

$$a = \pi E_{cc} U_{dc} N_c D_c^2 = 1.3 \times q_c^{\frac{4}{3}} N_c^{-\frac{1}{3}} \left(\frac{\rho_0}{\bar{\rho}}\right) \tag{8.134}$$

[23]村上 (1999), Lin et al. (1983), Ferrier (1994), Ikawa and Saito (1991)
[24]村上 (1991)
[25]Ikawa and Saito (1991)

(c)Lin et al.(1983)'s way

Lin et al. (1983) improved Berry (1968)'s method as following.

$$CN_{cr} = \bar{\rho}\,(q_c - q_{cm})^2 \left[1.2 \times 10^{-4} + 1.569 \times 10^{-12\frac{N_c}{\sigma^2(q_c - q_{cm})}}\right] \tag{8.135}$$

where σ^2 indicates dispersion of the number concentration of cloud (= 0.15). q_{cm} is selected as 2×10^{-3}[kg kg^{-1}].

Meanings of signs used in (a)∼(c) are shown as following.

g	Acceleration of gravity	9.8	m s^{-2}
N_c	Number concentration of cloud particle	1×10^8	m^{-3}
μ	Viscosity Coefficient of air		kg m^{-1} s^{-1}
$\bar{\rho}$	Air density in the basic state		kg m^{-3}
ρ_0	Air density at the surface in the basic state		kg m^{-3}
ρ_w	Density of water	1×10^3	kg m^{-3}

(2)Conversion from Cloud Ice to Snow ä CN_{is}

The conversion from cloud ice to snow is formulized with an assumption of which the conversion occurs by two processes of sublimation growth of cloud ice and condensation[26].

By sublimation condensation growth, it takes Δt_{is1} for ice crystal with $\bar{R}_i$ radius to grow into snow with R_{s0} radius. Δt_{is1} is

$$\Delta t_{is1} = \frac{R_{s0}^2 - \bar{R}_i^2}{2a_1}\rho_i \tag{8.136}$$

where a_1, as well as (8.100) and (8.97), is expressed as following.

$$a_1 = (S_i - 1)\left(\frac{\mathcal{L}_s^2}{\kappa R_v T^2} + \frac{1}{\bar{\rho} q_{vsi} \mathcal{D}_v}\right)^{-1} \tag{8.137}$$

Variation of mixing ratio from ice cloud to snow per unit time by sublimation condensation growth (conversion rate) CN_{is}^{dep} is showed as following.

$$CN_{is}^{dep} = \frac{q_i}{\Delta t_{is1}} \tag{8.138}$$

[26]村上 (1999), Murakami (1990), Ikawa and Saito (1991)

By aggregation growth, it takes Δt_{is1} for cloud ice with $\bar{R}_i$ radius to grow into snow with R_{s0} radius. We suppose that the Δt_{is1} is the same time as it takes for the number concentration of cloud ice to decrease from N_i to $N_i \left(R_i / R_{s0}\right)^3$ on condition $\rho_i = const.$. Therefore, the Δt_{is1} is expressed as following.

$$\Delta t_{is2} = \frac{2}{c_1} \log \left(\frac{R_{s0}}{\bar{R}_i} \right)^3 \tag{8.139}$$

where c_1 is given by the equation (8.122). Variation of mixing ratio from ice cloud to snow per unit time by aggregation growth (conversion rate) is showed as following.

$$CN_{is}^{agg} = \frac{q_i}{\Delta t_{is2}} \tag{8.140}$$

Finaly, the conversion rate from cloud ice to snow (CN_{is}) is given as following.

$$CN_{is} = CN_{is}^{dep} + CN_{is}^{agg} \tag{8.141}$$

Meaning of signs used here are shown as following.

$\mathcal{D}_v$	Diffusion coefficient of water vapor		$\mathrm{m^2\ s^{-1}}$
$\mathcal{L}_s$	Latent heat of sublimation of water		$\mathrm{J\ kg^{-1}}$
q_{vsi}	Saturation mixing ratio over ice		$\mathrm{kg\ kg^{-1}}$
R_v	Gas constant of water vapor	461.0	$\mathrm{J\ K^{-1}\ kg^{-1}}$
$S_i - 1$	Supersaturation of air to ice		
T	Temperature		K
κ	Conduction efficiency of heat of air	2.4×10^{-2}	$\mathrm{J\ m^{-1}\ s^{-1}\ K^{-1}}$
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m^{-3}}$
ρ_i	Density of cloud ice	5.0×10^{2}	$\mathrm{kg\ m^{-3}}$

(3)Conversion from Snow to Graupel: CN_{sg}

We suppose that the conversion from snow to graupel takes place when CL_{cs} (growth rate by caputuring cloud droplets), which is given in (8.109), exceeds VD_{vs} (growth rate by sublimation condensation), which is given in (8.100)[27]。 When the conversion takes place, the rate of change of mixing ratio per unit time by converting from snow to graupel is shown as following.

[27]村上 (1999), Murakami (1990)

When $CL_{cs} > VD_{vs}$,

$$CN_{sg} = CL_{cs}\frac{\rho_s}{\rho_g - \rho_s}. \tag{8.142}$$

The variation of the number concentration is shown here,

$$CN_{sg}^{N} = \frac{3\alpha_{us}E_{cs}q_c\left(\rho_0\bar{\rho}\right)^{\frac{1}{2}}\Gamma\left(\beta_{us}\right)}{2\bar{\rho}\left(\rho_g - \rho_s\right)}\lambda_s^{1-\beta_{us}}N_s. \tag{8.143}$$

Meanings of signs used here are shown as following.

E_{cs}	Collection efficiency of snow in the coalescence with cloud ice	$Stk^2 \Big/ (Stk + 0.5)^2$	
α_{us}	The coefficient which appears in the equation of fall velocity and a diameter of snow.	17	$\mathrm{m}^{1-\beta_{us}}\ \mathrm{s}^{-1}$
β_{us}	The coefficient which appears in the equation of fall velocity and a diameter of snow	0.5	
λ_s	Inclination in the reverse exponential function expressing particle size distribution of snow		m^{-1}
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_0	Air density at the surface in the basic state		$\mathrm{kg\ m}^{-3}$
ρ_g	Density of graupel	3.0×10^2	$\mathrm{kg\ m}^{-3}$
ρ_s	Density of snow	8.4×10^1	$\mathrm{kg\ m}^{-3}$

Melting of Solid Particle: ML

(1)Melting of Cloud Ice: ML_{ic}

We suppose that cloud ice melts and is converted into cloud water in an instant when $T > T_0$, because the radius of cloud ice (ice crystal) is very small.

When $T_c > T_0$,

$$ML_{ic} = \frac{q_i}{2\Delta t}. \tag{8.144}$$

(2)Melting of Snow and Graupel: ML_{sr}, ML_{gr}

With regard to snow and graupel, ML_{xr} $(x = s, g)$, which indicates the conversion rate from solid particles to rain, is controled by heat budget[28], that is,

When $T > T_0$,

$$ML_{xr} = \frac{2\pi}{\bar{\rho}\mathcal{L}_f}\left[\kappa T_c + \mathcal{L}_v\mathcal{D}_v\bar{\rho}\left(q_v - q_{vs}\left(T_0\right)\right)\right]VENT_x + \frac{C_wT_c}{\mathcal{L}_f}\left(CL_{cx} + CL_{rx}\right). \tag{8.145}$$

If $ML_{xr} > 0$, snow or graupel melts and is converted to rain, however if $ML_{xr} < 0$, even if $T > T_0$, melting does not take place, that is, $ML_{xr} = 0$. $VENT_g$ is ventiration efficiency(8.104).

Meanings of signs used in (1),(2) are shown as following.

CL_{cx}	Growth rate of a particle in x catogory by collision with cloud water		s^{-1}
CL_{rx}	Growth rate of a particle in x catogory by collision with rain water		s^{-1}
C_w	Specific heat at constant pressure of water	4.17×10^3	$\mathrm{J\ K^{-1}\ kg^{-1}}$
$\mathcal{D}_v$	Diffusion coefficient of water vapor		$\mathrm{m^2\ s^{-1}}$
$\mathcal{L}_f$	Latent heat of fusion of water		$\mathrm{J\ kg^{-1}}$
$\mathcal{L}_v$	Latent heat of evaporation of water		$\mathrm{J\ kg^{-1}}$
$q_{vs}\left(T_0\right)$	Saturated mixing ratio at melting point of water		$\mathrm{kg\ kg^{-1}}$
T	Temperature		K
T_0	Melting point of ice	273.16	K
T_c	Temperature in Celsius scale		°C
κ	Efficiency of conduction of heat of air	2.4×10^{-2}	$\mathrm{J\ m^{-1}\ s^{-1}\ K^{-1}}$
$\bar{\rho}$	Air density in the basic state		$\mathrm{kg\ m^{-3}}$

Freezing of Rain: FR

We suppose that freezing of rain takes place in an instant. All frozen rain is categorized into graupel because we do not consider hail. If there is a category of hail, frozen rain is categorized into the category of hail.

The rate of variation of mixing ratio by freezing rain (FR_{rg}) is based on Bigg (1953)'s empirical equation[29], and is given as following.

$$FR_{rg} = 20\pi^2B'n_{r0}\frac{\rho_w}{\bar{\rho}}\left[\exp\left(A'T_s\right) - 1\right]\lambda_r^{-7} \tag{8.146}$$

The rate of variation of the number concentration is given as following.

$$FR_{rg}^N = \frac{\pi}{6\bar{\rho}}B'n_{r0}\left[\exp\left(A'T_s\right) - 1\right]\lambda_r^{-4} \tag{8.147}$$

[28]Ikawa and Saito (1991)

[29]Lin et al. (1983), 村上 (1999)

Meanings of signs used here are shown as following.

A'	A coefficient in Bigg's empirical equation	0.66	K^{-1}
B'	A coefficient in Bigg's empirical equation	100.0	m^{-3} s^{-1}
n_{r0}	Density of rain in y section	8.0×10^6	m^{-4}
T	Temperature		K
T_0	Melting point of ice	273.16	K
T_s	Supercooled temperature $(T_0 - T)$		K
λ_r	Inclination in the reverse exponential function expressing particle size distribution of rain		m^{-1}
$\bar{\rho}$	Air density in the basic state		kg m^{-3}
ρ_w	Density of water	$1{\times}10^3$	kg m^{-3}

Shedding of Water from Snow and Graupel: SH

In our model, we do not consider that snow and graupel partly contain liquid water in them. Therefore, melted water sheds from snow or ice, and becomes rain water. The process is expressed as following.

When $T > T_0$,

$$SH_{sr} = CL_{cs} + CL_{rs} \tag{8.148}$$

$$SH_{gr} = CL_{cg} + CL_{rg} \tag{8.149}$$

If graupel grows by the wet growth process (even when $T \leq T_0$),

$$SH_{gr} = CL_{cg} + CL_{rg} + CL'_{ig} + CL'_{sg} - PG_{wet}. \tag{8.150}$$

Meanings of signs used here are shown as following.

CL_{cg}	Growth rate of graupel by collision with cloud water		s^{-1}
CL_{cs}	Growth rate of snow by collision with cloud water		s^{-1}
CL_{rg}	Growth rate of graupel by collision with rain water		s^{-1}
CL_{rs}	Growth rate of snow by collision with rain water		s^{-1}
CL'_{ig}	Growth rate of graupel by collision with cloud water in wet growth process		s^{-1}
CL'_{sg}	Growth rate of graupel by collision with snow in wet growth process		s^{-1}
PG_{wet}	Growth rate of graupel in wet growth process		s^{-1}
T	Temperature		K
T_0	Melting point of ice	273.16	K

Ferrier (1994) formulizes the wet growth process of snow, graupel and hail (frozen ice) with a time development equation, and takes the wet snow and wet graupel and wet hail into consideration.

Such formulization will be required in the future.

Breakup of Waterdrop

It is known that if it becomes large to some extent (it is 8mm at the diameter of a sphere), rain water becomes unstable and breaks up.

From this reason, the waterdrop more than the size of 8 mm does not fall down in an actual nature. Although the breakup of waterdrop does not change mixing ratio of rain water, it changes number concentration of rain water. However, since the number concentration of rain water is not predicted in this model, the process is not considered.

8.2.5 Moist Saturation Adjustment

The exchange between water vapor and cloud water is expressed by the moist saturation adjustment[30].

We add the mark * to the variable which is calculated by forecast equation. The supersaturated mixing ratio which is defined as

$$\Delta q_c = q_v^* - q_{vsw}^*. \tag{8.151}$$

It $\Delta q_c > 0$, the moist saturation adjustment is applied. The the moist saturation adjustment caluculates $\theta,\ q_v,\ q_c$ with following way.

$$\theta^{t+\Delta t} = \theta^* + \gamma\left(q_v^* - q_{vsw}^*\right) \Big/ \left[1 + \gamma q_{vsw}^* \frac{4093\Pi}{\left(\Pi\theta^* - 36\right)^2}\right] \tag{8.152}$$

$$q_v^{t+\Delta t} = q_v^* + \left(\theta^* - \theta^{t+\Delta t}\right) \Big/ \gamma \tag{8.153}$$

$$q_c^{t+\Delta t} = q_v^* + q_c^* - q_v^{t+\Delta t} \tag{8.154}$$

First, provisional $\theta,\ q_v,\ q_c$ are calculated with these equations. If $q_c^{t+\Delta t} > 0$, the calculation with these equations (8.152)$\sim$(8.154) is repeatedly done until when the value of $\theta,\ q_v,\ q_c$ converges. Therefore, final $\theta,\ q_v,\ q_c$ is decided by the moist saturation adjustment. Here, we defined γ as $\gamma \equiv \mathcal{L}_v / \left(C_p\Pi\right)$.

If $\Delta q_c \leq 0$, we apply following equations.

$$\theta^{t+\Delta t} = \theta^* - \gamma q_c^* \tag{8.155}$$

$$q_v^{t+\Delta t} = q_v^* + q_c^* \tag{8.156}$$

$$q_c^{t+\Delta t} = 0 \tag{8.157}$$

[30]Soong and Ogura (1973)

Meanings of signs used here are shown as following.

C_p	Specific heat at constant pressure of dry air	1004	J K kg^{-1}
$\mathcal{L}_v$	Latent heat of evaporation of water		J kg^{-1}
q_{vsw}	Saturation mixing ratio over water		kg kg^{-1}
Π	Exner function		

8.2.6 Formulization of Variational Terms on Mixing Ratio and Number Concentration with Precipitation

The rate of variation of mixing ratio by fall of particles of cloud and precipitation are shown as following.

$$\text{Fall.}q_x = \frac{1}{\bar{\rho}} \frac{\partial \bar{\rho} \bar{U}_{xq} q_x}{\partial z} \tag{8.158}$$

The mass-weighted terminal fall velocity of particle in category x ($\bar{U}_{xq}$) is given by the equation (8.68) in the section 8.2.3.

The rate of variation of the number concentration is shown as following.

$$\text{Fall.}\frac{N_x}{\bar{\rho}} = \frac{1}{\bar{\rho}} \frac{\partial N_x \bar{U}_{xN}}{\partial z} \tag{8.159}$$

$\bar{U}_{xN}$ indicates mean fall velocity of particle in category x. Similarly, it is given by the equation (8.67) in the section 8.2.3.

When the grid interval in the vertical direction is small, Δt must be enough small so that CFL condition is filled, and so that a falling precipitation particle should not pass next grid for 1 time step.

Since the differentiation in the vertical direction is the differentiation in real space, the differentiation in z^* coordinate （ζ coordinate） must be multiplied by a metric, as shown in equation (6.58).

8.2.7 The other constants

Here, we show variables and constants, which are used in equations but are not explained.

Water Saturation Mixing Ratio and Ice Saturation Mixing Ratio ä q_{vsw}, q_{vsi}[31]

$$q_{vsw} = \epsilon \frac{610.78}{p} \exp\left(17.269 \frac{T - T_0}{T - 35.86}\right) \tag{8.160}$$

$$q_{vsi} = \epsilon \frac{610.78}{p} \exp\left(21.875 \frac{T - T_0}{T - 7.86}\right) \tag{8.161}$$

[31] Orville and Kopp (1977), Murray (1966)

Latent Heat with Evaporation, Sublimation and Melting of Water ä $\mathcal{L}_v, \mathcal{L}_s, \mathcal{L}_f$ [J kg^{-1}]

$$\mathcal{L}_v = 2.50078 \times 10^6 \left(\frac{T_0}{T}\right)^{\left(0.167+3.67\times10^{-4}T\right)} \tag{8.162}$$

$$\mathcal{L}_s = 2.834 \times 10^6 + 100\,(T - T_0) \tag{8.163}$$

$$\mathcal{L}_f = 3.34 \times 10^5 + 2500\,(T - T_0) \tag{8.164}$$

Coefficient of Kinematic Viscosity of Air ä ν [m^2 s^{-1}]

$$\nu = 1.328 \times 10^{-5} \frac{p_0}{p} \left(\frac{T}{T_0}\right)^{1.754} \tag{8.165}$$

Coefficient of Viscosity of Air: μ [kg m^{-1} s^{-1}]

$$\mu = \rho\nu \tag{8.166}$$

Coefficient of Water Vapor Diffusion: $\mathcal{D}_v$ [m^2 s^{-1}]

$$\mathcal{D}_v = 2.23 \times 10^{-5} \frac{p_0}{p} \left(\frac{T}{T_0}\right)^{1.81} \tag{8.167}$$

Meanings of signs used here are shown as following.

p	Pressure		Pa
p_0	Standard pressure	101325	Pa
T	Temperature		K
T_0	Melting point of ice	273.16	K
ϵ	Ratio of molecular weight of water vapor to molecular weight of dry air	0.622	
ρ	Air density		kg m^{-3}

Chapter 9

Boundery Layer and Surface Process

The integration time of a cloud model is short ranging from few to ten hours. Therefore, in the model, most of phenomena provided by the atmospheric processes. When energy flux from the ground surface (land or sea) is large in a cloud / precipitation system as in the case of a thunderstorm in summer or a snow cloud over Japan Sea in winter, however, ground surface process is substantially important for the development of the system.
Following processes are considered as the ground surface processes and they are expressed by simple bulk methods.

- vertical one-dimensional diffusion in the atmospheric boundary layer
- fluxes of sensible heat and latent heat from the ground to atmosphere
- momentum flux (friction of the ground surface)
- soil temperature variation

In CReSS, these processes are expressed with vertical one-dimensional equations. We adopt the ground surface flux with a bulk method and soil temperature forecasting which are used in JSM (Japan Spectral Model; Segami et al., 1989). Complicated soil and vegetation models are incorporated in some cloud models (e.g. RAMS, ARPS). However, they are important only when integration for a long time (month order) in a wide area (e.g. the region covering all the Eurasian Continent) is conducted. Therefore, the complicated soil and vegitation models will be added to ***CReSS*** in the future.

9.1 Basic Theories of Atmospheric Boundary Layer

Here, we summarize briefly basic theories for the atmospheric boundary layer processes[1].

9.1.1 Structure of Atmospheric Boundary Layer

Atmosphere is bordered on the lower boundary by the ground surface. Generally, effects of the ground surface which includes various conditions like land and sea become stronger as the altitude becomes lower. The effects exist from the surface to about 1-2 km high though the depth is affected with conditions of the ground surface and atmosphere. Such atmospheric layer which is thermodynamically and dynamically affected with the ground surface is called atmospheric boundary layer or planetary boundary layer; PBL.

Atmospheric boundary layer is devided into two layers; surface boundary layer which ranges from the surface to 20-50 m high and is affected by the ground surface, and Ekman layer or outer boundary layer above the surface boundary layer. Atmosphere above the atmospheric boundary layer is called free atmosphere.

The surface boundary layer has a property that vertical fluxes of sensible heat, latent heat, momentum and so on are uniform vertically and equal to their values on the ground. In other words, the atmospheric layer where these vertical fluxes are equal to the values on the ground can be called the surface boundary layer. The surface boundary layer is sometimes called constant flux layer. Because momentum flux is vertically constant, stress of turbulence is vertically constant in the surface boundary layer, that is, wind direction is also constant vertically.

9.1.2 Parameterization of Turbulence Transportation

Motions of various scale exist in the atmosphere. Among them, motion which can be expressed on the grid points of a numerical model is called grid-scale motion or mean motion. Motion which has a scale smaller than a grid size is called subgrid-scale motion or eddy motion.

To classify these motions, physical quantity A (e.g. wind speed, temperature, mixing ratio of water vapor) is devided into two quantity; averaged quantity which can be expressed on the grid points and deviation from that, as follows.

$$A = \bar{A} + A'' \tag{9.1}$$

There are some ways to average physical quantities, we will not mentione about them here. Average of deviation and that of product of two physical quantities are expressed as follows;

$$\overline{A''} = 0 \tag{9.2}$$

$$\overline{AB} = \bar{A}\bar{B} + \overline{A''B''} \tag{9.3}$$

The average of the pruducts of some quantities is not always equal to the product of averages of each quantity, and the second term of the equation above appears. This equation is applied to

[1]Introduction to dynamic meteorology (Ogura, 1978) and Atmospheric Science Course 1 - Atmosphere near the ground surface (Takeuchi and Kondo) are mainly reffered here.

the x component of equations of motion. Considering an uncompressed fluid ($\rho = const.$), each term of the equation can be expressed with the average and deviation from that as follows;

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} + \bar{w}\frac{\partial \bar{u}}{\partial z} - f\bar{v} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x} - \frac{1}{\rho}\left(\frac{\partial}{\partial x}\rho\overline{u''u''} + \frac{\partial}{\partial y}\rho\overline{u''v''} + \frac{\partial}{\partial z}\rho\overline{u''w''}\right) + \nu\nabla^2\bar{u}. \tag{9.4}$$

The terms of $-\rho\overline{u''u''}, -\rho\overline{u''v''} and -\rho\overline{u''w''}$ express the stress of turbulence and are called eddy stress or Reynolds stress. These stresses can be regarded as a momentum transportation, and the momentum transportation by eddy is the stress.

Similarly, equations of potential temperature and mixing ratio of water vapor are as follows.

$$\frac{\partial \bar{\theta}}{\partial t} + \bar{u}\frac{\partial \bar{\theta}}{\partial x} + \bar{v}\frac{\partial \bar{\theta}}{\partial y} + \bar{w}\frac{\partial \bar{\theta}}{\partial z} = -\frac{\partial}{\partial x}\overline{u''\theta''} - \frac{\partial}{\partial y}\overline{v''\theta''} - \frac{\partial}{\partial z}\overline{w''\theta''} \tag{9.5}$$

The quantities with a prime cannot be expressed with the grid points. It is neccessary to express them with quantities on the grid points (averaged quantities) for including effects of momentum transportation by eddy which is smaller than the grid size. A problem of turbulent parameterization is to decide the way to express deviation of quantity by eddy with prime on the grid points. Note that definitions of quantities with $^-$ used in this chapter and in the other chapter (e.g. Chapter 2) are different.

9.1.3 Mixing Length Theory

As mentioned in the chapter of turbulence, atmospheric motions should have a motion which is smaller than the grid size. Therefore, the equation system is not closed due to correlation quantities of deviation as in (9.4) and (9.5). This is called closure problem and the terms of transportation by turbulence must be expressed as functions of quantities on grid points to close the equation system. One of the turbulent parameterization methods is mixing length theory by Prandtl (1925). This is explained by an analogy of kinetic theory of gases as follows.

We assume that the mean vertical speed of air mass at the altitude of $z - l''$ is $\bar{u}(z - l'')$, and gives momentum to the environment when the air mass moves upward for l'' reaching to the altitude of z. The variation of the speed is

$$u'' = \bar{u}(z - l'') - \bar{u}(z) \cong -l''\frac{\partial \bar{u}}{\partial z}. \tag{9.6}$$

Raynolds tension is

$$\tau_{zx} = -\rho\overline{u''w''} = \rho\overline{l''w''}\frac{\partial \bar{u}}{\partial z} = \rho K_{mz}\frac{\partial \bar{u}}{\partial z} \tag{9.7}$$

where

$$K_{mz} = \overline{l'' w''} \tag{9.8}$$

is eddy viscosity coefficient.

Considering fluid with neutral stratification, the effect of buoyancy can be neglected, and it can be assumed $|w''| \sim |u''|$. We can assume the following relation.

$$w'' \sim u'' \cong -l'' \frac{\partial \bar{u}}{\partial z} \tag{9.9}$$

Then,

$$\tau_{zx} = -\rho \overline{u'' w''} = \rho \overline{l''^2} \frac{\partial \bar{u}}{\partial z} \left| \frac{\partial \bar{u}}{\partial z} \right|. \tag{9.10}$$

and,

$$K_{mz} = \overline{l''^2} \left| \frac{\partial \bar{u}}{\partial z} \right|. \tag{9.11}$$

Using these equations,

$$-\frac{1}{\rho} \frac{\partial}{\partial z} \rho \overline{u'' w''} = -\frac{1}{\rho} \frac{\partial}{\partial z} \left(\rho K_{mz} \frac{\partial \bar{u}}{\partial z} \right) \tag{9.12}$$

$$-\frac{\partial}{\partial z} \overline{w'' \theta''} = -\frac{\partial}{\partial z} \left(K_{hz} \frac{\partial \bar{\theta}}{\partial z} \right) \tag{9.13}$$

are gained.

As mentioned above, the transportation of physical quantities with turbulence could be explained by using mean field based on an analogy of kinetic theory of gases. That is called Prandtl's mixing length theory. It is difficult to apply generally this theory for turbulent transportation. For the surface boundary layer, we can apply the theory as a basic theory to calculate ground surface flux. The other theories are needed to solve a closure problem for the whole of the atmospheric boundary layer including Ekman layer.

9.1.4 Surface Boundary Layer

The surface boundary layer is called constant flux layer where a value of vertical momentum flux at all altitudes is equal to that of the ground surface, and stress is constant. Therefore, wind direction is constant in the surface boundary layer. We set that the direction of an averaged wind is the x direction. Considering (9.10),

$$\frac{\partial \bar{u}}{\partial z} = \frac{1}{l}\sqrt{\frac{\tau_{zx}}{\rho}} \tag{9.14}$$

where $l = \sqrt{\overline{l''^2}}$. The size of turbulence which transports momentum within the surface boundary layer is restricted by the ground surface. The mixing length l is assumed as follows;

$$l = k\ z \tag{9.15}$$

where z is the distance from the surface and k is nondimensional quantity called Karman constant. k is decided as $k = 0.4$ from wind tunnel experiments and observations[2].

We define u_* as follows;

$$u_*^2 \equiv \frac{\tau_{zx}}{\rho} = -\overline{u''w''}. \tag{9.16}$$

This is called friction velocity and has a dimension of velocity, corresponding to the scale of turbulent strength.

Observations showed that the amount of vertical transportation of momentum (or stress) is constant to altitudes within the surface boundary layer. Using these equations,

$$u_* = \sqrt{\frac{\tau_{zx}}{\rho}} = kz\frac{\partial \bar{u}}{\partial z} = const. \tag{9.17}$$

is gained. Integrating this equation for z,

$$\bar{u} = \frac{u_*}{k}\ln z + C. \tag{9.18}$$

Since the actual ground surface has various irregurality, it can not say that $\bar{u} = 0$ at $z = 0$. Then, the constant of integration C is decided as the value which makes $\bar{u} = 0$ at $z = Z_{0m}$.

$$\bar{u} = \frac{u_*}{k}\ln\frac{z}{Z_{0m}} \tag{9.19}$$

[2]Karman constant is decided as $k = 0.4 \pm 0.01$ by simultaneous observations of Raynolds stress $\tau = -\rho\overline{u''w''}$ and vertical profile of wind speed (Meteorology of water environment, p99, Kondo).

This is the logarithmic profile of wind speed. Z_{0m} means the extent of roughness on the ground and is called aerodynamical roughness parameter (or length) or ground surface roughness for wind profile. The logarithmic profile expressed with this equation corresponds well with the actual wind profile when the atmosphere has a neutral stratification.

Eddy viscosity coefficient in the surface boundary layer K_{mz} can be expressed by (9.11) and (9.17) as follows;

$$K_{mz} = (kz)^2 \frac{u_*}{kz} = kzu_*. \tag{9.20}$$

9.1.5 Vertical Flux

Momentum flux and Bulk coefficient

In the surface boundary layer, vertical flux is constant for altitude and it is enough to estimate flux at an altitude in the surface boundary layer for the estimation of flux on the ground. Momentum flux, i.e., wind stress on the ground is as follows;

$$\tau_0 = -\rho\overline{u''w''} = \rho u_*^2 = \rho\bar{u}^2 \left[\frac{k}{\ln \dfrac{z}{Z_{0m}}}\right]^2 = \rho C_m \bar{u}^2 \tag{9.21}$$

$\bar{u}$ can be gained by measuring average wind speed at an altitude (usually 10 m) in the surface boudnary layer. C_m defined as

$$C_m = \left[\frac{k}{\ln \dfrac{z}{Z_{0m}}}\right]^2 \tag{9.22}$$

is nondimensional quantity and called drag coefficient or bulk transfer coefficient.

Potential Temperature Flux

In the surface boundary layer, it can be considered that the amount of transportations (fluxes) of sensible heat and latent heat H_s, H_L are constant at all altitudes. If turbulence is decided by u_* and z in the surface boundary layer, equation which appears in (9.13);

$$K_{mz}\frac{\partial\bar{\theta}}{\partial z} = \overline{w''\theta''} \tag{9.23}$$

is expressed as follows using eddy viscosity coefficient (9.20) as in the case of momentum;

$$\frac{\partial \bar{\theta}}{\partial z} = \frac{\overline{w''\theta''}}{kzu_*} \tag{9.24}$$

Sensible heat flux from the ground H_s $[\mathrm{J\ m^{-2}\ s^{-1}}]$ is offered as follows;

$$H_s = -\rho C_p \overline{w''\theta''} \tag{9.25}$$

where potential temperature scale θ_* which is defined by

$$\theta_* = \frac{\overline{w''\theta''}}{u_*} \tag{9.26}$$

and potential temperature flux is expressed as follows;

$$\frac{H_s}{\rho C_p} = -\overline{w''\theta''} = -kz\frac{\partial \bar{\theta}}{\partial z}u_* = -\theta_* u_* \tag{9.27}$$

θ_* defined here is constant in the surface boundary layer. Therefore,

$$\frac{\partial \bar{\theta}}{\partial z} = \frac{\theta_*}{kz} \tag{9.28}$$

is obtained. By integrating this vertically and defining Z_{0h} as the altitude of $\bar{\theta} = \theta_G$ as in the case of momentum,

$$\bar{\theta} - \theta_G = \frac{\theta_*}{k} \ln \frac{z}{Z_{0h}} \tag{9.29}$$

is gained. Potential temperature flux is expressed with this as follows;

$$\frac{H_s}{\rho C_p} = -\overline{w''\theta''} = -\theta_* u_* = -\frac{k}{\ln \dfrac{z}{Z_{0h}}} \frac{k}{\ln \dfrac{z}{Z_{0m}}} (\bar{\theta} - \theta_G)\bar{u} \tag{9.30}$$

where

$$\frac{H_s}{\rho C_p} = -\frac{k}{\ln \dfrac{z}{Z_{0h}}} \frac{k}{\ln \dfrac{z}{Z_{0m}}} (\bar{\theta} - \theta_G)\bar{u} = -C_h(\bar{\theta} - \theta_G)\bar{u}. \tag{9.31}$$

Bulk coefficients of sensible heat and potential temperature can be defined as follows;

$$C_h = \frac{k}{\ln \dfrac{z}{Z_{0h}}} \frac{k}{\ln \dfrac{z}{Z_{0m}}}. \tag{9.32}$$

Mixing Ratio of Water Vapor Flux

Latent heat flux H_L can be regarded to be constant in the surface boundary layer,

$$H_L = {}_l E = -\mathcal{L}_v \overline{w'' q_v''} \tag{9.33}$$

and

$$\frac{\partial \bar{q}_v}{\partial z} = \frac{\overline{w'' q_v''}}{kzu_*} \tag{9.34}$$

where L is latent heat of evaporation. Flux of mixing ratio of water vapor is expressed as follows.

$$E = -\overline{w'' q_v''} = -kzu_* \frac{\partial \bar{q}_v}{\partial z} \tag{9.35}$$

In the same way, we define the scale of mixing ratio of water vapor (friction mixing ratio) q_{v*} as

$$q_{v*} = \frac{\overline{w'' q_v''}}{u_*} \tag{9.36}$$

and flux of mixing ratio of water vapor is expressed as

$$E = -\overline{w'' q_v''} = -kz \frac{\partial \bar{q}_v}{\partial z} u_* = -q_{v*} u_* \tag{9.37}$$

where q_{v*} defined here is also constant in the surface boundary layer. By integrating this vertically and defining an altitude Z_{0h} of which $\bar{q}_v = q_{vsfc}$ as same as the case of momentum and potential temperature,

$$\bar{q}_v - q_{vsfc} = \frac{q_{v*}}{k} \ln \frac{z}{Z_{0h}} \tag{9.38}$$

is given. This equation is expressed with Mixing ratio of water vapor as follows;

$$-\overline{w'' q_v''} = -q_{v*} u_* = -\frac{k}{\ln \frac{z}{Z_{0h}}} \frac{k}{\ln \frac{z}{Z_{0m}}} (\bar{q}_v - q_{vsfc}) \bar{u} \tag{9.39}$$

This is expressed by using the bulk coefficient of potential temperature defined by (9.32) as follows;

$$-\frac{k}{\ln \frac{z}{Z_{0h}}} \frac{k}{\ln \frac{z}{Z_{0m}}} (\bar{q}_v - q_{vsfc}) \bar{u} = -C_h (\bar{q}_v - q_{vsfc}) \bar{u} \tag{9.40}$$

Vertical flux of momentum, potential temperature and water vapor mixing ratio in the atmospheric boundary layer have been described with bulk coefficient. In the ground surface process of ***CReSS***, the exchange of sensible and latent heat between land and atmosphere is described by calculating these bulk coefficients. The effect of ground surface friction is included in a caluculation of momentum flux.

9.2 Boundary Layer, Surface Process and Soil Temperature

The ways of calculation used in JSM (Japan Spectral Model) by Segami et al. (1989) are basically adopted in the calculations of atmospheric boundary layer, ground surface process and soil temperature in ***CReSS***Their calculations are one-dimensional vertically. However, following main differences exist between the calculations in ***CReSS***and those in JSM.

- On the calculation of soil temperature, a four layer model is used in JSM but we can select an arbitrary number (m) of layers in ***CReSS***.
- On the calculation of soil temperature, it starts at $n-1$ step and uses a complicated calculation method adopting differential of flux on temperature in JSM. On the other hand in ***CReSS***, the calculation method is a simple implicit one, just solving simultaneous equations.
- Cloudiness is estimated from relative humidity by the method of Ohno and Isa (1983) in JSM, but it is estimated from mixing ratio of cloud water and cloud ice in ***CReSS***.

- In JSM, quantities of atmosphere are calculated at first and after that, sensible heat flux and latent heat flux on the ground are calculated from temperature and relative humidity of the atmosphere which are modified by the first calculation of the atmosphere. In ***CReSS***on the other hand, they are calculated from temperature and mixing ratio of water vapor of $n-1$ step and the result is applied for the atmosphere and the ground.
- The calculation of vertical diffusion is conducted for all layers of the atmosphere at every step in JSM and flux entering the atmosphere from the ground affects instantly (within the time step of Δt) to top of the atmosphere. In ***CReSS***, the layers affected by the flux can be designed arbitrary.
- On the calculation of vertical diffusion, an implicit method modifying dependent variables is used in JSM but an explicit method is used in ***CReSS***. Rate of change for time can be calculated and it is added as a forcing term for a time integration.

The calculations of the atmospheric boundary layer, ground surface process and soil temperature are conducted mainly as follows. They are independently conducted for the vertical one-dimension on each horizontal grid.

- Cloudiness is calculated. Usually cloudinesses are calculated for upper, middle and lower layers.
- Solar radiation and downward long wave radiation are calculated.
- Coefficients of fluxes of momentum, sensible heat and latent heat on the ground surface are calculated.
- Horizontal wind speed, potential temperature and mixing ratio of water vapor are modified by calculation of the atmospheric boundary layer process.
- Soil temperature is calculated.

It is notable that, on the ground surface process, flux is not calculated but its coefficient is calculated. The reason is that sensible and latent heat fluxes,

$$H_S = -\rho_a C_p C_h |V_a|(T_a - T_G) \tag{9.41}$$

$$_lE = -\rho_a \mathcal{L}_v C_h |V_a| \beta \left[q_{va} - q_{vs}(T_G)\right] \tag{9.42}$$

are given as external forces on the ground surface, and potential temperature flux [3] and flux of mixing ratio of water vapor,

$$F_\theta = -\rho_a C_h |V_a|(\theta_{va} - \theta_{vG}) \tag{9.43}$$

$$F_{q_v} = -\rho_a C_h |V_a| \beta \left[q_{va} - q_{vs}(T_G)\right] \tag{9.44}$$

[3]Virtural potential temperature is used here instead of potential temperature. It makes no difference because flux is proportional to the difference of quantities. However, it is notable that it makes big difference when they are used to calculate CAPE.

are given as external forces from the ground surface in the atmosphere. Coefficients common to the two above $\rho_a C_h |V_a|$ (actually it is different between for potential temperature and for mixing ratio of water vapor) are calculated in the ground surface process and are given to the processes of the atmospheric boundary layer and soil temperature. Then sensible and latent heat fluxes are included in the process of soil temperature, while the effects of fluxes of potential temperature and mixing ratio of water vapor are included in the process of the atmospheric boundary layer. Mixing ratio of water vapor corresponding to the ground surface temperature is calculated in the ground surface process and given to each process simultaneously.

Processes for the calculation of the atmospheric boundary layer, the ground surface process and soil temperature are explained in the next section.

9.2.1 Cloudiness

The amount of solar radiation reaching to the ground decreases by the effect of cloud. Cloudiness is used to estimate this effect. The way of Ohno and Isa (1984) to estimate it by an empirical formula of cloudiness and relative humidity is explained in this section.

Cloudiness is nondimensional quantity from 0 to 1. Cloudiness is given as a function of relative humidity for upper, middle and lower layers, respectively. The height of each layer is not exact and 400hPa is for the upper layer, 500 and 700hPa for the middle layer and 850hPa for the lower layer in Ohno and Isa (1984). Upper layer corresponds to 7-7.5km, middle corresponds to 5-6 and 3km, and lower corresponds to 1.5km.

Cloudiness is given by 21 data every 5% ranging from 0 to 100% from the result of Ohno and Isa (1984). When average relative humidity in each layer is defined as $\overline{Rh}$[%], an integer k_n is defined as follows;

$$k_n = 1 + 20 \times \frac{\overline{Rh}}{100} \tag{9.45}$$

and cloudiness of lower layer CD_L、middle layer CD_M and upper layer CD_H is defined as follows (written in the format of `data` description of Fortran);

```
 data cdldef / 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0,
.              0.e0, 0.e0, 0.e0, 0.e0,.07e0,.11e0,.19e0,.40e0,
.             .85e0, 1.e0, 1.e0, 1.e0, 1.e0 /

 data cdmdef / 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0,
.              0.e0, 0.e0, 0.e0,.05e0,.12e0,.30e0,.40e0,.50e0,
.             .70e0,.95e0, 1.e0, 1.e0, 1.e0 /

 data cdhdef / 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0, 0.e0,
.              0.e0,.05e0,.15e0,.30e0,.55e0,.75e0,.87e0,.95e0,
.              1.e0, 1.e0, 1.e0, 1.e0, 1.e0 /
```

To calculate relative humidity from mixing ratio of water vapor q_v and air temperature T_a, partial pressure of water vapor estimated from q_v and saturated water vapor pressure at the air temperature is calculated by using Tetens (1930)'s equation. Relative humidity is given from the ratio of the partial pressure of water vapor and saturated water vapor pressure.

Then, air pressure, air temperature and mixing ratio of water vapor are defined as $p\ (=\bar{p}+p')$, T_a and q_v, respectively, partial pressure of water vapor is given as follows;

$$e_a = \frac{q_v p}{\epsilon + (1-\epsilon)\, q_v} \tag{9.46}$$

where ϵ is the ratio of molecular mass of water vapor to that of dry air, which is 0.622 when mixing ratio of water vapor is provided with the unit of [kg kg^{-1}], and is 622.0 when it is provided with the unit of [g kg^{-1}]. The unit of air pressure p and partial pressure of water vapor e_a is [Pa].

We assume that air temperature T[°C] (note that the unit is degrees Celsius) is given, saturated water vapor pressure of the air temperature (Tetens (1930)'s equation) becomes

$$e_{as} = e_s \cdot 10^{\frac{a \cdot T}{b+T}} \tag{9.47}$$

or

$$e_{as} = e_s \exp\left(\frac{a \cdot T}{b+T} \ln 10\right) \tag{9.48}$$

where e_s (= 610.78 Pa) is saturated water vapor pressure at 0°C. The constant values a and b are defined as follows;

	a	b
$(T \geq 0°\mathrm{C})$ on the water	7.5	237.3
$(T < 0°\mathrm{C})$ on the ice	9.5	265.3

By using these, relative humidity Rh is gained as follows.

$$Rh = \frac{e_a}{e_{as}} \times 100 \tag{9.49}$$

Cloudiness is derived with this.

By using the method above, cloudiness is decided irrespective of the actual cloud distribution. Because mixing ratios of water substances are calculated in ***CReSS***, cloudiness should be estimated from these values. The total amount of cloud water [m^{-2}] is calculated as follows.

$$q_L = \int_{z_s}^{z_L} \rho \left(q_c + q_r + q_i + q_s + q_g \right) dz \tag{9.50}$$

$$q_M = \int_{z_L}^{z_M} \rho \left(q_c + q_r + q_i + q_s + q_g \right) dz \tag{9.51}$$

$$q_H = \int_{z_M}^{z_t} \rho \left(q_c + q_r + q_i + q_s + q_g \right) dz \tag{9.52}$$

where, z_L, z_M are the heights of interfaces between upper, middle and lower layer, respectively. Then, cloudiness is estimated from the relation between these total amount of cloud water and the solar radiation[4].

9.2.2 Solar Radiation and Downward Long Wave Radiation

Solar radiation (short wave radiation) and downward long wave radiation from atmosphere are calculated as a function of cloudiness, zenith angle of the sun, near–surface temperature, water vapor pressure and albedo at the ground surface.

The estimation of solar radiation and atmospheric long wave radiation are summarized as follows.

- Partial pressure of water vapor e_a is obtained by the temperature and mixing ratio of water vapor q_v at the near surface layer (around the first or second layer).
- The zenith angle of the sun for each grid at the time is calculated.
- Downwarad short wave radiation is estimated by the partial pressure of water vapor e_a, zenith angle and cloud amount with the consideration for albedo.
- Downward long wave radiation is represented as a function of atmospheric temperature T_a, partial pressure of water vapor e_a and cloud amount.

Following subsections explain about net (downward) short wave radiation and downward long wave radiation, respectively.

Partial Pressure of Water Vapor at the Ground Surface

Partial pressure of water vapor at lower layer of atmosphere is estimated utilizing temperature and mixing ratio of water vapor at the lowest atmospheric layer or averaged among some lower atmospheric layers[5].

[4] Kondo (2000)

[5] In JSM, averaged value of first and second atmospheric layers is used

Zenith Angle of the Sun

Zenith angle of the sun is calculated as a function of the number of days from the first day of Janualy $jday$, local time T_{LC} and latitude at the point lat.

Local time is a function of longuitude based on the universal standard time UTC. It is expressed by

$$T_{LC} = UTC + lon/15 \tag{9.53}$$

Angle of the sun ϕ_s is expressed as

$$\phi_s = 23.44\cos(172 - jday) \tag{9.54}$$

then, zenith angle of the sun is expressed as follows (see fig.9.1):

$$\zeta = \sin(lat)\sin(\phi_s) + \cos(lat)\cos(\phi_s)\cos[0.2618(T_{LC} - 12)]. \tag{9.55}$$

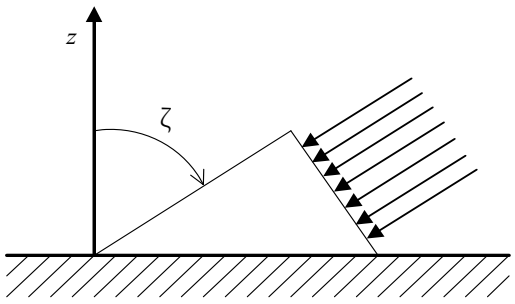

Figure 9.1. Zenith angle of the sunζ.

Net (Downward) Short Wave Radiation

The solar radiation which reaches at the ground surface is called downward short wave radiation. Considering it's reflection from the ground surface with a rate of albedo, downward short wave radiation which is absorbed into the ground is estimated. It is called net downward short wave solar radiation.

In order to estimate total solar radiation which reaches at the ground surface, it is needed to estimate solar radiation which arrive at the top of atmosphere S_∞[6].

$$S_\infty = S_0\cos\zeta \qquad (S_0 = 1,367[\text{W m}^{-2}] : \text{solar constant}) \tag{9.56}$$

[6]in actual, $S_\infty = S_0(\bar{r}_E/r_E)^2\cos\zeta$. r_E is distance between earth and sun, $\bar{r}_E$ is its average

where ζ is zenith angle of the sun. If we now let

$$a = 1.12 - b - 0.06 \log_{10} e_a \qquad (1 \leq e_a \leq 3000[\text{Pa}] : \text{water vapor pressure near surface}) \quad (9.57)$$

$$b = 0.43 + 0.00016 e_a \quad (9.58)$$

then solar radiation of the clear weather is

$$S^{\downarrow} = \begin{cases} S_{\infty}(a + b \times 10^{-0.13 \sec\zeta}) & \cos\zeta > 0 \\ 0 & \cos\zeta \leq 0 \end{cases} \quad (9.59)$$

Adding the effect of the cloud and its albedo, net downward short wave radiation which is absorbed into the ground is

$$RS^{\downarrow} = (1 - A_l)S^{\downarrow}(1 - cd_L \cdot \mathrm{CD}_L)(1 - cd_M \cdot \mathrm{CD}_M)(1 - cd_H \cdot \mathrm{CD}_H) \quad (9.60)$$

where

cd_L	absorbing and refrecting effects with the lower cloud	0.7
cd_M	absorbing and refrecting effects with the middle cloud	0.6
cd_H	absorbing and refrecting effects with the upper cloud	0.3

As mentioned before, a cloud amount is calculated by empirical formula using relative humidity (Ohno and Isa, 1984). Albedo A_l is shown in the following table.

land surface	given from data
sea ice or snow surface	constant to 0.6

At the sea surface, there is no need to consider solar radiation because sea temperature is constant during a calculation, so that there is no need to give the albedo.

Downward Long Wave Radiation from the Atmospere at the Ground Surface

Using empirical formula of Kondo (1976), downward long wave radiation from atmosphere including effects of cloud and water vapor is

$$L^{\downarrow} = \varepsilon_G \sigma T_a^4 [1 + (-0.49 + 0.0066\sqrt{e_a})(1 - \mathrm{CD} \cdot C \cdot A_c)] \quad (9.61)$$

where T_a is estimated by a temperature at the lowest atmospheric layer or averaged for some lower atmospheric layers[7] and

$$C = 0.75 - 0.005e_a \tag{9.62}$$

$$A_c = \frac{\mathrm{CD}_L + 0.85\mathrm{CD}_M + 0.5\mathrm{CD}_H}{\mathrm{CD}} + 0.1\frac{N_r}{N} \tag{9.63}$$

$$\mathrm{CD} = \mathrm{CD}_L + \mathrm{CD}_M + \mathrm{CD}_H \tag{9.64}$$

[8] Then, utilized constants are expressed in the following table.

ε_G	emissivity for infrared region at the ground surface	0.95	
σ	Stefan-Boltzman constant	5.67×10^{-8}	W m^{-2}K^{-4}

9.2.3 Flux at the Ground Surface

Momentum flux at the ground surface τ_x, τ_y[kg m s^{-1} m^{-2} s^{-1} = N m^{-2}] is expressed as follows with bulk coefficient and the absolute value of wind speed.

$$\tau_x = \rho_a C_m |V_a| u_a \tag{9.65}$$
$$\tau_y = \rho_a C_m |V_a| v_a \tag{9.66}$$

Sendible and latent heat flux H_S, ${}_lE$ [J m^{-2} s^{-1} = W m^{-2}] is given as

$$H_S = -\rho_a C_p C_h |V_a|(T_a - T_G) \tag{9.67}$$
$${}_lE = -\rho_a \mathcal{L}_v C_h |V_a| \beta \left[q_{va} - q_{vs}(T_G)\right] \tag{9.68}$$
$$\tag{9.69}$$

where a and G mean the first layer of the atmosphere and the ground surface (first layer of the ground temperature), C_m and C_h are the bulk coefficient (with no dimension) of the momentum, and heat and water vapor, $q_{vs}(T_G)$ is saturated mixed ratio of ground temperature T_G, β is evapotranspiration coefficient and $\mathcal{L}_v$ is latent heat of evaporation of water. They are multiplied atmospheric density and rewrited to equivarent temperature flux F_θ [kg K m^{-2} s^{-1}] and water vapor mixed ratio flux F_{q_v}[kg m^{-2} s^{-1}] as

[7] In JSM, averaged value of first and second atmospheric layers is used

[8] N_r/N means appearance rate of precipitating time. In temporal integration, it is set to 0 in case of no precipitation and to 1 in case of precipitation through the time. In JMS, on the other hand, it is set to 0 constantly though there is no confidence to be finest method.

$$F_\theta = \frac{H_S}{C_p}\left(\frac{p_0}{p}\right)^{\frac{R}{C_p}} = -\rho_a C_h |V_a|(\theta_{va} - \theta_{vG}) = -\rho_a u_* \theta_* \tag{9.70}$$

$$F_{q_v} = E = -\rho_a C_h |V_a| \beta \left[q_{va} - q_{vs}(T_G)\right] = -\rho_a u_* q_{v*} \tag{9.71}$$

where θ_* is scale of potential temperature (frictional temperature) and q_{v*} is scale of mixing ratio (frictional mixing ratio). Friction velocity u_* is defined as

$$u_*^2 = C_m |V_a|^2 \tag{9.72}$$

where $|V_a|$ is absolute value of ground surface wind speed which expressed as

$$|V_a| = (u_a^2 + v_a^2)^{\frac{1}{2}} \tag{9.73}$$

Virtual potential temperature θ_v is

$$\theta_v = \frac{1 + \epsilon_{iv} q_v}{1 + q_v}\theta \tag{9.74}$$

where ϵ_{iv} is ratio of molecular weight of water vapor to that of dry air.

Non-dimensional bulk coefficients C_m, C_h are expressed by the scheme of Louis et al. (1980) considering momentum, heat and roughness parameter of water vapor as follows:

$$C_m = \left[\frac{k}{\ln\left(\frac{z_a}{Z_{0m}}\right)}\right]^2 f_m(Ri, z_a, Z_{0m}) \tag{9.75}$$

$$C_h = \frac{k}{\ln\left(\frac{z_a}{Z_{0m}}\right)} \frac{k}{\ln\left(\frac{z_a}{Z_{0h}}\right)} f_h(Ri, z_a, Z_{0m}, Z_{0h}) \tag{9.76}$$

where k (=0.4), z_a, Z_{0m}, Z_{0h} and Ri are Kalman coefficient, height of the first atmospheric layer, roughness of momentum, roughness of heat and water vapor, and Richardson number, respectively.

In unstable case ($Ri < 0$),

$$f_m = 1 - \frac{2b \cdot Ri}{1 + 3bc\left[\frac{k}{\ln\left(\frac{z_a}{Z_{0m}}\right)}\right]^2 \left(-\frac{Ri \cdot z_a}{Z_{0m}}\right)^{\frac{1}{2}}} \tag{9.77}$$

$$f_h = 1 - \frac{3b \cdot Ri}{1 + 3bc\frac{k}{\ln\left(\frac{z_a}{Z_{0m}}\right)}\frac{k}{\ln\left(\frac{z_a}{Z_{0h}}\right)}\left(-\frac{Ri \cdot z_a}{Z_{0h}}\right)^{\frac{1}{2}}} \tag{9.78}$$

$$b = c = 5 \tag{9.79}$$

In stable case ($Ri > 0$),

$$f_m = \frac{1}{1 + 2b \cdot Ri \cdot \sqrt{1 + d \cdot Ri}} \tag{9.80}$$

$$f_h = \frac{1}{1 + 3b \cdot Ri \cdot \sqrt{1 + d \cdot Ri}} \tag{9.81}$$

$$b = d = 5 \tag{9.82}$$

Roughness parameters of momentum Z_{0m} and of heat and water vapor Z_{0h} are shown in the following table. Roughness of the sea surface is refined at the every calculating step.

roughness	land surface	sea surface	sea ice or snow surface
Z_{0m}	given from data set	calculating as a function of u_*	optional constant value
Z_{0h}	constant value (0.1 m)	same with Z_{0m}	same with Z_{0m}

Standing on the above, actual calculation of land surface flux is shown as follows.

(1) Calculation of Absolute Value of Horizontal Wind Speed

Its absolute value $|V_a|$ is calculated by using horizontal wind speed at the first atmospheric layer (u_a, v_a) (9.73).

(2) Calculataion of Saturated Mixing Ratio of Water Vapor Corresponding to Ground Surface Temperature

Saturated mixing ratio of water vapor $q_{vs}(T_G)$ corresponding to ground temperature at the first ground layer or sea surface temperature is obtained by the equation of Tetens (1930) or (9.48).

From partial pressure of saturated water vapor by the equation of Tetens (1930), saturated mixing ratio of water vapor [kg/kg] is expressed as

$$q_{vs}(T_G) = \epsilon \frac{e_{as}}{p} \tag{9.83}$$

Originally this is the equation to define specific humidity, but we can regard it as mixing ratio with a small error (difference). p is expressed as $p = \bar{p} + p'$, and ϵ is ratio of molecular weight of dry air to water vapor.

(3) Definition of Mixing Ratio of Water Vapor at the Ground Surface

Mixing ratio of water vapor at the ground surface is defined for land, sea, sea ice and snow surfaces, respectively. Considering evapotranspirate efficiency WET which is given to land surface only, it is defined as follows:

for land surface,

$$q_{vsfc} = \beta \left[q_{vs}(T_G) - q_{va} \right] + q_{va} \tag{9.84}$$

for sea, sea ice and snow surfaces,

$$q_{vsfc} = q_{vs}(T_G) \tag{9.85}$$

(4) Calculation of Richardson Number

Richardson number of the ground surface is calculated in order to decide stability at the ground surface and to calculate f_m, f_h in the scheme of Louis et al. (1980).

Richardson number Ri can be expressed as

$$Ri = \frac{g z_a \Delta\theta_v}{\theta_{va} |V_a|^2} \tag{9.86}$$

where ϵ_{iv} indicate ratio of the molecular weight between dry air and water vapor, then

$$\theta_{vG} = T_G \frac{1 + \epsilon_{iv} q_{vsfc}}{1 + \ q_{vsfc}} \left(\frac{p_0}{p} \right)^{\frac{R}{C_p}} \tag{9.87}$$

$$\Delta\theta_v = \theta_{va} - \theta_{vG} \tag{9.88}$$

and z_a is the height of first atmospheric layer and g is the acceleration of gravity.

(5) Calculation of Z_{0m} and u_* at the Sea Surface at the First Step by Iteration

Though both roughnesses at the land surface Z_{0m} and Z_{0h} are constant value during the calculation, those at the sea surface are estimated by the iteration with frictional velocity considering

its dependence on wind speed. The iteration method is used at the only first step and Z_{0m} is estimated by frictional velocity of the one step before; u_*^{n-1} on and after the second step.

Z_{0m} is expressed as a function of u_* based on the formula of Kondo (1975) as follows:

$$Z_{0m} = -34.7 \times 10^{-6} + 8.28 \times 10^{-4} u_* \quad \text{for} \quad u_* \leq 1.08[\text{m s}^{-1}] \tag{9.89}$$

$$Z_{0m} = -0.277 \times 10^{-2} + 3.39 \times 10^{-3} u_* \quad \text{for} \quad u_* > 1.08[\text{m s}^{-1}] \tag{9.90}$$

where maximum of Z_{0m} is 1.5×10^{-5} m. From (9.72), u_* would be

$$u_* = |V_a| \sqrt{C_m} \tag{9.91}$$

Although bulk coefficient C_m is estimated by (9.75), its coefficient f_m is depended on Richardson number which given by (9.86); in unstable case, it is estimated by (9.77) while in stable case, it by (9.80).

Z_{0m} is calculated with 9.89 or 9.90 using the friction velocity u_*. The calculation using the roughness is iterated untill the value of Z_{0m}, u_* converges. Then, we can decide Z_{0m}, u_* at the sea surface.

(6) Calculation of f_m, f_h for the Whole of a Region

In the subsection (5), the roughness parameter of momentum at the sea surface Z_{0m} is estimated. Roughness parameters of heat and water vapor Z_{0h} are defined as $Z_{0h} = Z_{0m}$. At the land surface, the roughness of momentum Z_{0m} is given as a data and the roughnesses of heat and water vapor Z_{0h} are defined as constant value 0.1 m. In case of sea ice and snow cover, roughnesses of momentum, heat and water vapor are defined as constant values.

Roughness parameters of momentum, heat and water vapor for the whole of a region are decided, so coefficients f_m, f_h which are used in the scheme of Louis et al. (1980) and multiply by bulk coefficient are calculated by using (9.77), (9.78), (9.80), and (9.81).

(7) Calculation of Bulk Coefficients C_m, C_h for the Whole of a Region

Using f_m, f_h and roughness parameters $Z_{0m} and Z_{0h}$ which are estimated from (5) and (6), bulk coefficients C_m, C_h are calculated with (9.75) and (9.76).

(8) Calculation of Friction Velocity u_* for the Whole of a Region

According to (9.91), friction velocity u_* is estimated for the whole of a region using bulk coefficients of momentum C_m and $|V_a|$.

(9) Calculation of Correction Term for Potential Temperature and Flux of Mixing Ratio Used in Kondo(1975).

The correction term by Kondo (1975) is estimated to calculate fluxes of water vapor and heat at the sea surface.

Kondo (1975) suggest the following equations of correction factor $B_h^{-1} and B_e^{-1}$ by the observation.

At the sea surface,

$$B_h^{-1} = \frac{1}{k}\ln\left(\frac{\nu + ku_* Z_{0m}}{\mathcal{D}_a}\right) \tag{9.92}$$

$$B_e^{-1} = \frac{1}{k}\ln\left(\frac{\nu + ku_* Z_{0m}}{\mathcal{D}_v}\right) = B_h^{-1} + \frac{1}{k}\ln\left(\frac{\mathcal{D}_a}{\mathcal{D}_v}\right) \tag{9.93}$$

or,

$$B_h^{-1} = \frac{1}{k}\ln\left(0.71 + 4.64 \times 10^4 k u_* Z_{0m}\right) \tag{9.94}$$

$$B_e^{-1} = B_h^{-1} - 0.168\frac{1}{k} \tag{9.95}$$

At the land surface, sea ice and snow cover,

$$B_h^{-1} = 0 \tag{9.96}$$

$$B_e^{-1} = 0 \tag{9.97}$$

where ν is kinetic viscosity coefficient of air [m^2 s^{-1}], $\mathcal{D}_a$ is diffusion coefficient of air [m^2 s^{-1}] and $\mathcal{D}_v$ is diffusion coefficient of water vapor [m^2 s^{-1}][9]. [10]

[9]Kinetic viscosity coefficient of air ν[m^2 s^{-1}] is expressed by,

$$\nu = \nu_0 \frac{101325}{p}\left(\frac{T}{273.16}\right)^{1.754} \tag{9.98}$$

$$\nu_0 = 1.328 \times 10^{-5} \tag{9.99}$$

Diffusion coefficient of air $\mathcal{D}_a$[m^2 s^{-1}] is expressed by,

$$\mathcal{D}_a = \mathcal{D}_{a0} \frac{101325}{p}\left(\frac{T}{273.16}\right)^{1.78} \tag{9.100}$$

$$\mathcal{D}_{a0} = 1.87 \times 10^{-5} \tag{9.101}$$

Diffusion coefficient of water vapor $\mathcal{D}_v$[m^2 s^{-1}] is expressed by,

$$\mathcal{D}_v = \mathcal{D}_{v0} \frac{101325}{p}\left(\frac{T}{273.16}\right)^{1.81} \tag{9.102}$$

$$\mathcal{D}_{v0} = 2.23 \times 10^{-5} \tag{9.103}$$

However, it is enough to use apploximation in the text.

[10]B_h is Stanton number at the bottom layer and B_e is Dalton number at the bottom layer (大気科学講座 1 P95).

(10) Calculation of Flux Coefficients of Potential Temperature and Mixing Ratio of Water Vapor

In order to simplify the estimation of fluxes of potential temperature and mixing ratio of water vapor, the coefficients $E_\theta and E_{q_v}$ are calculated.

$$E_\theta = \frac{\theta_*}{\theta_{va} - \theta_{vG}} = \left[\frac{u_*}{C_h|V_a|} + B_h^{-1}\right]^{-1} \tag{9.104}$$

$$E_{q_v} = \frac{q_{v*}}{q_{va} - q_{vsfc}} = \left[\frac{u_*}{C_h|V_a|} + B_e^{-1}\right]^{-1} \tag{9.105}$$

Though same bulk coefficient C_h has been used in case of water vapor and heat fluxes, the use of above coefficients adopting $B_h^{-1} and B_e^{-1}$ make different coefficient value in case of the sea surface. In cases of the land surface, sea ice and snow cover, however, these values become equally 0.

(11) Calculation of Flux

Standing on the above, fluxes of momentum, sensible heat, latent heat, potential temperature and mixing ratio of water vapor at the ground surface are given as follows.

At first, absolute value of momentum flux is expressed by

$$\tau = \rho_a u_*^2 \tag{9.106}$$

and its x and y components are

$$\tau_x = \tau \frac{u_a}{|V_a|} \tag{9.107}$$

$$\tau_y = \tau \frac{v_a}{|V_a|} \tag{9.108}$$

Secondly, sensible heat and latent heat fluxes are expressed by

$$H_S = -\rho_a C_p u_* E_\theta (T_a - T_G) \tag{9.109}$$

$$_lE = -\rho_a \mathcal{L}_v u_* E_{q_v} (q_{va} - q_{vsfc}) \tag{9.110}$$

and, fluxes of potential temperature F_θ [kg K m^{-2} s^{-1}] and mixing ratio of water vapor F_{q_v} [kg m^{-2} s^{-1}] which are multiplied by atmospheric density are expressed as follows:

$$F_\theta = -\rho_a u_* E_\theta (\theta_{va} - \theta_{vG}) \tag{9.111}$$

$$F_{q_v} = -\rho_a u_* E_{q_v} (q_{va} - q_{vsfc}) \tag{9.112}$$

At land surface, $q_{va} - q_{vsfc} = \beta\,[q_{va} - q_{vs}(T_G)]$ is obtained using (9.84). Then, (9.110) and (9.112) take account of evapolation efficiency β through q_{vsfc}.

The above is the calculation of ground surface flux, however, (9.109) and (9.110) are used in the calculating process of ground temperature actually. (9.111) and (9.112) are used in the calculating process of atmosperic boundary layer. Therefore, this process returns following three coefficients which are common to these equations,

$$C_w = \rho_a u_*^2 \frac{1}{|V_a|} \tag{9.113}$$

$$C_\theta = \rho_a u_* E_\theta \tag{9.114}$$

$$C_{q_v} = \rho_a u_* E_{q_v} \tag{9.115}$$

and mixing ratio at the ground surface q_{vsfc} which include β in case of the land surface. Utilising these coefficients, x, y components of momentum flux and flux of mixing ratio are expressed as follows:

$$\tau_x = C_w u_a \tag{9.116}$$

$$\tau_y = C_w v_a \tag{9.117}$$

$$F_\theta = C_\theta(\theta_{va} - \theta_{vG}) \tag{9.118}$$

$$F_{q_v} = C_{q_v}(q_{va} - q_{vsfc}) \tag{9.119}$$

When we don't use an implicit method in the calculation of atmospheric boundary layer process, we can calculate fluxes of momentum, sensible heat, latent heat, potential temperatre and mixing ratio, instead.

(12) Calculation of Monitor Data at the Ground Surface

Temperature and wind speed at the ground surface are important to be compared with observed physical parameter at the ground surface. Surface boundary layer touching the ground surface is called as constant flux layer where vertical flux is constant regardless of height. According to the nature, temperature and mixing ratio of water vapor at a height of 1.5 m and wind speed at 10 m are calculated as a monitor value.

The method of calculation is simple. u_* is estimated previously and wind speed at a height of 10 m is calculated by (9.72) or (9.91) as

$$|V_{a10}| = \frac{u_*}{\sqrt{C_{m10}}} \tag{9.120}$$

C_{m10} is estimated by (9.75)–(9.81) in wnich $f_m and C_m$ are calculated with application of $z_a = 10$. Since wind direction is constant in vertical within the surface ground layer, x and y components are

$$u_{x10} = |V_{a10}|\frac{u_a}{|V_a|} \tag{9.121}$$

$$v_{y10} = |V_{a10}|\frac{v_a}{|V_a|} \tag{9.122}$$

In case of temperature and mixing ratio of water vapor, $z_a = 1.5$ is used to calculate $f_m and C_m$ by (9.75)–(9.81). Using the obtained bulk coefficient $C_{m1.5}$ and friction velocity u_* at a height of 1.5 m, wind speed at a height of 1.5 m is calculated as follows.

$$|V_{a1.5}| = \frac{u_*}{\sqrt{C_{m1.5}}} \tag{9.123}$$

Simultaneously, bulk coefficient $C_{h1.5}$ of potential temperature and mixing ratio at a height of 1.5m is calculated. Substituting these coefficients into (9.104) and (9.105), we get

$$E_{\theta 1.5} = \left[\frac{u_*}{C_{h1.5}|V_{a1.5}|} + B_h^{-1}\right]^{-1} \tag{9.124}$$

$$E_{q_v 1.5} = \left[\frac{u_*}{C_{h1.5}|V_{a1.5}|} + B_e^{-1}\right]^{-1} \tag{9.125}$$

Virtual potential temperature at a height of 1.5m $\theta_{v1.5}$ is related to that at the surface layer $\theta_{va} and at the ground \theta_{vG}$ as

$$\frac{\theta_{v1.5} - \theta_{vG}}{\theta_{va} - \theta_{vG}} = \frac{E_\theta}{E_{\theta 1.5}} \tag{9.126}$$

Then, vertual potential temperature at height of 1.5m is given by

$$\theta_{v1.5} = \theta_{vG} + (\theta_{va} - \theta_{vG})\frac{E_\theta}{E_{\theta 1.5}} \tag{9.127}$$

Similarly, mixing ratio of water vapor at height of 1.5m $q_{va1.5}$ is also related to that at the surface layer $q_{va} and at the ground q_{vsfc}$ as

$$\frac{q_{va1.5} - q_{vsfc}}{q_{va} - q_{vsfc}} = \frac{E_{q_v}}{E_{q_v 1.5}} \tag{9.128}$$

Then, mixing ratio of water vapor at a height of 1.5m $q_{va1.5}$ is given by

$$q_{va1.5} = q_{vsfc} + (q_{va} - q_{vsfc})\frac{E_{q_v}}{E_{q_v 1.5}} \tag{9.129}$$

where q_{vsfc} includes the effect of evapotranspiration coefficient.

(13) Calculation of the Roughness Parameter at the Sea Surface Z_{0m} for the Next Step

At the end of this calculation process, roughness Z_{0m} which relate to the momentum at the sea surface is calculated for the next step. Adopting the friction velocity u_* into (9.89) and (9.90), Z_{0m} at the sea surface is calculated and conserved untill the next step.

9.2.4 Atmospheric Boundery Layer Process

Momentum, sensible heat and latent heat which are given to the atmosphere from the ground surface are transported upward by the vertical diffusion. Although diffusion is caused by the turbulent flow, we should consider at some points for the diffusion in the boundary layer.

In CReSS, the vertical diffusion other than the usual turbulent flow is taken into consideration in the atmospheric boundary layer process. The height of boundary layer is specified by the user. Since vertical diffusion becomes so small that it is far from the ground surface, the height of which diffusion attains to should just be given. Vertical diffusion of the atmospheric boundary layer process is fundamentally calculated using level 2 of the turbulent closure model by Mellor and Yamada (1974) which is used in JSM (Segami et al. (1989)), and that is also calculation about only 1 dimension in vertical.

The equation of vertical diffusion is expressed with a z coordinate as follows.

$$\frac{\partial u}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_m \frac{\partial u}{\partial z}\right) \tag{9.130}$$

$$\frac{\partial v}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_m \frac{\partial v}{\partial z}\right) \tag{9.131}$$

$$\frac{\partial \theta_v}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_h \frac{\partial \theta_v}{\partial z}\right) \tag{9.132}$$

$$\frac{\partial q_v}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_h \frac{\partial q_v}{\partial z}\right) \tag{9.133}$$

where the turbulent mixing coefficient added $K_0 = 1.0[\mathrm{m}^2 - \mathrm{s}^{-1}]$ was used. That is,

$$K_m = K_0 + l^2 \left|\frac{\partial \mathbf{V}}{\partial z}\right| S_m \tag{9.134}$$

$$K_h = K_0 + l^2 \left|\frac{\partial \mathbf{V}}{\partial z}\right| S_h \tag{9.135}$$

where S_m, S_h are the function of flux Richardson number (Mellor and Yamada, 1974) and length scale l is

$$l = \frac{kz}{1 + \frac{kz}{l_0}} \tag{9.136}$$

k is Karman's constant and l_0 is expressed by making E into turbulent energy as follows.

$$l_0 = 0.10 \times \frac{\int_{z_s}^{\infty} \rho E z dz}{\int_{z_s}^{\infty} \rho E dz} \tag{9.137}$$

Based on the above, the calculation method of boundary layer process is as follows.

(1) Calculation of Air Density between Grids

Air density ρ_k^w is calculated between each vertical grid which is needed in the case of difference specialization (see Fig. 9.2).

ć 2 Ĉ Calculation of the square of vertical shear

Because of the use for following calculation of Richardson number, or of flux, the absolute value of vertical shear is calculated here.

$$\left|\frac{\partial \mathbf{V_k}}{\partial z}\right|^2 = \left(\frac{\partial u_k}{\partial z}\right)^2 + \left(\frac{\partial v_k}{\partial z}\right)^2 \tag{9.138}$$

ć 3 Ĉ Gradient Richardson Number

In the level 2 by Mellor and Yamada (1974), flux Richardson number R_f is given as a function of gradient Richardson number R_i. It is defined as follows.

$$R_i = \frac{g\frac{\partial \theta_v}{\partial z}}{\bar{\theta}_v \left|\frac{\partial \mathbf{V_k}}{\partial z}\right|^2} \tag{9.139}$$

where $\bar{\theta}_v = \frac{1}{2}(\theta_{vk+1} + \theta_{vk})$.

ć 4 Ĉ Flux Richardson Number

Following to Mellor and Yamada (1974), flux Richardson number R_f is given as follows.

$$R_f = 0.725\left(R_i + 0.186 - \sqrt{R_i^2 - 0.316R_i + 0.0346}\right) \tag{9.140}$$

ć 5 Ĉ Calculation of $\tilde{S}_H and \tilde{S}_M$

The constants used by Mellor and Yamada (1974) are as follows.

A_1	A_2	B_1	B_2	C_1
0.78	0.78	15.0	8.0	0.056

Using these values, the following constants are estimated.

$$\gamma_1 \equiv \frac{1}{3} - \frac{2A_1}{B_1} = 0.2293333 \tag{9.141}$$

$$\gamma_2 \equiv \frac{B_2}{B_1} + \frac{6A_1}{B_1} = 0.8453333 \tag{9.142}$$

and,

$$\Gamma \equiv \frac{R_f}{1 - R_f} \tag{9.143}$$

Using these values, $\tilde{S}_H and \tilde{S}_M$ are expressed as follows.

$$\tilde{S}_H = 3A_2(\gamma_1 - \gamma_2\Gamma) \tag{9.144}$$

$$\tilde{S}_M = 3A_1(\gamma_1 - \gamma_2\Gamma)\frac{\gamma_1 - C_1 - (6A_1 + 3A_2)\dfrac{\Gamma}{B_1}}{\gamma_1 - \gamma_2\Gamma + 3A_1\dfrac{\Gamma}{B_1}} \tag{9.145}$$

These can be simplified more in the actual calculation. Eliminating Γ and considering $A_1 = A_2$, they can expresses as

$$\tilde{S}_H = 3A_2\frac{\gamma_1 - (\gamma_1 + \gamma_2)R_f}{1 - R_f} \tag{9.146}$$

$$\tilde{S}_M = \tilde{S}_H\frac{X_1 - X_2R_f}{X_3 - X_4R_f} \tag{9.147}$$

The constants used here are as follows.

$$\gamma_1 + \gamma_2 = 1.074667 \tag{9.148}$$
$$3A_2 = 2.34 \tag{9.149}$$
$$X_1 = 0.173333 \tag{9.150}$$
$$X_2 = 0.641333 \tag{9.151}$$
$$X_3 = 0.229333 \tag{9.152}$$
$$X_4 = 0.918667 \tag{9.153}$$

(6)Calculation of $S_H and S_M$

Using $\tilde{S}_H and \tilde{S}_M$ which are calculated previously, $S_H and S_M$ are estimated. As the factor common to both $S_H and S_M$,

$$S_e = \sqrt{B_1(1 - R_f)\left|\frac{\partial \mathbf{V}}{\partial z}\right|^2 \tilde{S}_M} \tag{9.154}$$

is defined. Then, $S_H and S_M$ are expressed as follows.

$$S_H = S_e \tilde{S}_H \tag{9.155}$$
$$S_M = S_e \tilde{S}_M \tag{9.156}$$

(7)Calculation of Standard of a Length Scale l_0

In Mellor and Yamada (1974), standard of a length scale l_0 is calculated as a function of turbulent kinetic energy (9.137). A turbulent kinetic energy which is estimated in Chapter 7 "diffusion in sub-grid scale" is used.

(8)Calculation of a Length Scale l

Length scale l is given by (9.136) as follows.

$$l = \frac{z}{\dfrac{1}{k} + \dfrac{z}{l_0}} \tag{9.157}$$

(9)Calculation of $K_m and K_h$

Although $K_m and K_h$ are defined as (9.134) and (9.135), the absolute value of vertical shear is already applied to $S_M and S_H$ in the above-mentioned calculation. Then, they are rewritten as follows.

$$K_m = K_0 + l^2 S_M \tag{9.158}$$
$$K_h = K_0 + l^2 S_H \tag{9.159}$$

Here, $K_0 = 1.0\ [\mathrm{m^2\ s^{-1}}]$.

(10)The Finite Difference Method for Equation of Vertical Diffusion

The vertical distributions of $K_m and K_h$ in boundary layer were calculated on the level between grids. Then, the vertical diffusion equations (9.130)–(9.133) are solved and a time changing rate and a correction value (integration value) are calculated in the cases of the explicit method and the inplicit method, respectively.

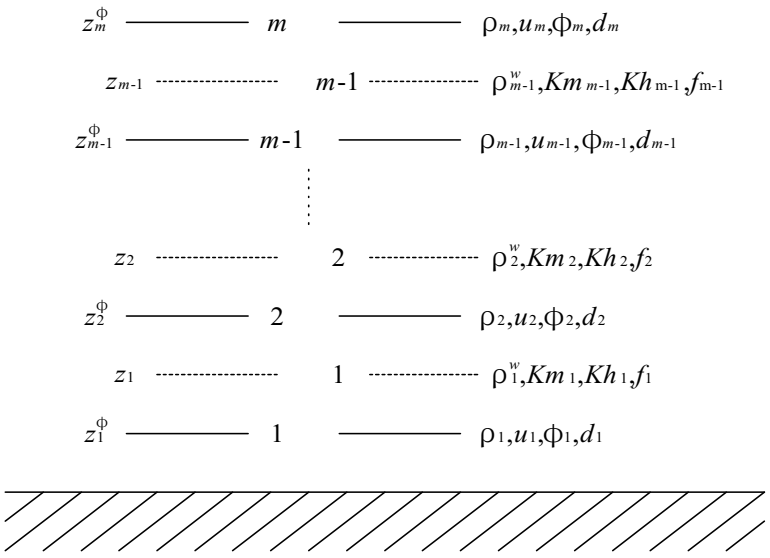

Figure 9.2. vertival grid setting used for atmospheric boundary layer process

We conduct a discretization for these equations with the arrangement as shown in Fig.9.2.

$$\frac{u_k^n - u_k^{n-1}}{\Delta t} = \frac{1}{\rho_k}\frac{1}{z_k - z_{k-1}}\left(\rho_k^w K_{m_k}\frac{u_{k+1}^N - u_k^N}{z_{k+1}^{\phi} - z_k^{\phi}} - \rho_{k-1}^w K_{m_{k-1}}\frac{u_k^N - u_{k-1}^N}{z_k^{\phi} - z_{k-1}^{\phi}}\right) \tag{9.160}$$

$$\frac{\phi_k^n - \phi_k^{n-1}}{\Delta t} = \frac{1}{\rho_k}\frac{1}{z_k - z_{k-1}}\left(\rho_k^w K_{m_k}\frac{\phi_{k+1}^N - \phi_k^N}{z_{k+1}^{\phi} - z_k^{\phi}} - \rho_{k-1}^w K_{m_{k-1}}\frac{\phi_k^N - \phi_{k-1}^N}{z_k^{\phi} - z_{k-1}^{\phi}}\right) \tag{9.161}$$

where u_k^n and ϕ_k^n (subscript with bottom k represents a vertical grid number and that with top n represents the time step) are used as a representative of the velocity component and of the scalar, respectively. N which appears in the right-hand side of difference equation becomes discretization of the explicit method in the case of $N = n-1$, and becomes discretization of the implicit method in the case of $N = n$.

Coefficients of surface fluxes; C_w in (9.113), C_θ in (9.114) and C_{q_v} in (9.115) are used at the bottom layer ($k = 1$).

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} = \frac{1}{\rho_1}\frac{1}{z_1}\left[\rho_1^w K_{m_1}\frac{u_2^N - u_1^N}{z_2^{\phi} - z_1^{\phi}} - C_w(u_1^N - u_{sfc}^N)\right] \tag{9.162}$$

$$\frac{\phi_1^n - \phi_1^{n-1}}{\Delta t} = \frac{1}{\rho_1}\frac{1}{z_1}\left[\rho_1^w K_{m_1}\frac{\phi_2^N - \phi_1^N}{z_2^\phi - z_1^\phi} - C_\phi(\phi_1^N - \phi_{sfc}^N)\right] \tag{9.163}$$

where

$$u_{sfc}^N = 0 \tag{9.164}$$

and representative of the scalar is ϕ. In these equations, $z_0 = 0$ can define the forrowing signs,

$$d_k = \frac{1}{\rho_k}\frac{1}{z_k - z_{k-1}} \tag{9.165}$$

$$f_k = \rho_k^w K_{m_k}\frac{1}{z_{k+1}^\phi - z_k^\phi} \tag{9.166}$$

We can understand K_{h_k} using f_k .
In the case of $k = 1$,

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} = d_1\left[f_1(u_2^N - u_1^N) - C_w u_1^N\right] \tag{9.167}$$

$$\frac{\phi_1^n - \phi_1^{n-1}}{\Delta t} = d_1\left[f_1(\phi_2^N - \phi_1^N) - C_\phi(\phi_1^N - \phi_{sfc}^N)\right] \tag{9.168}$$

In the case of $k \geq 2$,

$$\frac{u_k^n - u_k^{n-1}}{\Delta t} = d_k\left[f_k(u_{k+1}^N - u_k^N) - f_{k-1}(u_k^N - u_{k-1}^N)\right] \tag{9.169}$$

$$\frac{\phi_k^n - \phi_k^{n-1}}{\Delta t} = d_k\left[f_k(\phi_{k+1}^N - \phi_k^N) - f_{k-1}(\phi_k^N - \phi_{k-1}^N)\right] \tag{9.170}$$

About $k = m - 1$,

$$\frac{u_{m-1}^n - u_{m-1}^{n-1}}{\Delta t} = d_{m-1}\left[f_{m-1}(u_m^{n-1} - u_{m-1}^N) - f_{m-2}(u_{m-1}^N - u_{m-2}^N)\right] \tag{9.171}$$

$$\frac{\phi_{m-1}^n - \phi_{m-1}^{n-1}}{\Delta t} = d_{m-1}\left[f_{m-1}(\phi_m^{n-1} - \phi_{m-1}^N) - f_{m-2}(\phi_{m-1}^N - \phi_{m-2}^N)\right] \tag{9.172}$$

where since u_m^{n-1} and ϕ_m^{n-1} in the case of $k = m - 1$ are given as boundary condition, this process is eternal.

In the case of $N = n - 1$, these equations are solved by the explicit method and time change rate is acquired. In the case of $N = n$, they are solved by the implicit method and integration value is acquired.

9.2.5 Ground Temperature

The ground temperature is calculated by the model with m layers.

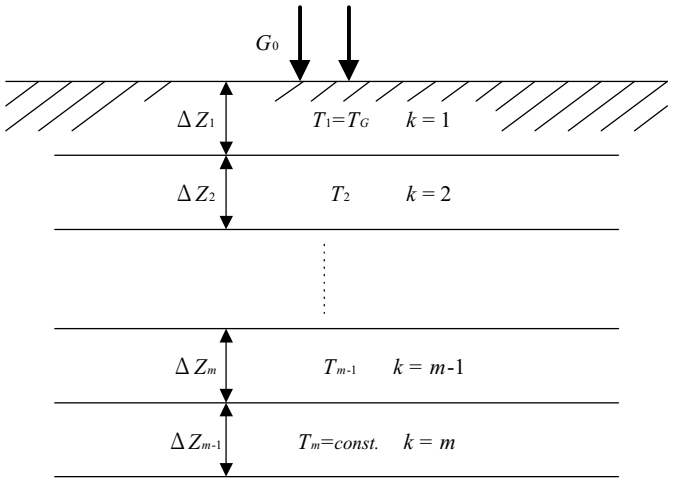

Figure 9.3. A underground setup used for calculation of ground temperature.

Figure 9.3 shows a ground grid setting used for calculation of ground temperature. In the calculation, only the vertical heat conduction is taken into consideration, while the horizontal heat diffusion does not take into consideration.

Here, time evolution equation system of the ground temperature in the model with m layer is expressed as follows.

$$
\begin{aligned}
\frac{\partial T_1}{\partial t} &= \frac{G_0}{\rho_G C_G \Delta Z_1} + \frac{2\nu_G}{\Delta Z_1(\Delta Z_2 + \Delta Z_1)}(T_2 - T_1) \\
\frac{\partial T_2}{\partial t} &= -\frac{2\nu_G}{\Delta Z_2(\Delta Z_2 + \Delta Z_1)}(T_2 - T_1) + \frac{2\nu_G}{\Delta Z_2(\Delta Z_3 + \Delta Z_2)}(T_3 - T_2) \\
&\cdots\cdots \\
\frac{\partial T_k}{\partial t} &= -\frac{2\nu_G(T_k - T_{k-1})}{\Delta Z_k(\Delta Z_k + \Delta Z_{k-1})} + \frac{2\nu_G(T_{k+1} - T_k)}{\Delta Z_k(\Delta Z_{k+1} + \Delta Z_k)} \\
&\cdots\cdots \\
\frac{\partial T_{m-1}}{\partial t} &= -\frac{2\nu_G(T_{m-1} - T_{m-2})}{\Delta Z_{m-1}(\Delta Z_{m-1} + \Delta Z_{m-2})} + \frac{2\nu_G(T_m - T_{m-1})}{\Delta Z_{m-1}(\Delta Z_m + \Delta Z_{m-1})}
\end{aligned}
\tag{9.173}
$$

In the m layer, temperature T_m is constant during a calculation period. Heat capacitiy of the ground per unit volume $\rho_G C_G$ and heat diffusion coefficient ν_G are

$$\rho_G C_G = 2.3 \times 10^6 \qquad [\mathrm{Jm^{-3}K^{-1}}] \tag{9.174}$$

$$\nu_G = 7.0 \times 10^{-7} \qquad [\mathrm{m^2 s^{-1}}] \tag{9.175}$$

In (9.173) , G_0 which is appeared in the first ground layer $(k = 1)$ is heat flux which goes to the ground. It is sum of net radiation R_{net}, sensible heat H_S and latent heat ${}_lE$.

$$G_0 = R_{net} - H_S - {}_lE \tag{9.176}$$

They are given as follows, respectively.

$$R_{net} = RS^{\downarrow} + L^{\downarrow} - L^{\uparrow} \tag{9.177}$$

$$H_S = -C_p C_\theta (T_a - T_G) \tag{9.178}$$

$${}_lE = -\mathcal{L}_v C_{q_v} \beta \left[q_{va} - q_{vs}(T_G)\right] \tag{9.179}$$

where the last term of net radiation R_{net} shows net upward longwave radiation $T_1 = T_G$ and,

$$L^{\uparrow} = \varepsilon_G \sigma T_G^4 \tag{9.180}$$

β in the equation of latent heat ${}_lE$ is evapotranspiration coefficient which is supposed to take constant value during the integration period. Then, radiation process is used only for calculation of the heat balance at the ground surface.

Time integration of the ground temperature solved by implicit scheme. In order to disperse the time evolution equation of ground temperature (9.173),

about $k = 1 \sim m - 1$, the followings are defined.

$$a_1 = 0 \tag{9.181}$$

$$a_k = -\Delta t \frac{2\nu_G}{\Delta z_k(\Delta z_k + \Delta z_{k-1})} \tag{9.182}$$

$$b_k = -\Delta t \frac{2\nu_G}{\Delta z_k(\Delta z_{k+1} + \Delta z_k)} \tag{9.183}$$

Then, (9.173) is dispersed as follows.

$$\begin{aligned}
T_1^n &= T_1^{n-1} + \frac{\Delta t G_0}{\rho_G C_G \Delta Z_1} + b_1(T_1^n - T_2^n) \\
T_2^n &= T_2^{n-1} + a_2(T_2^n - T_1^n) + b_2(T_2^n - T_3^n) \\
&\cdots\cdots \\
T_k^n &= T_k^{n-1} + a_k(T_k^n - T_{k-1}^n) + b_k(T_k^n - T_{k+1}^n) \\
&\cdots\cdots \\
T_{m-1}^n &= T_{m-1}^{n-1} + a_{m-1}(T_{m-1}^n - T_{m-2}^n) + b_{m-1}(T_{m-1}^n - T_m)
\end{aligned} \tag{9.184}$$

The simultaneous equations can be expressed using a procession as follows.

$$\begin{pmatrix} T_1^n \\ T_2^n \\ \vdots \\ T_k^n \\ \vdots \\ T_{m-1}^n \end{pmatrix} = \begin{pmatrix} T_1^{n-1} + \dfrac{\Delta t G_0}{\rho_G C_G \Delta Z_1} \\ T_2^{n-1} \\ \vdots \\ T_k^{n-1} \\ \vdots \\ T_{m-1}^{n-1} - b_{m-1} T_m \end{pmatrix} \tag{9.185}$$

$$+\begin{pmatrix} b_1 & -b_1 & 0 & \cdots & \cdots & \cdots & 0 \\ -a_2 & a_2+b_2 & -b_2 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -a_k & a_k+b_k & -b_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & -a_{m-1} & a_{m-1}+b_{m-1} \end{pmatrix} \begin{pmatrix} T_1^n \\ T_2^n \\ \vdots \\ T_k^n \\ \vdots \\ T_{m-1}^n \end{pmatrix}$$

Furthermore, it can be expressed briefly like

$$\mathbf{T}^n = \mathbf{F} + A\mathbf{T}^n \tag{9.186}$$

If the term of $\mathbf{T}^n$ is shifted using unit procession I of $(m-1) \times (m-1)$,

$$(I - A)\mathbf{T}^n = \mathbf{F} \tag{9.187}$$

is obtained. While it can be solved easily by the elimination of a gauss (refer to the 10.1.3 for explanation of a concrete solution method), ground temperature of all layers at the time of n are calculated.

Chapter 10

Numerical Computation Method

Generally, the equation of motion, the equation of potential temperature derived by the equation of thrmodynamics, the equation of pressure derived by the continity eqution and the eqution of gaseous state, and the equations of water vapor and hydrometeors are solved simultaneously and integrated with respect to time under appropriate boundary and initial values in cloud resolving models.

In ***CReSS***, the equation of motion, the equations of the potential temperature perturbation and the pressure perturbation, and the equations of hydrometeors and water vapor which are shown in Chapter 6 'Formulation of the system of basic equations' are also temporally and spatially integrated by using finite difference method. There are many studies about the methods which are discretizated and integrated these equation systems and the various calculation techniques are established.

In this chapter, the structure of a grid point, the method of discretization, the method of time integration and so on are explained about the numerical computation method used by ***CReSS***.

10.1 The discretization of basic equation system

10.1.1 The outline of the numerical solution

There are various techniques from the selection of dependent variables to the method of time integration in the numerical solution by cloud resolving models. The outline of the numerical solution used in ***CReSS*** is as follows.

- The basic equation system is the equation of motion, potential temperature perturbation, pressure perturbaton, water vapor and hydrometeors.
- The dependent variables are three-dimensionl velocity components u, v, w, potential temperature perturbation θ', pressure perturbation p', mixing ratio of watar vapor, and mixing ratios and number concentrations of hydrometeors.
- The basic equations are defined by a terrain-following curvilinear coordinates are used to include the effect of orography.
- The finite differences are used for the derivation of the dependent variable with respect to space. The types of Arakawa-C and Lorenz staggered grids are used for hrizontal and vertical grid arrangement, respectively.
- The sound wave is contained as a solution by using compressible equation system. The terms related to the sound wave are integrated with small time step $\Delta\tau$, the other terms are integrated with large time step Δt.
- The leap-frog scheme with the Asselin time filter is used for time integration with respect to the large time step.
- The explicited forward-backward scheme or the inplicited Crank-Nicolson scheme in only vertical is used for time integration with respect to the small time step.
- The effects of sound wave are weakened by adding the divergence terms explicit scheme with the small time step.

Furthermore, ξ, η are expressed as x, y in this chapter as it is $\xi = x, \eta = y$ in a terrain-following coordinates.

10.1.2 Grid in the model and setting of variables

As the space are represented by the method of grid points in ***CReSS***, finite difference method is used for the spatial derivation. Here, the strucuture of grid points and the setting of variables in ***CReSS*** are shown.

The strucuture of grid points and the setting of variables are set on the staggered grid [1] as Fig10.1 in horizontal and vertical. Arakawa-C grid and Lorenz grid are used for the settings of horizontal and vertical grid points, respectively.

[1] There are various setting in the staggered grid which variables are defined alternately.

In this case, the point of the all scalars, such as pressure perturbations, potential temperature perturbation, mixing ratios of watar vapor and physical quantities of hydrometeors, are located at the center of the grid box[2].

The point of x-component velocity u is dislocated to a half interval from the center of the grid in the x-direction, the point of y-component velocity v is dislocated to a half interval from the center of the grid in the y-direction in horizontal. The point of u and v are defined at the same level as the scalars points. The point of z-component velocity w is vertically dislocated at a half interval just above and below the scalar variables points.

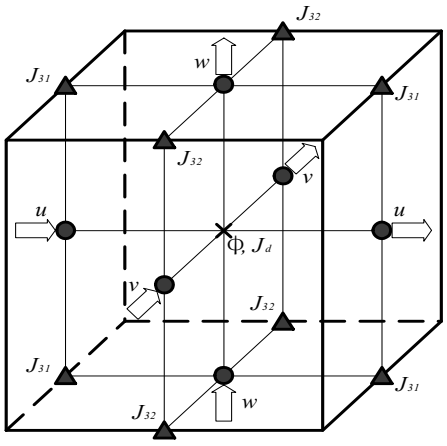

Figure 10.1. Structure of the staggered grid and setting of dependent variables.

Furthermore, as a terrain-following coordinate is used in ***CReSS***, the Jacobian with transform for coordinates $G^{\frac{1}{2}}$ and the metric tensor which is not zero or 1, that is, $J_{31}, J_{32}, J_d\ (= G^{\frac{1}{2}})$ are shown as follows. The $G^{\frac{1}{2}}\ (= J_d)$ point is defined at the same level as the scalars points. The J_{31} point is dislocated to a half interval from the center of the grid in the x-direction and the z-direction. The J_{32} is dislocated to a half interval from the center of the grid in the y-direction and the z-direction.

The coordinates x, y, ζ which are set on the staggered grid as shown in Fig 10.1 are difined as the vectors points. The domain in a thick line except the most outside grid points is the physical domain as shown in Fig10.2. So four grid points in the direction without a dimension need to be calculated at the time of the 2-dimensional caluculation.

10.1.3 Discretization by finite difference method

Generally, since elastic equations without any approximations are used in the cloud resolving model as the basic equations, all waves in the atmosphere are represented in the model. Waves in the atmosphere are divided into the sound wave, the gravity wave and the Rossby wave by phase velocity or frequency, and the phase velocities become small in this order. The phase velocity of sound wave is the fastest but we have no meteorological interest for it. On the other hand, the gravity wave and the Rossby wave are important for a cloud and a mesocale meteorological

[2]The point of potential temperature (temperature) and the point of pressure are located at the same level in Lorenz grid. On the other hand, the point of potential temperature (temperature) is dislocated at a half interval below the pressure point, and defined at the same level as the vertical velocity point in the Charney Phillips' grid.

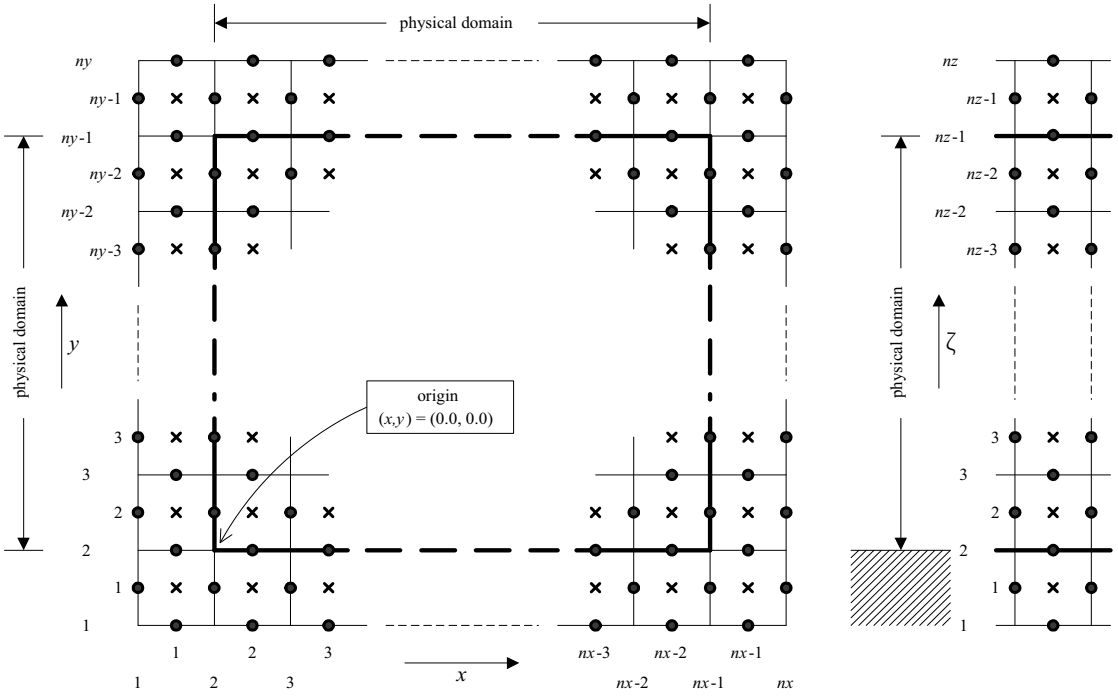

Figure 10.2. Setting of variables in a calculation domain (● represents a point for velocities. Ĝ represents a point for scalars).

phenomena[3] However, the sound wave is contained in models with elastic equation system, and the special technique is needed to be used for the calculation about the wave. So the time step is very small to satisfy CFL condition for the sound wave which the phase velocity is fastest if the time integration is calculated explicitly in meteorological models the generall numerical integration for fluid. However, the time steps for the gravity and the Rossby waves which are meteorologically important are satisfied with the much larger time step than that of the sound wave, since the phase velocities of these waves are much slower than the sound wave.

There are various calculation techniques not to make the time step of the sound wave small[4]. The well-known calculation techniques are as follows.

HI-VI method The terms of the sound wave are solved inplicitly, and the other terms are solved explicitly. This technique is called semi-inplicit time integration method[5].

[3]The Rossby wave is important for larger scale phenomena in mesocale meteorology, synoptic meteorological phenomena and planetary meteorological phenomena. Weather is influenced by the Rossby-scale atmospheric phenomena every day, and usual forecast models represent these phenomena.

[4]If the anelastic equation system by Ogura and Phillips (1962) is used for the basic equation system in a numerical model, the time step can be defined by the phase velocity of gravity wave because the sound wave is not contained in this equation system. In this case, pressure is calculated by the equation of motion and the continity eqution. There are the advantage that the sound wave is removed, but the demerits of which the equation of pressure is complicated and a error is large at the place with large variations of density in this techniques.

[5]The terms with large time step are explicitly calculated and the other terms are inplicitly solved in the semi-inplicit time integration method. For example, the semi-inplicit time integration method of which the terms for the gravity wave are explicitly solved, and the other terms are solved by using the leap-frog scheme is used in the model of JMA.

HE-VE method The terms of the sound wave and the other terms are solved explicitly using different time steps, respectively. This technique which is called the mode-splitting technique (Klemp and Wilhelmson, 1978) is often used.

HE-VI method This technique is almost the same as the mode-splitting technique. But the terms of the sound wave are solved inplicitly in vertical.

The schematic representation of this mode-splitting technique is shown in Fig10.3.

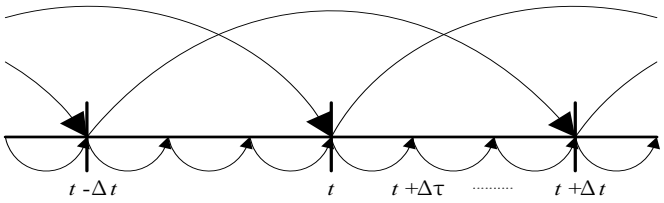

Figure 10.3. The schematic representation of the mode-splitting technique.

Many data are exchanged between some nodes in the case of using parallel computer since large simultaneous equations are solved in the HI-VI method. The HE-VE method or HE-VI method are used in ***CReSS*** since it is assumed to be mainly used on a parallel computer. These techniques are easy to be applied to the large parallel computing since the phisycal space are corrsponded to the setting of a node and only the data in halo region of a node communicates between nodes. The detail description is shown in 11 'Mounting of a parallel processing'. The efficiency of calculation does not become so bad to integrate the terms with respect to sound wave in small time step.

The method of discretization for physical quanities with the sound wave mode

The method of discretization for physical quantities with the sound wave mode is shown from here. The gravity wave mode is calculated with the large time step. In this case, the discretizations for potential temperature, mixing ratios of watar vapor, mixing ratios and number concentrations of hydrometeors is not took into consideration as they are calculated in the large time step.

Physical quantities with the sound wave mode are shown in the basic equation system in ***CReSS***. In the following equations, $[rm]$ is indicates terms which are related to the Rossby wave mode (rotational mode), $[gm]$ the gravity wave mode (divergence mode) and $[am]$ the sound wave mode, respectively. Terms of physical processes are indicated by $[physics]$.

The equation of motion

$$\frac{\partial u^*}{\partial t} = \underbrace{-\left(u^*\frac{\partial u}{\partial x} + v^*\frac{\partial u}{\partial y} + W^*\frac{\partial u}{\partial \zeta}\right)}_{[rm]} \underbrace{-\left[\frac{\partial}{\partial x}\left\{J_d\left(p' - \alpha Div^*\right)\right\} + \frac{\partial}{\partial \zeta}\left\{J_{31}\left(p' - \alpha Div^*\right)\right\}\right]}_{[am]} + \underbrace{\left(f_s v^* - f_c w^*\right)}_{[rm]} + \underbrace{G^{\frac{1}{2}}\mathrm{Turb}.u}_{[physics]} \quad (10.1)$$

$$\frac{\partial v^*}{\partial t} = \underbrace{-\left(u^*\frac{\partial v}{\partial x} + v^*\frac{\partial v}{\partial y} + W^*\frac{\partial v}{\partial \zeta}\right)}_{[rm]} \underbrace{-\left[\frac{\partial}{\partial y}\left\{J_d\left(p' - \alpha Div^*\right)\right\} + \frac{\partial}{\partial \zeta}\left\{J_{32}\left(p' - \alpha Div^*\right)\right\}\right]}_{[am]} - \underbrace{f_s u^*}_{[rm]} + \underbrace{G^{\frac{1}{2}}\mathrm{Turb}.v}_{[physics]} \quad (10.2)$$

$$\frac{\partial w^*}{\partial t} = \underbrace{-\left(u^*\frac{\partial w}{\partial x} + v^*\frac{\partial w}{\partial y} + W^*\frac{\partial w}{\partial \zeta}\right)}_{[rm]} \underbrace{-\frac{\partial}{\partial \zeta}\left(p' - \alpha Div^*\right)}_{[am]} - \rho^* g\left(\underbrace{\frac{\theta'}{\bar{\theta}}}_{[gm]} - \underbrace{\frac{p'}{\bar{\rho}c_s^2}}_{[am]} + \underbrace{\frac{q_v' - \sum q_x}{\epsilon + \bar{q}_v} - \frac{q_v' + \sum q_x}{1 + \bar{q}_v}}_{[physics]}\right) + \underbrace{f_c u^*}_{[rm]} + \underbrace{G^{\frac{1}{2}}\mathrm{Turb}.w}_{[physics]}$$

The equation of pressure

$$\frac{\partial G^{\frac{1}{2}}p'}{\partial t} = \underbrace{-\left(G^{\frac{1}{2}}u\frac{\partial p'}{\partial x} + G^{\frac{1}{2}}v\frac{\partial p'}{\partial y} + G^{\frac{1}{2}}W\frac{\partial p'}{\partial \zeta}\right)}_{[rm]} + \underbrace{G^{\frac{1}{2}}\bar{\rho}gw}_{[am]} \underbrace{-\bar{\rho}c_s^2\left(\frac{\partial G^{\frac{1}{2}}u}{\partial x} + \frac{\partial G^{\frac{1}{2}}v}{\partial y} + \frac{\partial G^{\frac{1}{2}}W}{\partial \zeta}\right)}_{[am]} + \underbrace{G^{\frac{1}{2}}\bar{\rho}c_s^2\left(\frac{1}{\theta}\frac{d\theta}{dt} - \frac{1}{Q}\frac{dQ}{dt}\right)}_{[am]} \quad (10.4)$$

Here, the density $\bar{\rho}$ and the Jacobian $G^{\frac{1}{2}}$ for coordinate transformation are defined at the same point as the scalars. The vertical velocity W in a terrain-following coordinates and the vertical velocity w in the Cartesian coordinates are defined at the same point.

$$W = \frac{\left(\overline{\overline{u}^{\zeta} J_{31}}^{x} + \overline{\overline{v}^{\zeta} J_{32}}^{y} + w\right)}{\overline{G^{\frac{1}{2}}}^{\zeta}} \tag{10.15}$$

In the case of explicit scheme, descretization of the equations of (10.1) $\sim$ (**??**) are given with respect to u, v, w, p' as follows.

$$\overline{\rho^*}^{x} \frac{u^{\tau+\Delta\tau} - u^{\tau}}{\Delta\tau} = -\left[\partial_x \left(J_d P_\alpha\right) + \partial_\zeta \left(J_{31} \overline{P_\alpha}^{\zeta x}\right)\right]^{\tau} + F_u^t \tag{10.16}$$

$$\overline{\rho^*}^{y} \frac{v^{\tau+\Delta\tau} - v^{\tau}}{\Delta\tau} = -\left[\partial_y \left(J_d P_\alpha\right) + \partial_\zeta \left(J_{32} \overline{P_\alpha}^{\zeta y}\right)\right]^{\tau} + F_v^t \tag{10.17}$$

$$\overline{\rho^*}^{\zeta} \frac{w^{\tau+\Delta\tau} - w^{\tau}}{\Delta\tau} = -\left[\partial_\zeta P_\alpha\right]^{\tau} - \left[\overline{g G^{\frac{1}{2}} \frac{p'}{c_s^2}}^{\zeta}\right]^{\tau} + F_w^t \tag{10.18}$$

$$\begin{aligned} \overline{G^{\frac{1}{2}}}^{\zeta} \frac{p'^{\tau+\Delta\tau} - p'^{\tau}}{\Delta\tau} = & -\bar{\rho} c_s^2 \left[\partial_x \left(\overline{G^{\frac{1}{2}}}^{x} u\right) + \partial_\zeta \left(\overline{J_{31} \overline{u}^{\zeta}}^{x}\right) + \partial_y \left(\overline{G^{\frac{1}{2}}}^{y} v\right) + \partial_\zeta \left(\overline{J_{32} \overline{v}^{\zeta}}^{y}\right) + \partial_\zeta w\right]^{\tau+\Delta\tau} \\ & + g\rho^* \left[\overline{w}^{\zeta}\right]^{\tau+\Delta\tau} + F_p^t \end{aligned} \tag{10.19}$$

where

$$P_\alpha = p' - \alpha Div^* \tag{10.20}$$

$$Div^* = \frac{1}{G^{\frac{1}{2}}} \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} + \frac{\partial W^*}{\partial \zeta}\right) \tag{10.21}$$

where α is a factor of damping for divergence terms to weaken the effects of the sound wave. The terms concerning the sound wave, which are integrated with a small time step $\Delta\tau$, are explicitly written, the other terms are included in F_ϕ^t ($\phi = u, v, w, p'$). The first term on the right side in the equations of (10.16) and (10.17), and the second term on the right side in the equation of (10.18) are pressure gradient force term. Buoyancy force with pressure perturbations and divergence damping are shown by the first and third term on the right side in the equation (10.18), respectively. The first term on the right side in the equation of (10.19) is a divergence term and the second term of that is a vertical advection term.

When terms for sound wave are solved explicitly, u, v, and w are calculated by the equations of (10.16) $\sim$ (10.18) with forward-difference for $\Delta\tau$. Using the values of u, v, and w, pressure perturbation p' is calculated by the equation of (10.19) with backward-difference for $\Delta\tau$. During this integration with a small time step for $2\Delta\tau$ $(= n\Delta\tau)$, F_ϕ^t is constant. Terms of the sound wave in most of non-hydrostatic models are not solved explicitly in vertical and horizontal. Although some models (e.g. ARPS) include the HE-VE scheme as an option, the NHM of Meteorological Research Institute adopts the HI-VI and HE-VI schemes. The technique to solve inplicitly in only vertical is described later. However, it is not necessarily that terms for sound wave are calculated inplicitly in vertical when the horizontal resolution is the same as the verical resolution as an object of ***CReSS***. There is a option to calculate the gravity mode in the small time step in ***CReSS***. This technique is also shown later. The gravity wave mode is integrated in the small time step in ARPS and in the large time step in NHM.

In the case of using vertical implicit scheme, the equations of verical velocity w and pressure perturbation p' in the equations (10.18 and 10.19) are replaced by the following equations.

$$\begin{aligned}\overline{\rho^*}^\zeta \frac{w^{\tau+\Delta\tau}-w^\tau}{\Delta\tau} &= [\partial_\zeta(\alpha Div^*)]^\tau - \left[\beta\left(\partial_\zeta p'\right)^{\tau+\Delta\tau} + (1-\beta)\left(\partial_\zeta p'\right)^\tau\right] \\ &\quad - \left[\beta\left(\overline{gG^{\frac{1}{2}}\frac{p'}{c_s^2}}^\zeta\right)^{\tau+\Delta\tau} + (1-\beta)\left(\overline{gG^{\frac{1}{2}}\frac{p'}{c_s^2}}^\zeta\right)^\tau\right] + F_w^t \qquad (10.22)\end{aligned}$$

$$\begin{aligned}\overline{G^{\frac{1}{2}}}^\zeta \frac{p'^{\tau+\Delta\tau}-p'^\tau}{\Delta\tau} &= -\bar{\rho}c_s^2\left[\partial_x\left(\overline{G^{\frac{1}{2}}}^x u\right) + \partial_\zeta\left(\overline{J_{31}\overline{u}^\zeta}^x\right) + \partial_y\left(\overline{G^{\frac{1}{2}}}^y v\right) + \partial_\zeta\left(\overline{J_{32}\overline{v}^\zeta}^y\right)\right]^{\tau+\Delta\tau} \\ &\quad -\bar{\rho}c_s^2\left[\beta\left(\partial_\zeta w\right)^{\tau+\Delta\tau} + (1-\beta)\left(\partial_\zeta w\right)^\tau\right] \\ &\quad +g\rho^*\left[\beta\left(\overline{w}^\zeta\right)^{\tau+\Delta\tau} + (1-\beta)\left(\overline{w}^\zeta\right)^\tau\right] + F_p^t \qquad (10.23)\end{aligned}$$

These equations are gained by averaging for τ and $\tau+\Delta\tau$ of the equations (10.18) and (10.19) weighted by β. The equations of w (10.22) with $\beta = 0$ and p' (10.23) with $\beta = 1$ result in the equations of (10.18) and (10.19), respectively. When $\beta = 0$, (10.22) and (10.23) are teh simultaneous equations for $w^{\tau+\Delta\tau}$ and $p^{\tau+\Delta\tau}$. They are implicitly calculated in vertical wtih the Crank-Nicolson scheme.

In the vertical implicit scheme, the forward difference for the equations of (10.16) and (10.17) are calculated and the values of u and v at $\tau+\Delta\tau$ are gained. In the vertical implicit scheme, the forward difference for the equations of (10.16) and (10.17) are calculated and the values of u and v at $\tau+\Delta\tau$ are gained. Concerning $p'^{\tau+\Delta\tau}$ from (10.22) and (10.23), we get the following equations for w and p.

$$\begin{aligned}\overline{\rho^*}^\zeta \frac{w^{\tau+\Delta\tau}-w^\tau}{\Delta\tau} &= -\Delta\tau\beta^2\partial_\zeta\left(\frac{g\rho^*}{G^{\frac{1}{2}}}\overline{w}^\zeta - \frac{\bar{\rho}c_s^2}{G^{\frac{1}{2}}}\partial_\zeta w\right)^{\tau+\Delta\tau} \\ &\quad -\Delta\tau\beta^2 g\left(\overline{\frac{g\rho^*}{c_s^2}\overline{w}^\zeta}^\zeta - \overline{\bar{\rho}\partial_\zeta w}^\zeta\right)^{\tau+\Delta\tau} + F_w' \qquad (10.24)\end{aligned}$$

where F'_w and F'_p are

$$\begin{aligned} F'_p = & \frac{\Delta\tau}{G^{\frac{1}{2}}}\left[F^t_p + (1-\beta)\left(g\rho^*\overline{w}^\zeta - \bar{\rho}c_s^2\partial_\zeta w\right)^\tau\right] \\ & -\frac{\Delta\tau}{G^{\frac{1}{2}}}\bar{\rho}c_s^2\left[\partial_x\left(\overline{G^{\frac{1}{2}}}^x u\right) + \partial_\zeta\left(\overline{J_{31}\overline{u}^\zeta}^x\right) + \partial_y\left(\overline{G^{\frac{1}{2}}}^y v\right) + \partial_\zeta\left(\overline{J_{32}\overline{v}^\zeta}^y\right)\right]^{\tau+\Delta\tau} \end{aligned} \tag{10.25}$$

$$F'_w = -\left[\partial_\zeta P_\alpha\right]^\tau - \left[\overline{gG^{\frac{1}{2}}\frac{p'}{c_s^2}}^\zeta\right]^\tau - \beta\left[\overline{\frac{gG^{\frac{1}{2}}F'_p}{c_s^2}}^\zeta + \partial_\zeta F'_p\right] + F^t_w. \tag{10.26}$$

The second-order discretization in vertical (10.24) is ordered as follows for the unknown $w^{\tau+\Delta\tau}$.

$$A_k w_{k-1}^{\tau+\Delta\tau} + B_k w_k^{\tau+\Delta\tau} + C_k w_{k+1}^{\tau+\Delta\tau} = F_k \tag{10.27}$$

Simultaneous equations whose factors are a triple diagonal matrix are obtained as this equation. This equation is solved simply if the top and bottom boundary conditions are given as follows, for example.

$$w = 0 \qquad k = nk - 1 \tag{10.28}$$

$$w = u\,\frac{\partial z_s}{\partial x} + v\frac{\partial z_s}{\partial y} \qquad k = 2 \tag{10.29}$$

where A_k, B_k, C_k, F_k are known quantities at τ. These quantities are given under the boundary conditions of (10.28) and (10.29) as follows.

$$A_k = \begin{cases} 0 & k = 3 \\ \left(-\mathcal{Q}_k c_{sk-1}^2 + \mathcal{P}_k G_{k-1}^{\frac{1}{2}}\right)\left(\mathcal{R}_{k-1} + \mathcal{S}_{k-1}\right), & 4 \le k \le nk-2 \end{cases} \tag{10.30}$$

$$\begin{aligned} B_k = & 1 + \mathcal{Q}_k\left\{\left(\mathcal{R}_k + \mathcal{S}_k\right)c_{sk}^2 - \left(\mathcal{R}_{k-1} - \mathcal{S}_{k-1}\right)c_{sk-1}^2\right\} \\ & +\mathcal{P}_k\left\{\left(\mathcal{R}_k + \mathcal{S}_k\right)G_k^{\frac{1}{2}} - \left(\mathcal{R}_{k-1} - \mathcal{S}_{k-1}\right)G_{k-1}^{\frac{1}{2}}\right\}, \quad 3 \le k \le nk-2 \end{aligned} \tag{10.31}$$

$$C_k = \begin{cases} \left(\mathcal{Q}_k c_{sk}^2 + \mathcal{P}_k G_k^{\frac{1}{2}}\right)\left(\mathcal{R}_k + \mathcal{S}_k\right), & 3 \le k \le nk-3 \\ 0, & k = nk-2 \end{cases} \tag{10.32}$$

$$F_k = \begin{cases} F'_{wk} + w_k^\tau + \overline{u}_k^\zeta \left(\partial_x z_s\right)_k + \overline{v}_k^\zeta \left(\partial_y z_s\right)_k, & k = 3 \\ F'_{wk} + w_k^\tau, & 4 \le k \le nk-2 \end{cases} \tag{10.33}$$

$$\mathcal{P}_k = \frac{\Delta\tau^2\beta^2 g}{2\overline{\rho_k^*}^\zeta}, \quad \mathcal{Q}_k = \frac{\Delta\tau^2\beta^2}{\Delta\zeta_k \overline{\rho_k^*}^\zeta}, \quad \mathcal{R}_k = \frac{g\rho_k^*}{2G_k^{\frac{1}{2}} c_{sk}^2}, \quad \mathcal{S}_k = \frac{\rho_k^*}{\Delta\zeta_k G_k} \tag{10.34}$$

This simultaneous equation (10.27) is represented by a matrix by replacing $w_k^{\tau+\Delta\tau}$ with w_k as follows.

$$\begin{pmatrix} B_3 & C_3 & 0 & \cdots & \cdots & \cdots & 0 \\ A_4 & B_4 & C_4 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & A_k & B_k & C_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & A_{nk-2} & B_{nk-2} \end{pmatrix} \begin{pmatrix} w_3 \\ w_4 \\ \vdots \\ w_k \\ \vdots \\ w_{nk-2} \end{pmatrix} = \begin{pmatrix} F_3 \\ F_4 \\ \vdots \\ F_k \\ \vdots \\ F_{nk-2} \end{pmatrix} \tag{10.35}$$

The forward elimination and the backward substitution are carried out to solve this matrix by the Gaussian Elimination. The first line in (10.35) are divided by B_3 to carry out the forward elimination.

$$\begin{pmatrix} 1 & c_3 & 0 & \cdots & \cdots & \cdots & 0 \\ A_4 & B_4 & C_4 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & A_k & B_k & C_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & A_{nk-2} & B_{nk-2} \end{pmatrix} \begin{pmatrix} w_3 \\ w_4 \\ \vdots \\ w_k \\ \vdots \\ w_{nk-2} \end{pmatrix} = \begin{pmatrix} f_3 \\ F_4 \\ \vdots \\ F_k \\ \vdots \\ F_{nk-2} \end{pmatrix} \tag{10.36}$$

where $c_3 = C_3 / B_3$ and $f_3 = F_3 / B_3$. The diagonal components of the values which subtracted from the second line to the values which multiplied the first line by A_4 are standardized as follows.

$$\begin{pmatrix} 1 & c_3 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & c_4 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 \cdots & A_k & B_k & C_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 \cdots & \cdots & \cdots & \cdots & A_{nk-2} & B_{nk-2} \end{pmatrix} \begin{pmatrix} w_3 \\ w_4 \\ \vdots \\ w_k \\ \vdots \\ w_{nk-2} \end{pmatrix} = \begin{pmatrix} f_3 \\ f_4 \\ \vdots \\ F_k \\ \vdots \\ F_{nk-2} \end{pmatrix} \tag{10.37}$$

where $c_4 = C_4 / (B_4 - A_4 c_3)$ and $f_4 = (F_4 - A_4 f_3) / (B_4 - A_4 c_3)$. Similarly, using $c_k = C_k / (B_k - A_k c_{k-1})$ and $f_k = (F_k - A_k f_{k-1}) / (B_k - A_k c_{k-1})$,

$$\begin{pmatrix} 1 & c_3 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & c_4 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 \cdots & 0 & 1 & c_k & \cdots & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 \cdots & \cdots & \cdots & \cdots & 0 & 1 & \end{pmatrix} \begin{pmatrix} w_3 \\ w_4 \\ \vdots \\ w_k \\ \vdots \\ w_{nk-2} \end{pmatrix} = \begin{pmatrix} f_3 \\ f_4 \\ \vdots \\ f_k \\ \vdots \\ f_{nk-2} \end{pmatrix} \tag{10.38}$$

Finally, solutions for all k are obtained to carry out the backward substitution in order of $k = nk-2, nk-3$ and $\cdots, 4$.

$$\begin{aligned} w_{nk-2} &= f_{nk-2} \\ w_{nk-3} &= f_{nk-3} - c_{nk-3} w_{nk-2} \\ &\vdots \\ w_k &= f_k - c_k w_{k+1} \\ &\vdots \\ w_4 &= f_4 - c_4 w_5 \\ w_3 &= f_3 - c_3 w_4 \end{aligned} \tag{10.39}$$

The term (F_ϕ^t) which is calculated with the large time step is represented as follows.

$$F_u^t = -\mathrm{Adv}.u^t + \left[\overline{\rho^* f_s \overline{v}^y}^x - \overline{\rho^* f_c \overline{w}^\zeta}^x\right]^t + \left[G^{\frac{1}{2}} \mathrm{Turb}.u\right]^{t-\Delta t} \tag{10.40}$$

$$F_v^t = -\mathrm{Adv}.v^t - \left[\overline{\rho^* f_s \overline{u}^x}^y\right]^t + \left[G^{\frac{1}{2}} \mathrm{Turb}.v\right]^{t-\Delta t} \tag{10.41}$$

$$F_w^t = -\mathrm{Adv}.w^t + \left[\overline{\rho^* \left(\mathrm{Buoy}.\theta + \mathrm{Buoy}.q\right)}^\zeta + \overline{\rho^* f_c \overline{u}^x}^\zeta\right]^t + \left[G^{\frac{1}{2}} \mathrm{Turb}.w\right]^{t-\Delta t} \tag{10.42}$$

$$F_p^t = -\mathrm{Adv}.p^t \tag{10.43}$$

where Buoy.θ and Buoy.q are buoyancy terms for temperature perturbations, watar vapor and hydrometeors, and are defined as follows.

$$\text{Buoy}.\theta \equiv g\frac{\theta'}{\bar{\theta}} \tag{10.44}$$

$$\text{Buoy}.q \equiv g\left(\frac{q_v'}{\epsilon + \bar{q}_v} - \frac{q_v' + \sum q_x}{1 + \bar{q}_v}\right) \tag{10.45}$$

The advection term is calculated with the second or fourth-order accuracy by using the values at the time (t). The second-order advection term is descretized as follows.

$$\text{Adv}.u^t = \overline{\overline{u^*}^x \partial_x u}^x + \overline{\overline{v^*}^x \partial_y u}^y + \overline{\overline{W^*}^x \partial_\zeta u}^\zeta \tag{10.46}$$

$$\text{Adv}.v^t = \overline{\overline{u^*}^y \partial_x v}^x + \overline{\overline{v^*}^y \partial_y v}^y + \overline{\overline{W^*}^y \partial_\zeta v}^\zeta \tag{10.47}$$

$$\text{Adv}.w^t = \overline{\overline{u^*}^\zeta \partial_x w}^x + \overline{\overline{v^*}^\zeta \partial_y w}^y + \overline{\overline{W^*}^\zeta \partial_\zeta v}^\zeta \tag{10.48}$$

$$\text{Adv}.p^t = \overline{\overline{G^{\frac{1}{2}}}^x u \partial_x p'}^x + \overline{\overline{G^{\frac{1}{2}}}^y v \partial_y p'}^y + \overline{\overline{G^{\frac{1}{2}}}^y W \partial_\zeta p'}^\zeta \tag{10.49}$$

The fourth-order advection term is descretized as follows.

$$\begin{aligned}\text{Adv}.u^t = &\frac{4}{3}\left[\overline{\overline{u^*}^x \partial_x u}^x + \overline{\overline{v^*}^x \partial_y u}^y + \overline{\overline{W^*}^x \partial_\zeta u}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^{2x} \partial_{2x} u}^{2x} + \overline{\overline{v^*}^{xy} \partial_{2y} u}^{2y} + \overline{\overline{W^*}^{x\zeta} \partial_{2\zeta} u}^{2\zeta}\right]\end{aligned} \tag{10.50}$$

$$\begin{aligned}\text{Adv}.v^t = &\frac{4}{3}\left[\overline{\overline{u^*}^y \partial_x v}^x + \overline{\overline{v^*}^y \partial_y v}^y + \overline{\overline{W^*}^y \partial_\zeta v}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^{xy} \partial_{2x} v}^{2x} + \overline{\overline{v^*}^{2y} \partial_{2y} v}^{2y} + \overline{\overline{W^*}^{y\zeta} \partial_{2\zeta} v}^{2\zeta}\right]\end{aligned} \tag{10.51}$$

$$\begin{aligned}\text{Adv}.w^t = &\frac{4}{3}\left[\overline{\overline{u^*}^\zeta \partial_x w}^x + \overline{\overline{v^*}^\zeta \partial_y w}^y + \overline{\overline{W^*}^\zeta \partial_\zeta v}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^{x\zeta} \partial_{2x} w}^{2x} + \overline{\overline{v^*}^{y\zeta} \partial_{2y} w}^{2y} + \overline{\overline{W^*}^{2\zeta} \partial_{2\zeta} v}^{2\zeta}\right]\end{aligned} \tag{10.52}$$

$$\text{Adv}.p^t = \frac{4}{3}\left[\overline{\overline{G^{\frac{1}{2}}}^x u \partial_x p'}^x + \overline{\overline{G^{\frac{1}{2}}}^y v \partial_y p'}^y + \overline{\overline{G^{\frac{1}{2}}}^y W \partial_\zeta p'}^\zeta\right]$$

$$-\frac{1}{3}\left[\overline{\overline{G^{\frac{1}{2}}}^{x}}^{2x}\ u\ \partial_{2x}p' \quad +\overline{\overline{G^{\frac{1}{2}}}^{y}}^{2y}\ v\ \partial_{y}p' \quad +\overline{\overline{G^{\frac{1}{2}}}^{\zeta}}^{2\zeta}\ W\ \partial_{2\zeta}p'\right] \tag{10.53}$$

The computing instability is caused in the calculation with discretization of central difference. So an artificial numerical viscosity term which is shown in the following section 10.2 is added to the above terms.

Variables except for pressure in a turbulent diffusion term are applied in the turbulent diffsion terms. The tensor for deformation velocity which are indicated in (7.18) $\sim$ (7.23) are solved as follows by using calculus of finite differences.

$$S_{11} = \frac{2}{G^{\frac{1}{2}}}\left[\partial_x\left(\overline{J_d}^{x}u\right) + \partial_\zeta\left(\overline{J_{31}\overline{u}^{\zeta}}^{x}\right)\right] \tag{10.54}$$

$$S_{22} = \frac{2}{G^{\frac{1}{2}}}\left[\partial_y\left(\overline{J_d}^{y}v\right) + \partial_\zeta\left(\overline{J_{32}\overline{v}^{\zeta}}^{y}\right)\right] \tag{10.55}$$

$$S_{33} = \frac{2}{G^{\frac{1}{2}}}\partial_\zeta w \tag{10.56}$$

$$S_{12} = \frac{1}{\overline{\overline{G^{\frac{1}{2}}}^{xy}}}\left[\partial_y\left(\overline{J_d}^{x}u\right) + \partial_x\left(\overline{J_d}^{y}v\right) + \partial_\zeta\left(\overline{J_{32}}^{x}\overline{u}^{\zeta y} + \overline{J_{31}}^{y}\overline{v}^{\zeta x}\right)\right] \tag{10.57}$$

$$S_{13} = \frac{1}{\overline{\overline{G^{\frac{1}{2}}}^{x\zeta}}}\left[\partial_x\left(\overline{J_d}^{\zeta}w\right) + \partial_\zeta\left(u + \overline{J_{31}\overline{w}^{x}}^{\zeta}\right)\right] \tag{10.58}$$

$$S_{23} = \frac{1}{\overline{\overline{G^{\frac{1}{2}}}^{y\zeta}}}\left[\partial_y\left(\overline{J_d}^{\zeta}w\right) + \partial_\zeta\left(v + \overline{J_{32}\overline{w}^{y}}^{\zeta}\right)\right] \tag{10.59}$$

The divergence term Div which is indicated in 7.24 are solved as follows by using calculus of finite differences.

$$Div = \frac{1}{G^{\frac{1}{2}}}\left[\partial_x\left(\overline{G^{\frac{1}{2}}}^{x}u\right) + \partial_y\left(\overline{G^{\frac{1}{2}}}^{y}v\right) + \partial_\zeta\left(\overline{G^{\frac{1}{2}}}^{\zeta}W\right)\right] \tag{10.60}$$

Using these, the stress tensor which are indicated in (7.9) $\sim$ (7.17) are written by finit differences as follows.

$$\tau_{11} = \bar{\rho}\nu_{\tau h}\left(S_{11} - \frac{2}{3}Div\right) \tag{10.61}$$

$$\tau_{12} = \overline{\bar{\rho}\nu_{\tau h}}^{xy}S_{12} \tag{10.62}$$

$$\tau_{13} = \overline{\bar{\rho}\nu_{\tau v}}^{x\zeta}S_{13} \tag{10.63}$$

$$\tau_{21} = \overline{\bar{\rho}\nu_{\tau h}}^{yx} S_{21} \tag{10.64}$$

$$\tau_{22} = \bar{\rho}\nu_{\tau h}\left(S_{22} - \frac{2}{3}Div\right) \tag{10.65}$$

$$\tau_{23} = \overline{\bar{\rho}\nu_{\tau v}}^{y\zeta} S_{23} \tag{10.66}$$

$$\tau_{31} = \overline{\bar{\rho}\nu_{\tau h}}^{\zeta x} S_{31} \tag{10.67}$$

$$\tau_{32} = \overline{\bar{\rho}\nu_{\tau h}}^{\zeta y} S_{32} \tag{10.68}$$

$$\tau_{33} = \bar{\rho}\nu_{\tau v}\left(S_{33} - \frac{2}{3}Div\right) \tag{10.69}$$

Finally, the turbulent diffusion terms are indicated in differences as follows.

$$G^{\frac{1}{2}}\mathrm{Turb}.u = \partial_x\left(J_d\tau_{11}\right) + \partial_y\left(\overline{J_d}^{xy}\tau_{12}\right) + \partial_\zeta\left(\tau_{13} + J_{31}\overline{\tau_{11}}^{x\zeta} + \overline{\overline{J_{32}}^{x}\overline{\tau_{12}}^{\zeta}}^{y}\right) \tag{10.70}$$

$$G^{\frac{1}{2}}\mathrm{Turb}.v = \partial_x\left(\overline{J_d}^{yx}\tau_{21}\right) + \partial_y\left(J_d\tau_{22}\right) + \partial_\zeta\left(\tau_{23} + \overline{\overline{J_{31}}^{y}\overline{\tau_{21}}^{\zeta}}^{x} + J_{32}\overline{\tau_{22}}^{y\zeta}\right) \tag{10.71}$$

$$G^{\frac{1}{2}}\mathrm{Turb}.w = \partial_x\left(\overline{J_d}^{\zeta x}\tau_{31}\right) + \partial_y\left(\overline{J_d}^{\zeta y}\tau_{32}\right) + \partial_\zeta\left(\tau_{33} + \overline{\overline{J_{31}}^{\zeta}\overline{\tau_{31}}^{\zeta}}^{x} + \overline{\overline{J_{32}}^{\zeta}\overline{\tau_{32}}^{\zeta}}^{y}\right) \tag{10.72}$$

The method of discretization for physical quanities without the sound wave mode

When the gravity wave mode is calculated with the large time step, the equations of potential temperature, watar vapor and hydrometeors are solved with only large time step. These equations are shown by indicating watar vapor and hydrometeors as x as follows.

$$\rho^* \frac{\theta'^{t+\Delta t} - \theta'^{t-\Delta t}}{2\Delta t} = F_\theta^t \tag{10.73}$$

$$\rho^* \frac{q_x^{t+\Delta t} - q_x^{t-\Delta t}}{2\Delta t} = F_q^t \tag{10.74}$$

$$G^{\frac{1}{2}} \frac{N_x^{t+\Delta t} - N_x^{t-\Delta t}}{2\Delta t} = F_N^t \tag{10.75}$$

where

$$F_\theta^t = -\mathrm{Adv}.\theta^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.\theta\right]^{t-\Delta t} + \left[\rho^*\mathrm{Src}.\theta\right]^t - \left[\overline{\bar{\rho}^\zeta w\partial_\zeta\bar{\theta}}^{\zeta}\right]^t \tag{10.76}$$

$$F_q^t = -\text{Adv.}q_x^t + \left[G^{\frac{1}{2}}\text{Turb.}q_x\right]^{t-\Delta t} + [\rho^*\text{Src.}q_x]^t + [\rho^*\text{Fall.}q_x]^t \tag{10.77}$$

$$F_N^t = -\text{Adv.}\frac{N_x}{\rho^*}^t + \left[G^{\frac{1}{2}}\text{Turb.}\frac{N_x}{\rho^*}\right]^{t-\Delta t} + \left[\rho^*\text{Src.}\frac{N_x}{\rho^*}\right]^t + \left[\rho^*\text{Fall.}\frac{N_x}{\rho^*}\right]^t. \tag{10.78}$$

The fourth term on the right side of the equation of potential temperature perturbation (10.76) is The term of gravity wave with vertical advection of potential temperature in the basic state.

The advection term is calculated as physical quanities with the sound wave mode by using the values at the time t. The advection term is solved by using the second-order scheme as follows.

$$\text{Adv.}\theta^t = \overline{u^*\partial_x\theta}^x + \overline{v^*\partial_y\theta}^y + \overline{W^*\partial_\zeta\theta}^\zeta \tag{10.79}$$

$$\text{Adv.}q_x^t = \overline{u^*\partial_x q_x}^x + \overline{v^*\partial_y q_x}^y + \overline{W^*\partial_\zeta q_x}^\zeta \tag{10.80}$$

$$\text{Adv.}\frac{N_x}{\rho^*}^t = \overline{u^*\partial_x\frac{N_x}{\rho^*}}^x + \overline{v^*\partial_y\frac{N_x}{\rho^*}}^y + \overline{W^*\partial_\zeta\frac{N_x}{\rho^*}}^\zeta \tag{10.81}$$

The advection terms are solved by using the fourth-order scheme as follows.

$$\begin{aligned}\text{Adv.}\theta'^t = &\frac{4}{3}\left[\overline{u^*\partial_x\theta}^x + \overline{v^*\partial_y\theta}^y + \overline{W^*\partial_\zeta\theta}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^x\partial_{2x}\theta'}^{2x} + \overline{\overline{v^*}^y\partial_{2y}\theta'}^{2y} + \overline{\overline{W^*}^\zeta\partial_{2\zeta}\theta'}^{2\zeta}\right]\end{aligned} \tag{10.82}$$

$$\begin{aligned}\text{Adv.}q_x^t = &\frac{4}{3}\left[\overline{u^*\partial_x q_x}^x + \overline{v^*\partial_y q_x}^y + \overline{W^*\partial_\zeta q_x}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^x\partial_{2x}q_x}^{2x} + \overline{\overline{v^*}^y\partial_{2y}q_x}^{2y} + \overline{\overline{W^*}^\zeta\partial_{2\zeta}q_x}^{2\zeta}\right]\end{aligned} \tag{10.83}$$

$$\begin{aligned}\text{Adv.}\frac{N_x}{\rho^*}^t = &\frac{4}{3}\left[\overline{u^*\partial_x\frac{N_x}{\rho^*}}^x + \overline{v^*\partial_y\frac{N_x}{\rho^*}}^y + \overline{W^*\partial_\zeta\frac{N_x}{\rho^*}}^\zeta\right] \\ &- \frac{1}{3}\left[\overline{\overline{u^*}^x\partial_{2x}\frac{N_x}{\rho^*}}^{2x} + \overline{\overline{v^*}^y\partial_{2y}\frac{N_x}{\rho^*}}^{2y} + \overline{\overline{W^*}^\zeta\partial_{2\zeta}\frac{N_x}{\rho^*}}^{2\zeta}\right]\end{aligned} \tag{10.84}$$

The turbulent flux $H_{\phi 1}, H_{\phi 2}$, and $H_{\phi 3}$ for one scalar ϕ which is represented by (7.26) $\sim$ (7.28) are solved by using difference as follows.

$$H_{\phi 1} = \overline{\left(\bar{\rho}\nu_{Hh}\frac{1}{G^{\frac{1}{2}}}\right)}^x\left[\partial_x\left(J_d\phi\right) + \partial_\zeta\left(J_{31}\overline{\phi}^{x\zeta}\right)\right] \tag{10.85}$$

$$H_{\phi 2} = \overline{\left(\bar{\rho}\nu_{Hh}\frac{1}{G^{\frac{1}{2}}}\right)}^y\left[\partial_y\left(J_d\phi\right) + \partial_\zeta\left(J_{32}\overline{\phi}^{y\zeta}\right)\right] \tag{10.86}$$

$$H_{\phi 3} = \overline{\left(\bar{\rho}\nu_{Hv}\frac{1}{G^{\frac{1}{2}}}\right)}^{\zeta}\partial_\zeta\overline{\phi}^{\zeta} \tag{10.87}$$

Finally, the turbulent diffsion term is indicated as follows.

$$G^{\frac{1}{2}}\mathrm{Turb}.\theta = \partial_x\left(\overline{J_d}^{x}H_{\theta 1}\right) + \partial_y\left(\overline{J_d}^{y}H_{\theta 2}\right) + \partial_\zeta\left(H_{\theta 3} + \overline{J_{31}\overline{H_{\theta 1}}^{\zeta}}^{x} + \overline{J_{32}\overline{H_{\theta 2}}^{\zeta}}^{y}\right) \tag{10.88}$$

$$G^{\frac{1}{2}}\mathrm{Turb}.q_x = \partial_x\left(\overline{J_d}^{x}H_{q_x 1}\right) + \partial_y\left(\overline{J_d}^{y}H_{q_x 2}\right) + \partial_\zeta\left(H_{q_x 3} + \overline{J_{31}\overline{H_{q_x 1}}^{\zeta}}^{x} + \overline{J_{32}\overline{H_{q_x 2}}^{\zeta}}^{y}\right) \tag{10.89}$$

$$G^{\frac{1}{2}}\mathrm{Turb}.\frac{N_x}{\rho^*} = \partial_x\left(\overline{J_d}^{x}H_{N_x 1}\right) + \partial_y\left(\overline{J_d}^{y}H_{N_x 2}\right) + \partial_\zeta\left(H_{N_x 3} + \overline{J_{31}\overline{H_{N_x 1}}^{\zeta}}^{x} + \overline{J_{32}\overline{H_{N_x 2}}^{\zeta}}^{y}\right) \tag{10.90}$$

The method of discretization in the case of calculating the gravity wave mode with small time step

When the gravity wave mode is calculated with small time step, the equation of potential temperature is indicated as follows.

$$\frac{\partial\theta^*}{\partial t} = \underbrace{-\left(u^*\frac{\partial\theta'}{\partial x} + v^*\frac{\partial\theta'}{\partial y} + W^*\frac{\partial\theta'}{\partial\zeta}\right)}_{[rm]} \underbrace{-\bar{\rho}w\frac{\partial\bar{\theta}}{\partial\zeta}}_{[gm]} + \underbrace{G^{\frac{1}{2}}\mathrm{Turb}.\theta}_{[physics]} + \underbrace{\rho^*\mathrm{Src}.\theta}_{[physics]} \tag{10.91}$$

The second term on the right side of is calculated by using the values at the time step $\Delta\tau$. The time integration of this equation is shown as follows.

$$\rho^*\frac{\theta'^{\tau+\Delta\tau} - \theta'^{\tau}}{\Delta\tau} = -\left[\overline{\bar{\rho}^{\zeta}w\partial_\zeta\bar{\theta}}^{\zeta}\right]^{\tau} + F_\theta^t \tag{10.92}$$

The time integration of vertical velocity (w) is shown since the buoyancy terms are calculated by using the values at the time($\Delta\tau$).

$$\begin{aligned}\overline{\rho^*}^{\zeta}\frac{w^{\tau+\Delta\tau} - w^{\tau}}{\Delta\tau} = {} & [\partial_\zeta(\alpha Div^*)]^{\tau} + g\left[\overline{\rho^*\frac{\theta'}{\bar{\theta}}}^{\zeta}\right]^{\tau} - \left[\beta\left(\partial_\zeta p'\right)^{\tau+\Delta\tau} + (1-\beta)\left(\partial_\zeta p'\right)^{\tau}\right] \\ & - \left[\beta\left(\overline{gG^{\frac{1}{2}}\frac{p'}{c_s^2}}^{\zeta}\right)^{\tau+\Delta\tau} + (1-\beta)\left(\overline{gG^{\frac{1}{2}}\frac{p'}{c_s^2}}^{\zeta}\right)^{\tau}\right] + F_w^t\end{aligned} \tag{10.93}$$

Terms which are calculated with large time step are modified as follows by this.

$$F_w^t = -\mathrm{Adv}.w^t + \left[\overline{\rho^* \mathrm{Buoy}.q}^{\zeta} + \overline{\rho^* f_c \overline{u}^x}^{\zeta}\right]^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.w\right]^{t-\Delta t} \tag{10.94}$$

$$F_\theta^t = -\mathrm{Adv}.\theta^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.\theta\right]^{t-\Delta t} + [\rho^* \mathrm{Src}.\theta]^t \tag{10.95}$$

time filter

The large time step integration is solved by using the leap-frog scheme. The separation of solutions between the odd time step and the even time step is caused in the case of using this technique. So the Asselin time filter (Asselin, 1972) is used to control the separation of solutions. This is the technique to filter physical quanities at the time t after physical quanities are calculated at the time $t + \Delta t$.

The Asselin time filter is indicated by using a representative variable ψ for forecasted variables $u, v, w, p', \theta', q_x$.

$$\overline{\psi^t} = \psi^t + \mu_a \left(\overline{\psi^{t-\Delta t}} - 2\psi^t + \psi^{t+\delta t}\right) \tag{10.96}$$

where μ_a is a factor of filter, and $\mu_a = 0.1$ is a standard value.

10.2 Artificial numerical viscosity term

10.2.1 Necessity of numerical viscosity term

In previous section, it is shown that advection term is expressed by second-order or fourth-order central difference, and calculation instability will be generated if the artificial numerical viscosity term is not added. The reason for that is shown as follows.

One-dimensional linear wave equation :

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \qquad (c > 0) \tag{10.97}$$

The equation is discretized as follows. Arrangement of shifted grid points is not considered here.

$$\frac{\partial u}{\partial t} = \frac{u_i^{t+\Delta t} - u_i^t}{\Delta t}, \qquad \frac{\partial u}{\partial x} = \frac{u_{i+\Delta x}^t - u_{i-\Delta x}^t}{2\Delta x} \tag{10.98}$$

The equation (10.97) is written by

$$u_i^{t+\Delta t} = u_i^t - \frac{c}{2}\left(\frac{\Delta t}{\Delta x}\right)\left(u_{i+\Delta x}^t - u_{i-\Delta x}^t\right) \tag{10.99}$$

From this equation, the value at time $t + \Delta t$ is calculated by using the value at time t. However, when this calculation is performed, this calculation will break even if the relation of Δt and Δx satisfys the CFL condition.

As the other discrete method, we consider first-order windward difference because its calculation can be stable. However, it does not mean higher accuracy for the calculation. If this discretization is applied to an equation (10.97) and solved as follows.

$$u_i^{t+\Delta t} = u_i^t - c\left(\frac{\Delta t}{\Delta x}\right)\left(u_i^t - u_{i-\Delta x}^t\right) \tag{10.100}$$

However, if this is transformed into the form using the equation (10.99) which is the first discretization,

$$u_i^{t+\Delta t} = u_i^t - \frac{c}{2}\left(\frac{\Delta t}{\Delta x}\right)\left(u_{i+\Delta x}^t - u_{i-\Delta x}^t\right) + \frac{c}{2}\left(\frac{\Delta t}{\Delta x}\right)\left(u_{i+\Delta x}^t - 2u_i^t + u_{i-\Delta x}^t\right) \tag{10.101}$$

The added term is difference approximation of space second-degree differentiation $\dfrac{\partial^2 u}{\partial x^2}$, and affects as the diffusion term. Calculation is stably preformed using the first-order forward difference from this reason. In central difference, however, for making the diffusion term, the artificial numerical viscosity term is added to the advection term.

10.2.2 Numerical viscosity with the second-order or fourth-order diffusion term

Since the eventh-degree differentiation generally has the effect of diffusion, the deffision term explained in the previous subsection can change the degree of the differentiation. Here, numerical viscosities with the second-order and the fourth-order diffusion terms are explained.

Firstly, the second-order viscosity term is represented by the arbitrary physical quantities ϕ,

$$G^{\frac{1}{2}}\mathrm{Diff}.\phi = \nu_{2h}\left[\frac{\partial^2\left(\bar{\rho}\phi'\right)}{\partial x^2} + \frac{\partial^2\left(\bar{\rho}\phi'\right)}{\partial y^2}\right] + \nu_{2v}\frac{\partial^2\left(\bar{\rho}\phi'\right)}{\partial \zeta^2}. \tag{10.102}$$

When ϕ are $u, v, w, q_x, N_x/\rho^*$, ϕ' is perturbation from an initial value. When ϕ is θ, ϕ is the perturbation from the basic state.

ν_{2h} and ν_{2v} are viscosity coefficient, and are defined as follows.

$$\nu_{2h} = \frac{\alpha_{2h}\Delta_h^2}{\Delta t} \tag{10.103}$$

$$\nu_{2v} = \frac{\alpha_{2v}\Delta_v^2}{\Delta t} \tag{10.104}$$

where,

$$\Delta_h = (\Delta x \Delta y)^{\frac{1}{2}} \tag{10.105}$$

$$\Delta_v = \Delta\zeta \tag{10.106}$$

In these equations, Δ_h and Δ_v are the horizontal and vertical averaged grid interval in the domain. Moreover, α_{2h} and α_{2v} are the non-dimensional quantities. For a stable calculation, the following requirement needs to be satisfied.

$$\alpha_{2h} \leq \frac{1}{8}, \quad \alpha_{2v} \leq \frac{1}{8} \tag{10.107}$$

In this way, this viscosity term is applied to forecast variables other than pressure perturbation.

We define the second-order differentiation operator as

$$\left(\partial_x^2 \phi\right)_i = \frac{1}{\Delta x}\left(\phi_{i+\Delta x} - 2\phi_i + \phi_{i-\Delta x}\right) \tag{10.108}$$

When $u', v', w', q_x', N_x'/\rho^*$ are made into perturbation from the initial value and θ' is made into perturbation from the basic state, difference notation are shown as follows.

$$G^{\frac{1}{2}}\mathrm{Diff}.u = \nu_{2h}\left[\partial_x^2\left(\overline{\bar{\rho}}^x u'\right) + \partial_y^2\left(\overline{\bar{\rho}}^x u'\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^x u'\right)\right] \tag{10.109}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.v = \nu_{2h}\left[\partial_x^2\left(\overline{\bar{\rho}}^y v'\right) + \partial_y^2\left(\overline{\bar{\rho}}^y v'\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^y v'\right)\right] \tag{10.110}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.w = \nu_{2h}\left[\partial_x^2\left(\overline{\bar{\rho}}^\zeta w'\right) + \partial_y^2\left(\overline{\bar{\rho}}^\zeta w'\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^\zeta w'\right)\right] \tag{10.111}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.\theta = \nu_{2h}\left[\partial_x^2\left(\bar{\rho}\theta'\right) + \partial_y^2\left(\bar{\rho}\theta'\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\bar{\rho}\theta'\right)\right] \tag{10.112}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.q_x = \nu_{2h}\left[\partial_x^2\left(\bar{\rho}q_x'\right) + \partial_y^2\left(\bar{\rho}q_x'\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\bar{\rho}q_x'\right)\right] \tag{10.113}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.\frac{N_x}{\rho^*} = \nu_{2h}\left[\partial_x^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right) + \partial_y^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right)\right] + \nu_{2v}\left[\partial_\zeta^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right)\right] \tag{10.114}$$

The fourth-order numerical viscosity term is considered as well as the second-order. When ϕ is $u, v, w, q_x, N_x/\rho^*$, ϕ' is perturbation from an initial value. When ϕ is θ, ϕ is the perturbation from the basic state.

$$G^{\frac{1}{2}}\mathrm{Diff}.\phi = -\nu_{4h}\left[\frac{\partial^4(\bar{\rho}\phi')}{\partial x^4} + \frac{\partial^4(\bar{\rho}\phi')}{\partial y^4}\right] - \nu_{4v}\frac{\partial^4(\bar{\rho}\phi')}{\partial \zeta^4} \tag{10.115}$$

ν_{2h} and ν_{2v} are a viscosity coefficient, and are defined as follows.

$$\nu_{4h} = \frac{\alpha_{4h}\Delta_h^4}{\Delta t} \tag{10.116}$$

$$\nu_{4v} = \frac{\alpha_{4v}\Delta_v^4}{\Delta t} \tag{10.117}$$

where α_{2h} and α_{2v} are the non-dimensional quantities, and the values which are often used are as follows.

$$\alpha_{4h} = 0.001, \qquad \alpha_{4v} = 0.001. \tag{10.118}$$

As well as the second-order, this viscosity term is applied to forecast variables other than pressure perturbation.

When $u', v', w', q_x', N_x'/\rho^*$ are made into perturbation from initial value and θ' is made into perturbation from the basic state, difference notation are shown as follows.

$$G^{\frac{1}{2}}\mathrm{Diff}.u = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\overline{\bar{\rho}}^x u'\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\overline{\bar{\rho}}^x u'\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^x u'\right)\right] \tag{10.119}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.v = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\overline{\bar{\rho}}^y v'\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\overline{\bar{\rho}}^y v'\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^y v'\right)\right] \tag{10.120}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.w = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\overline{\bar{\rho}}^\zeta w'\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\overline{\bar{\rho}}^\zeta w'\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\overline{\bar{\rho}}^\zeta w'\right)\right] \tag{10.121}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.\theta = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\bar{\rho}\theta'\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\bar{\rho}\theta'\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\bar{\rho}\theta'\right)\right] \tag{10.122}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.q_x = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\bar{\rho}q_x'\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\bar{\rho}q_x'\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\bar{\rho}q_x'\right)\right] \tag{10.123}$$

$$G^{\frac{1}{2}}\mathrm{Diff}.\frac{N_x}{\rho^*} = -\nu_{4h}\left[\partial_x^2\left\{\partial_x^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right)\right\} + \partial_y^2\left\{\partial_y^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right)\right\}\right] - \nu_{4v}\partial_\zeta^2\left[\partial_\zeta^2\left(\bar{\rho}\frac{N_x'}{\rho^*}\right)\right] \tag{10.124}$$

It is more desirable to use this numerical viscosity term which has fourth-order diffusion term, since it can distinguish from the diffusion term Turb.ϕ in Section 7 - "subgrid-scale diffusion", and can diffuse the ingredient of high wave number.

As a result, including these numerical viscosity term, the term F_ϕ^t calculated with large time step t shown in (10.40) $\sim$ (10.43), (10.76) and (10.77) are able to be expressed as follows.

$$F_u^t = -\mathrm{Adv}.u^t + \left[\overline{\rho^* f_s \overline{v}^y}^x - \overline{\rho^* f_c \overline{w}^\zeta}^x\right]^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.u + G^{\frac{1}{2}}\mathrm{Diff}.u\right]^{t-\Delta t} \tag{10.125}$$

$$F_v^t = -\mathrm{Adv}.v^t - \left[\overline{\rho^* f_s \overline{u}^x}^y\right]^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.v + G^{\frac{1}{2}}\mathrm{Diff}.v\right]^{t-\Delta t} \tag{10.126}$$

$$F_w^t = -\mathrm{Adv}.w^t + \left[\overline{\rho^* \left(\mathrm{Buoy}.\theta + \mathrm{Buoy}.q\right)}^\zeta + \overline{\rho^* f_c \overline{u}^x}^\zeta\right]^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.w + G^{\frac{1}{2}}\mathrm{Diff}.w\right]^{t-\Delta t} \tag{10.127}$$

$$F_\theta^t = -\mathrm{Adv}.\theta'^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.\theta' + G^{\frac{1}{2}}\mathrm{Diff}.\theta\right]^{t-\Delta t} + \left[\rho^*\mathrm{Src}.\theta'\right]^t - \left[\overline{\overline{\rho}^\zeta w \partial_\zeta \bar{\theta}}^\zeta\right]^t \tag{10.128}$$

$$F_q^t = -\mathrm{Adv}.q_x^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.q_x + G^{\frac{1}{2}}\mathrm{Diff}.q_x\right]^{t-\Delta t} + \left[\rho^*\mathrm{Src}.q_x\right]^t + \left[\rho^*\mathrm{Fall}.q_x\right]^t \tag{10.129}$$

$$F_N^t = -\mathrm{Adv}.\frac{N_x}{\rho^*}^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.\frac{N_x}{\rho^*} + G^{\frac{1}{2}}\mathrm{Diff}.\frac{N_x}{\rho^*}\right]^{t-\Delta t} + \left[\rho^*\mathrm{Src}.\frac{N_x}{\rho^*}\right]^t + \left[\rho^*\mathrm{Fall}.\frac{N_x}{\rho^*}\right]^t \tag{10.130}$$

The terms F_w^t and F_θ^t in the case of calculating gravity wave mode with small time step shown in (10.94) and (10.95) is able to be expressed as follows.

$$F_w^t = -\mathrm{Adv}.w^t + \left[\overline{\rho^* \mathrm{Buoy}.q}^\zeta + \overline{\rho^* f_c \overline{u}^x}^\zeta\right]^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.w + G^{\frac{1}{2}}\mathrm{Diff}.w\right]^{t-\Delta t} \tag{10.131}$$

$$F_\theta^t = -\mathrm{Adv}.\theta^t + \left[G^{\frac{1}{2}}\mathrm{Turb}.\theta + G^{\frac{1}{2}}\mathrm{Diff}.\theta\right]^{t-\Delta t} + \left[\rho^*\mathrm{Src}.\theta\right]^t \tag{10.132}$$

10.3 Boundary condition

Finit difference of the basic equations shown in the previous section includes operators at $i-1,\ i+1,\ j-1, \ldots$. For calculation of forecast variables at he grid points which locate along each boundary:

$$\begin{array}{llll} u; & i=1,nx, & j=1,ny-1, & k=1,nz-1 \\ v; & i=1,nx-1, & j=1,ny, & k=1,nz-1 \\ w; & i=1,nx-1, & j=1,ny-1, & k=1,nz \\ \phi; & i=1,nx-1, & j=1,ny-1, & k=1,nz-1, \end{array} \tag{10.133}$$

where ϕ of the arbitrary scalar, and a boundary condition is needed to be given. We will explain about some kinds of a boundary condition in ***CReSS*** in the followings. In the fourth-order calculation, values at the grid points shifted one grid from the boundary are calculated with the second-order precision.

10.3.1 Lateral boundary condition

Four boundary conditions of periodic, fixed wall (mirror condition), non-gradient, and radiation, are applicable to a lateral boundary.

Periodic boundary condition

The periodic boundary condition is the condition that the values along the eastern (northern) boundary are equal to those along the western (southern) boundary during an integration.

In the cast that we use the periodic boundary condition in the east-west direction,

$$\begin{aligned}
&u_{1,j,k} = u_{nx-2,j,k}, && u_{nx,j,k} = u_{3,j,k} \\
&v_{1,j,k} = v_{nx-2,j,k}, && v_{nx-1,j,k} = v_{2,j,k} \\
&w_{1,j,k} = w_{nx-2,j,k}, && w_{nx-1,j,k} = w_{2,j,k} \\
&\phi_{1,j,k} = \phi_{nx-2,j,k}, && \phi_{nx-1,j,k} = \phi_{2,j,k} \\
&W_{1,j,k} = W_{nx-2,j,k}, && W_{nx-1,j,k} = W_{2,j,k},
\end{aligned} \tag{10.134}$$

or in the cast that we use the periodic boundary condition in the north-south direction,

$$\begin{aligned}
&u_{i,1,k} = u_{i,ny-2,k}, && u_{i,ny-1,k} = u_{i,2,k} \\
&v_{i,1,k} = v_{i,ny-2,k}, && v_{i,ny,k} = v_{i,3,k} \\
&w_{i,1,k} = w_{i,ny-2,k}, && w_{i,ny-1,k} = w_{i,2,k} \\
&\phi_{i,1,k} = \phi_{i,ny-2,k}, && \phi_{i,ny-1,k} = \phi_{i,2,k} \\
&W_{i,1,k} = W_{i,ny-2,k}, && W_{i,ny-1,k} = W_{i,2,k},
\end{aligned} \tag{10.135}$$

where ϕ is the arbitrary scalar.

Fixed wall boundary condition

In order to explain fixed wall boundary condition, for example, an east-and-west wall is shown here. Along the east and west boundaries, x differentiation in uncompressible equation of conservation of mass and free-slip condition are

$$\partial\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial \zeta}\right) \Big/ \partial x = 0, \qquad \frac{\partial w}{\partial x} = 0. \tag{10.136}$$

It is drawn that the velocity u of x direction satisfys the following relation.

$$\frac{\partial^2 u}{\partial x^2} = 0 \qquad \Longrightarrow \qquad u\left(x = -\Delta x, y, z\right) = u\left(x = \Delta x, y, z\right) \tag{10.137}$$

Under the situation, the fixed wall condition is determined as follows.

In the cast that we use the periodic boundary condition in the east-west direction,

$$\begin{aligned} u_{1,j,k} &= -u_{3,j,k}, & u_{nx,j,k} &= -u_{nx-2,j,k} \\ v_{1,j,k} &= v_{2,j,k}, & v_{nx-1,j,k} &= v_{nx-2,j,k} \\ w_{1,j,k} &= w_{2,j,k}, & w_{nx-1,j,k} &= w_{nx-2,j,k} \\ \phi_{1,j,k} &= \phi_{2,j,k}, & \phi_{nx-1,j,k} &= \phi_{nx-2,j,k} \\ W_{1,j,k} &= W_{2,j,k}, & W_{nx-1,j,k} &= W_{nx-2,j,k}, \end{aligned} \tag{10.138}$$

or in the cast that we use the periodic boundary condition in the north-south direction,

$$\begin{aligned} u_{i,1,k} &= u_{i,2,k}, & u_{i,ny-1,k} &= u_{i,nx-2,k} \\ v_{i,1,k} &= -v_{i,3,k}, & v_{i,ny,k} &= -v_{i,nx-2,k} \\ w_{i,1,k} &= w_{i,2,k}, & w_{i,ny-1,k} &= w_{i,nx-2,k} \\ \phi_{i,1,k} &= \phi_{i,2,k}, & \phi_{i,ny-1,k} &= \phi_{i,nx-2,k} \\ W_{i,1,k} &= W_{i,2,k}, & W_{i,ny-1,k} &= W_{i,nx-2,k}, \end{aligned} \tag{10.139}$$

where ϕ is the arbitrary scalar.

Non-gradient boundary condition

Non-gradient boundary condition is the condition that the gradient of forecast variables across boundary plane may be set to $grad\,(\phi) = 0$.

In the cast that we use the periodic boundary condition in the east-west direction,

$$\begin{aligned} u_{1,j,k} &= u_{3,j,k}, & u_{nx,j,k} &= u_{nx-2,j,k} \\ v_{1,j,k} &= v_{2,j,k}, & v_{nx-1,j,k} &= v_{nx-2,j,k} \\ w_{1,j,k} &= w_{2,j,k}, & w_{nx-1,j,k} &= w_{nx-2,j,k} \\ \phi_{1,j,k} &= \phi_{2,j,k}, & \phi_{nx-1,j,k} &= \phi_{nx-2,j,k} \\ W_{1,j,k} &= W_{2,j,k}, & W_{nx-1,j,k} &= W_{nx-2,j,k}, \end{aligned} \tag{10.140}$$

or in the cast that we use the periodic boundary condition in the north-south direction,

$$\begin{aligned} u_{i,1,k} &= u_{i,2,k}, & u_{i,ny-1,k} &= u_{i,nx-2,k} \\ v_{i,1,k} &= v_{i,3,k}, & v_{i,ny,k} &= v_{i,nx-2,k} \\ w_{i,1,k} &= w_{i,2,k}, & w_{i,ny-1,k} &= w_{i,nx-2,k} \\ \phi_{i,1,k} &= \phi_{i,2,k}, & \phi_{i,ny-1,k} &= \phi_{i,nx-2,k} \\ W_{i,1,k} &= W_{i,2,k}, & W_{i,ny-1,k} &= W_{i,nx-2,k}, \end{aligned} \tag{10.141}$$

where ϕ is the arbitrary scalar.

Radiational boundary condition

Radiational boundary condition is set up so that the radiational conditions with one-dimensional linear wave equation

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0 \tag{10.142}$$

is satisfied for stopping the action of the unnatural solution produced in a boundary plane. Adding the attenuation term,

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = -\gamma\phi \tag{10.143}$$

is used here. As nesting from a model output for a larger domain, such as objective analysis data, as an initial and a boundary values. The difference between ***CReSS*** and the model for a larger domain for each physical amount is used for a boundary condition.

$$\frac{\partial \left(\phi_f - \phi_c\right)}{\partial t} + c\frac{\partial \left(\phi_f - \phi_c\right)}{\partial x} = -\gamma\left(\phi_f - \phi_c\right) \tag{10.144}$$

where ϕ is the arbitrary physical quantities, bottom characters f and c indicate the values of the larger model and this model, respectively.

There are various methods in the way to solve phase velocity c in the above radiational condition. The three methods can be chosen in ***CReSS***. ϕ and $\phi_f - \phi_c$ is expressed with Φ below.

[(1) Method to calculate phase velocity for each grid point](1) Method to calculate phase velocity for each grid point

In this case, equations excluded the attenuation term on the right side from the equations (10.143), (10.144) is solved about phase velocity c. Non-dimensional phase velocity is directly solved.

$$C_b^t = c_b^t\frac{\Delta t}{\Delta x} = -\frac{\Phi_b^{t+\Delta t} - \Phi_b^{t-\Delta t}}{2\Phi_{b-1}^t - \Phi_b^{t+\Delta t} - \Phi_b^{t-\Delta t}}, \qquad -1 \le C_b^t \le 0, \quad \textit{western edge or southern edge} \tag{10.145}$$

$$C_b^t = c_b^t\frac{\Delta t}{\Delta x} = -\frac{\Phi_b^{t+\Delta t} - \Phi_b^{t-\Delta t}}{\Phi_b^{t+\Delta t} + \Phi_b^{t-\Delta t} - 2\Phi_{b-1}^t}, \qquad 0 \le C_b^t \le 1, \quad \textit{eastern edge or northern edge} \tag{10 146}$$

These phase velocities should be calculated for each forecast variables. In the actual calculation, the following values are used.

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = -\frac{\Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t}}{2\Phi_{b-2}^{t-\Delta t} - \Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t}}, \qquad -1 \le C_b^t \le 0,\ westernmostorsouthernmost \tag{10.147}$$

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = -\frac{\Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t}}{\Phi_{b-1}^t + \Phi_{b-1}^{t-2\Delta t} - 2\Phi_{b-2}^{t-\Delta t}}, \qquad 0 \le C_b^t \le 1,\ easternmostornorthernmost \tag{10.148}$$

where bottom character b is the number of grid point on the boundary given by the equation (10.133).

[(2) Method to calculate constant phase velocity in vertical direction](2) Method to calculate constant phase velocity in vertical direction

Wave equation which assumed constant phase velocity in vertical is considered as follows.

$$\int_{z_{sfc}}^{z_{top}} \frac{\partial \Phi}{\partial t} dz + c \int_{z_{sfc}}^{z_{top}} \frac{\partial \Phi}{\partial x} dz = -\gamma \Phi \tag{10.149}$$

Using the value calculated at the inner grid point on preceding time step,

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = -\frac{\sum_{k=kmin}^{kmax} \left\{ \left(\Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t} \right) sgn \left(2\Phi_{b-2}^{t-\Delta t} - \Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t} \right) \right\}}{\sum_{k=kmin}^{kmax} \left| 2\Phi_{b-2}^{t-\Delta t} - \Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t} \right|},$$

$$\cdots \quad -1 \le C_b^t \le 0,\ \ western\ edge\ or\ southern\ edge \tag{10.150}$$

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = -\frac{\sum_{k=kmin}^{kmax} \left\{ \left(\Phi_{b-1}^t - \Phi_{b-1}^{t-2\Delta t} \right) sgn \left(\Phi_{b-1}^t + \Phi_{b-1}^{t-2\Delta t} - 2\Phi_{b-2}^{t-\Delta t} \right) \right\}}{\sum_{k=kmin}^{kmax} \left| \Phi_{b-1}^t + \Phi_{b-1}^{t-2\Delta t} - 2\Phi_{b-2}^{t-\Delta t} \right|},$$

$$\cdots \quad 0 \le C_b^t \le 1,\ \ eastern\ edge\ or\ northern\ edge \tag{10.151}$$

where $kmin$ and $kmax$ are

$$u,\ v,\ \phi\ ; \quad kmin = 2,\ kmax = nz - 2 \tag{10.152}$$

$$w\ ; \quad kmin = 2,\ kmax = nz - 1. \tag{10.153}$$

[(3) Method to add advectional velocity to constant phase velocity](3) Method to add advectional

velocity to constant phase velocity

Phase velocities should be calculated for each forecast variable. c^* is given as the propagation velocity of typical gravity wave, such as about 30 m s^{-1}. Using x-component velocity u for the eastern and western boundaries and y-component velocity v for the northern and southern boundaries, a boundary condition is given as follows.

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = (u - c^*) \frac{\Delta t}{\Delta x}, \qquad -1 \le C_b^t \le 0, \ westernmost \tag{10.154}$$

$$C_b^t = c_b^t \frac{\Delta t}{\Delta x} = (u + c^*) \frac{\Delta t}{\Delta x}, \qquad 0 \le C_b^t \le 1, \ easternmost \tag{10.155}$$

$$C_b^t = c_b^t \frac{\Delta t}{\Delta y} = (v - c^*) \frac{\Delta t}{\Delta y}, \qquad -1 \le C_b^t \le 0, \ southernmost \tag{10.156}$$

$$C_b^t = c_b^t \frac{\Delta t}{\Delta y} = (v + c^*) \frac{\Delta t}{\Delta y}, \qquad 0 \le C_b^t \le 1, \ northernmost \tag{10.157}$$

[(4) Method to use constant phase velocity](4) Method to use constant phase velocity

Only c^* in (3) is used on assumption that a wave always goes away out of the calculation domain. Therefore, phase velocities at all grid points on the lateral boundary plane are the same value. However, a sign becomes reverse between eastern and western sides, and southern and northern sides.

In the case of (3) and (4), the same phase velocity as (2) is used for a vertical velocity component, such as u on the eastern and western planes and v on the southern and northern planes.

By using the phase velocity solved in (1), (2), (3), and (4), the value at a grid point on the boundary at next time step is,

for the variable integrated with small time step $\Delta\tau$,

$$\Phi_b^{\tau+\Delta\tau} = \Phi_b^{\tau} - \frac{C_b^t}{1 - C_b^t} \left(\Phi_{b-1}^t - \Phi_b^{t-\Delta t} \right) \frac{\Delta\tau}{\Delta t} - \gamma \Phi_b^{t-\Delta t} \Delta\tau, \qquad western\ edge\ or\ southern\ edge \tag{10.158}$$

$$\Phi_b^{\tau+\Delta\tau} = \Phi_b^{\tau} + \frac{C_b^t}{1 + C_b^t} \left(\Phi_{b-1}^t - \Phi_b^{t-\Delta t} \right) \frac{\Delta\tau}{\Delta t} - \gamma \Phi_b^{t-\Delta t} \Delta\tau, \qquad eastern\ edge\ or\ northern\ edge \tag{10.159}$$

for the variable integrated with large time step Δt,

$$\Phi_b^{t+\Delta t} = \Phi_b^{t-\Delta t} - \frac{2C_b^t}{1 - C_b^t} \left(\Phi_{b-1}^t - \Phi_b^{t-\Delta t} \right) - \gamma \Phi_b^{t-\Delta t} \Delta t, \qquad westernmostorsouthernmost \tag{10.160}$$

$$\Phi_b^{t+\Delta t} = \Phi_b^{t-\Delta t} + \frac{2C_b^t}{1 + C_b^t} \left(\Phi_{b-1}^t - \Phi_b^{t-\Delta t} \right) - \gamma \Phi_b^{t-\Delta t} \Delta t, \qquad easternmostornorthernmost \tag{10.161}$$

where attenuation constant γ has the dimension of time inverse number. It is thought that the inverse number $1/\gamma$ is suitable more than one hour and less than twelve hours.

10.3.2 Boundary conditions at the top and bottom

Two kinds of a boundary condition, fixed wall (mirror condition) and non-gradient, are applicable to boundary conditions at the top and bottom.

Fixed wall boundary condition

Basically, it can be considered as same as lateral fixed wall boundary condition. However, when we use the terrain-following coordinate system, the relation between the velocity on Cartesian coordinate system and contravariant velocity should be considered as follows.

$$W = (uJ_{31} + vJ_{32} + w) \Big/ G^{\frac{1}{2}} \tag{10.162}$$

Note that vertical velocity W when velocity is 0 on the boundary plane.

On the boundary plane at the top ($k = nz - 1$),

$$W = 0, \qquad w = 0 \tag{10.163}$$

and on the boundary plane at the bottom ($k = 2$),

$$W = 0, \qquad w = -uJ_{31} - vJ32. \tag{10.164}$$

Therefore, boundary condition is given as follows.

$$\begin{array}{ll} u_{i,j,1} = u_{i,j,2}, & u_{i,j,nz-1} = u_{i,j,nz-2} \\ v_{i,j,1} = v_{i,j,2}, & v_{i,j,nz-1} = v_{i,j,nz-2} \\ \phi_{i,j,1} = \phi_{i,j,2}, & \phi_{i,j,nz-1} = \phi_{i,j,nz-2} \\ W_{i,j,2} = 0, \quad W_{i,j,1} = -W_{i,j,3}, & W_{i,j,nz-1} = 0, \quad W_{i,j,nz} = -W_{i,j,nz-2} \end{array} \tag{10.165}$$

where ϕ is the arbitrary scalar. By replacing vertical velocity W at the top and bottom boundaries with the equation (10.162),

on boundary plane at the top ($k = nz - 1$),

$$w_{i,j,nz-1} = 0 \tag{10.166}$$

$$w_{i,j,nz} = -\left(\overline{\overline{u^\zeta} J_{31}}^x + \overline{\overline{v^\zeta} J_{32}}^y + W\overline{G^{\frac{1}{2}}}^\zeta\right)_{i,j,nz-2} \tag{10.167}$$

on the boundary plane at the bottom ($k = 2$),

$$w_{i,j,2} = -\left(\overline{\overline{u^{\zeta}J_{31}}^{x}} + \overline{\overline{v^{\zeta}J_{32}}^{y}}\right)_{i,j,2} \tag{10.168}$$

$$w_{i,j,1} = -\left(\overline{\overline{u^{\zeta}J_{31}}^{x}} + \overline{\overline{v^{\zeta}J_{32}}^{y}} + W\overline{G^{\frac{1}{2}}}^{\zeta}\right)_{i,j,3} \tag{10.169}$$

where pressure perturbation p' on the boundary plane at the bottom is given by extrapolation as follows.

$$p'_{i,j,1} = 2p'_{i,j,2} - p'_{i,j,3} \tag{10.170}$$

Non-gradient boundary condition

Similarly to the lateral boundary,

$$\begin{array}{ll} u_{i,j,1} = u_{i,j,2}, & u_{i,j,nz-1} = u_{i,j,nz-2} \\ v_{i,j,1} = v_{i,j,2}, & v_{i,j,nz-1} = v_{i,j,nz-2} \\ w_{i,j,1} = w_{i,j,2}, & w_{i,j,nz} = w_{i,j,nz-1} \\ \phi_{i,j,1} = \phi_{i,j,2}, & \phi_{i,j,nz-1} = \phi_{i,j,nz-2} \\ W_{i,j,1} = W_{i,j,2}, & W_{i,j,nz} = W_{i,j,nz-1} \end{array} \tag{10.171}$$

where ϕ is the arbitrary scalar.

10.3.3 Sponge layer

In ***CReSS***, the sponge layer set as arbitrary thickness is able to be set up. This is a layer for suppressing and absorbing wave reflections near the boundary plane. An additional term is added to the basic equation as follows.

$$\frac{\partial\left(\rho^{*}\phi\right)}{\partial t} = -\mathrm{Adv}.\phi + ... - \gamma_h\left(x,y\right)\rho^{*}\left(\phi - \phi_e\right) - \gamma_v\left(z\right)\rho^{*}\left(\phi - \phi_e\right) \tag{10.172}$$

where ϕ is an arbitrary forecast variable, ϕ_e is a variable of external data, such as an objective analysis, $\gamma_h\left(x,y\right)$ is an attenuation coefficient which becomes small toward each horizontal boundary plane. $\gamma_v\left(z\right)$ is attenuation coefficient which becomes small toward each vertical boundary plane.

For example, in the case of the west lateral boundary layer which made thickness of sponge layer d in horizontal, $\gamma_h\left(x,y\right)$ is given as follows.

$$\gamma_h = \begin{cases} \alpha_h \left(1 - \dfrac{x}{d}\right)^3, & x \leq d \\ \\ 0, & x > d \end{cases} \tag{10.173}$$

In the case of the bottom boundary layer which made minimum height of sponge layer in vertical, $\gamma_v(z)$ is given as follows.

$$\gamma_v = \begin{cases} 0, & z < z_{low} \\ \\ \alpha_v \left\{1 - cos\left(\pi \dfrac{z - z_{low}}{z_{top} - z_{low}}\right)\right\}, & z \geq z_{low} \end{cases} \tag{10.174}$$

where α_h and α_v are horizontal and vertical attenuation constants, and have dimension of the inverse number of time. These inverse numbers $1/\alpha_h$ and $1/\alpha_v$ is called e-folding time. It is appropriate that these constants are set up in the range about $1/100 \sim 1/300$ s^{-1} . In addition, it is suitable for the thickness of sponge layer to set up in the number of several grid points in horizontal and the thickness of vertical layer about the upper surface to 1/3 in vertical.

This term is calculated at $t - \Delta t$. About the forecast variable calculated at a small time step, it is treated as a value of a big time step $2\Delta t$ like advection term and numerical viscosity term.

The adjusted value to adjust was external data's value ϕ_e, such as objective analysis, in the equation (10.172). However, when a model is performed independently, and there is no objective analysis data for a forecast variable, it can adjust to initial value ϕ_0 or basic state value $\bar{\phi}$ as follows.

$$\frac{\partial(\rho^*\phi)}{\partial t} = -\text{Adv.}\phi + ... - \gamma_h(x,y)\,\rho^*(\phi - \phi_0) - \gamma_v(z)\,\rho^*(\phi - \phi_0) \tag{10.175}$$

$$\frac{\partial(\rho^*\phi)}{\partial t} = -\text{Adv.}\phi + ... - \gamma_h(x,y)\,\rho^*(\phi - \bar{\phi}) - \gamma_v(z)\,\rho^*(\phi - \bar{\phi}) \tag{10.176}$$

Chapter 11

Application of Parallel Processing

In this section, we will describe the computation method of ***CReSS*** with a parallel computing.

The parallel program of ***CReSS*** is fundamentally designed for running on a multiple memory type parallel computer. The parallel program is written with Message Passing Interface (MPI). ***CReSS*** is available to almost all parallel computers, because the MPI is a defactor standard library. Using some workstations, ***CReSS*** can also be performed in a cluster environment.

11.1 Technique for parallelization

11.1.1 Two-dimensional domain decomposition

CReSS is fundamentally designed for a multiple memory type parallel computer. A parallel program is available for not only large multiple memory type parallel computers but also cluster systems of workstations and PC-UNIX computers.

It is easy to use the parallel processing by adopting two-dimensional domain decomposition in horizontal, because a spacial difference in horizontal is explicitly calculated in ***CReSS***. Various physical processes, such as cloud microphysics, can be fundamentally calculated only with the physical quantities in the caluculating cell, there are some references between upper and lower cells. A z^* coordinate system (ζ system) makes parallelization easy that a coordinate system.

Firstly, we show the advantage of parallel processing by adopting two-dimensional domain decomposition in horizontal.

- It is easy to establish the boundary condition, because coordination system is z^*-system (ζ-system) has no loss of data near the surface.
- It is easy to understand the technique of parallelization intuitively.
- It is available to use the same function on nodes except the node included the boundary condition.
- The load balance is high. (That is, a waiting time for other node's computation would be short.)
- Parallelization efficiency for a calculation with a number of grid points would be high.
- As a calculation domain for each node is larger, data communication between each node is relatively smaller.

We show a schematic illustration of two-dimensional domain devomposition as follows.

Figure 11.1 shows a schematic illustration of the parallel computing in the case of calculating with the second-order precision. In the case of adopting central difference approximation with the second order precision, six grid points must be referred for a certain grid point. Considering the horizontal two-dimensional domain decomposition, there is no problem of reference in the vartical direction. In horizontal ,however, when the approximate solution of the grid point on the boundary of decomposition is calculated,the reference between nodes is needed. Then showing the enlarged figure on right side, we set the halo region in each node, and it is enable to obtain the same result as by one node on the computational domain computed by decomposed nodes had the halo region. For that purpose, communication between the nodes are performed only for getting the essential data from neighboring the node whenever the value in the halo region is needed (The detail is shown in subsection 11.1.2)

In addition, ***CReSS*** is available to compute with fourth-order precision. For that purpose, it is necessary to communicate much the value which is needed by doubling a halo region like the case of second order precision. It is possible to change the precision of the differential equation by the same method, for example, sixth-order, eighth-order, ... and N-th-order.

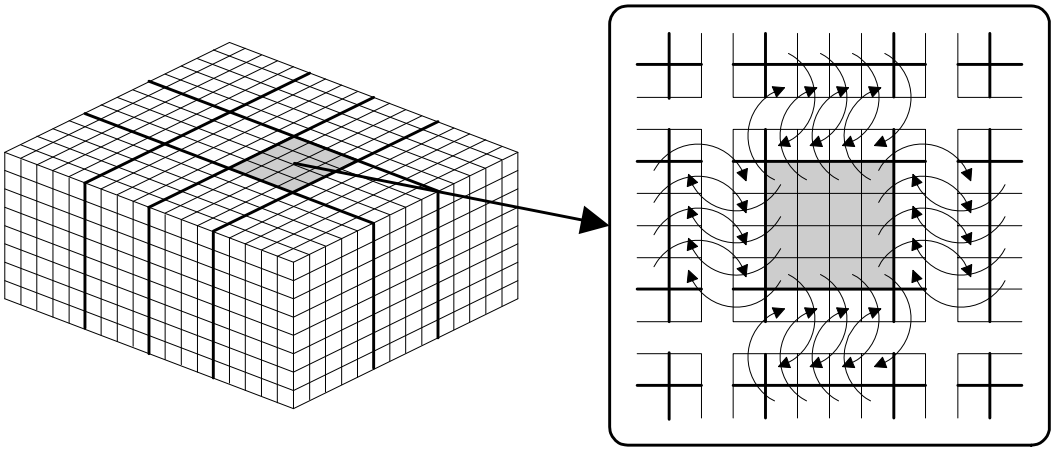

Figure 11.1. Schematic illustration of two-dimensional domain decomposition and exchange of the value in the halo region

11.1.2 Example of parallel computing

In this subsection, we describe the communication between halo regions explained in subsection 11.1.1.

The source code is written with Message Passing Interface (MPI). The MPI is the interface specifications for the call form of various communication routines required for a multiple memory type parallel computer and the function and subroutines of C and Fortran. But it is interface specification to the last, and the concrete application method is not determined.

As for the details about MPI, it is good to refer to the following homepages.

http://www.mpi-forum.org

For communicating by MPI such as figure 11.1, it is essential to repeat a shift in all directions except the boundary plane. Figure 11.2 is the example of a shift to the halo region of east side from the halo region of west side except the boundary plane. In this figure, *nipe* is numbers of node in the east-west direction and *ipe* is node number.

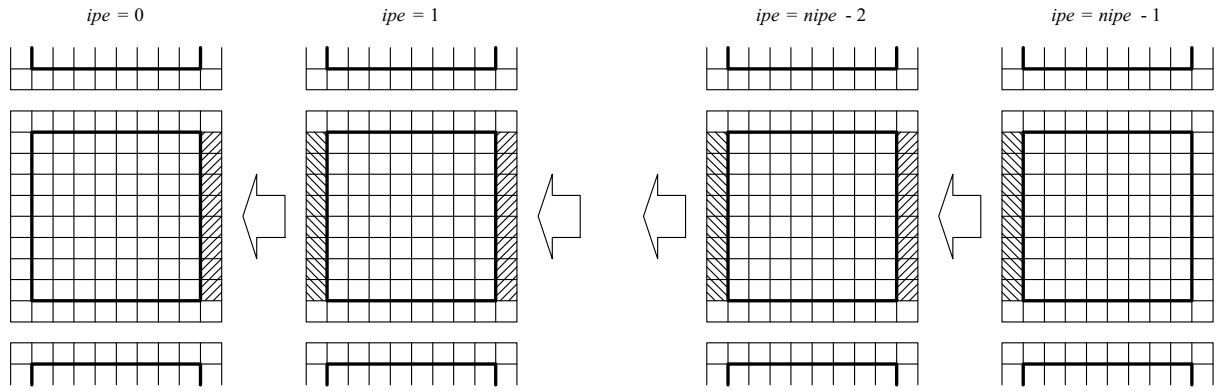

Figure 11.2. This is a example of a shift to the halo region of east side from the halo region of west side.

The source code of the MPI parts in ***CReSS***is shown as following. The part of it is shown here, so for understanding the detail it is necessary to refer `exchansn.f` and `exchanwe.f` in the directory `Src`.

```
! A node at the westernmost end has no transmitting buffer
if(ipe.eq.0) then
  dst=mpi_proc_null
end if

! A node at the easternmost end has no transmitting buffer
if(ipe.eq.nipe-1) then
  src=mpi_proc_null
end if

! A setup of transmitting buffers other than a node at the westernmost end
if(ipe.ne.0) then

  do 140 k=kbmin,kbmax
  do 140 j=jbmin,jbmax
    ib=(k-kbmin)*jbmxn1-jbmin1+j

    sbuf(ib)=var(iwsend,j,k)

140   continue

end if

! The call of the MPI routine for transmission and reception
call mpi_sendrecv(sbuf,siz,mpi_real,dst,tag,rbuf,siz,mpi_real,src,
```

```
     .                    tag,mpi_comm_world,stat,ierr)

      ! A setup of receiving buffers other than a node at the easternmost end
      if(ipe.ne.nipe-1) then

        do 170 k=kbmin,kbmax
        do 170 j=jbmin,jbmax
          ib=(k-kbmin)*jbmxn1-jbmin1+j

          var(ierecv,j,k)=rbuf(ib)

170     continue

      end if
```

Although the shift is mounted by `mpi_sendrecv` here, it is better not to mount this by `mpi_send` and `mpi_recv`. Although it is not impossible, various programming can be considered and may carry out a deadlock depending on the case. Moreover, by the formal specification of the MPI, the guarantee of operation of a program will be impossible, since it seems that it has not defined whether a deadlock is carried out by this or it does not carry out.

11.2 Test of the parallel program

11.2.1 Inspection of coincidence of a calculation result

Since parallel programming adopted in ***CReSS*** should just do a shift fundamentally shown in the previous section, the result performed using several nodes must be completely same as the result performed using one node (in calculation for the whole average value, it is not necessarily to get complete agreement. Since it does not apply operation to the value in shifting, disagreement cannot happen.). We checked the agreement by ***CReSS*** as follows.

When several nodes are used for calculation, a result is outputted from every node.

ex. : *exprim*.dmp*xxxxxx*.pe0000.bin ～ *exprim*.dmp*xxxxxx*.pe*yyyy*.bin

For comparing to the result by one node and that by several nodes, we gather the above data files up to one file. (A post processor "unite" can gather the output files up to a binary unformated file with direct access.) The file was compared with the result of calculation with one node. If the files of both one node and several nodes are text format, we can confirm their complete agreement as the following command.

ex. : % diff *exprim*.dmp*xxxxxx*.united.txt *exprim*.dmp*xxxxxx*.txt

As a result of executing this command, if nothing happens (0 byte of file will be made if a file

is outputted to a standard input/output), the result must completely be in agreement. *exprim* of file name is an experiment name (see section 4.1).

We perform this verification to various experiments with ***CReSS***, and there is no problem in the parallelization.

11.2.2 Efficiency of parallel processing

Finally, the efficiency of parallel processing is shown by a comparison of calculations with the various numbers of processing elements (PEs). The performance of parallel processing of ***CReSS*** was tested by a simulation which had $67 \times 67 \times 35$ grid points and which was integrated for 50 steps on HITACHI SR2201.

As increase of the numbers of PEs, the calculation time decreased almost linearly (Fig. 11.3). The results of the test showed a sufficiently high performance of the parallel computing of ***CReSS***. In figure 11.4, it can see that the efficiency of parallel processing is falling little by little. When the number of PEs was 32, the efficiency decreased significantly. Because the number of grid points was too small to use the 32 PEs, the communication between PEs became relatively large for the present verification setting. This result of the verification shows that the efficiency of parallel processing of CReSS is higher when each PE shares larger number of grid points. The calculation with a lot of PEs is more useful for larger calculation.

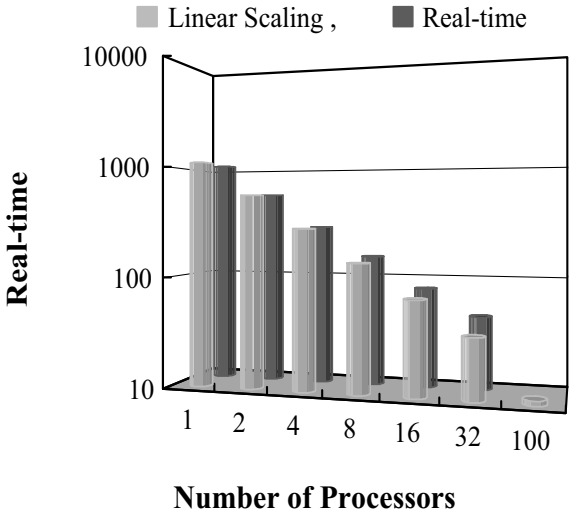

Figure 11.3. Time for parallel processing from a test experiment [s]

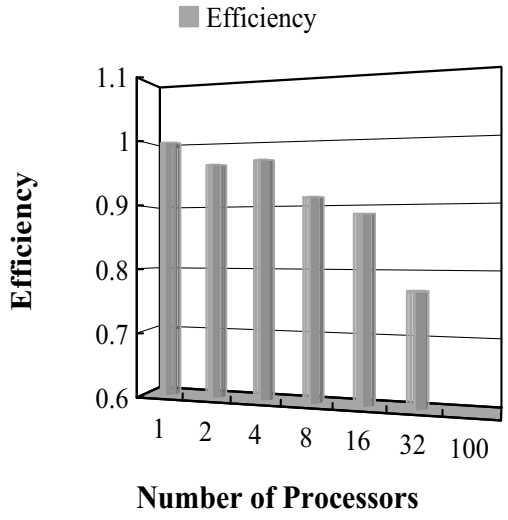

Figure 11.4. Efficiency of parallel processing from a test experiment

However, there is a problem. Especially, it is becoming more remarkable in the case of using the option for cloud microphysics. Because the program for cloud microphysics has so many `if` sentences. For example, when a node without clouds must wait for the processing of the node with clouds, efficiency would be small. In the present condition, ***CReSS*** does not have advanced parallel processing changed from the first setup to the more efficient setup in the calculation.

The values of efficiency in the graph is somewhat wavy in figure 11.4, since it is based on the

difference between the ways of a decomposition on two-dimensional domain decomposition. (For example, when a calculation is performed with four nodes, there is the ways of dividing of 1×4, 2×2, or 4×1.) However, it is sufficient for seeing the tendency of the efficiency of parallel processing.

References

Asai, T., 1972: Thermal instability of a shear flow turning the direction with height. *J. Meteor. Soc. Japan*, 50, 525-532.

Cotton, W. R., G. J. Tripoli, R. M. Rauber and E. A. Mulvihill, 1986: Numerical simulation of the effects of varying ice crystal nucleation rates and aggregation processes on orographic snowfall. *J. Climate Appl. Meteor.*, 25, 1658-1680.

Grasso, L. D., and W. R. Cotton, 1995: Numerical simulation of a tornado vortex. *J. Atmos. Sci.*, **52**, 1192–1203.

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

Ikawa, M. and K. Saito, 1991: Description of a nonhydrostatic model developed at the Forecast Research Department of the MRI. *Technical Report of the MRI*, 28, 238pp.

Klaassen, G. P. and W. R. Peltier, 1985: The evolution of finite amplitude Kelvin-Helmholtz billows in two spatial dimensions. *J. Atmos. Sci.*, **42**, 1321–1339.

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

Klemp, J. B., and R. Rotunno, 1983: A study of the tornadic region within a supercell thunderstorm. *J. Atmos. Sci.*, **40**, 359–377.

Lin, Y. L., R. D. Farley and H. D. Orville, 1983: Bulk parameterization of the snow field in a cloud model. *J. Climate Appl. Meteor.*, 22, 1065-1092.

Liu. A. Q., G. W. K. Moore, K. Tsuboki and I. A. Renfrew, 2006: The effect of the sea-ice zone on the development of boundary layer roll clouds during cold air outbreaks. *Boundary-Layer Meteorology*, Vol.118, No.3, 557–581

Louis, J. F., M. Tiedtke and J. F. Geleyn, 1981: A short history of the operational PBL parameterization at ECMWF. *Workshop on Planetary Boundary Layer Parameterization*, 25-27 Nov. 1981, 59-79.

Maesaka, T., G. W. Kent Moore, Liu, A. Q., and K. Tsuboki, 2006: A simulation of a lake effect snowstorm with a cloud resolving numerical model. *Geophysical Research Letters*, Vol.33, L20813.

Murakami, M., 1990: Numerical modeling of dynamical and microphysical evolution of an isolated convective cloud - The 19 July 1981 CCOPE cloud. *J. Meteor. Soc. Japan*, 68, 107-128.

Murakami, M., T. L. Clark and W. D. Hall 1994: Numerical simulations of convective snow clouds over the Sea of Japan; Two-dimensional simulations of mixed layer development and convective snow cloud formation. *J. Meteor. Soc. Japan*, 72, 43-62.

Orlanski, I., 1976: A simple boundary condition for unbounded hyperbolic flows. *J. Comput. Phys.*, **21**, 251–269.

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

Tsuboki, K. and A. Sakakibara, 2001: CReSS User's Guide 2nd Edition, 210p.

Tsuboki, K. and A. Sakakibara, 2002: Large-scale parallel computing of Cloud Resolving Storm Simulator. *High Performance Computing, Springer*, H. P. Zima et al. Eds, 243-259.

Wicker, L. J., and R. B. Wilhelmson, 1995: Simulation and analysis of tornado development and decay within a three-dimensional supercell thunderstorm. *J. Atmos. Sci.*, **52**, 2675–2703.

Weisman, M. L., and J. B. Klemp, 1982: The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. *Mon. Wea. Rev.*, **110**, 504–520.

Weisman, M. L., and J. B. Klemp, 1984: The structure and classification of numerically simulated convective storms in directionally varying wind shears. *Mon. Wea. Rev.*, **112**, 2478–2498.

Wilhelmson, R. B., and J. B. Klemp, 1978 A numerical study of storm splitting that leads to long-lived storms. *J. Atmos. Sci.*, **35**, 1974–1986.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro and K. Brewster, 1995: Advanced Regional Prediction System, Version 4.0. *Center for Analysis and Prediction of Storms, University of Oklahoma*, 380pp.

Outline of the IHP Training Course in 2007

Outline

As a part of the Japanese contribution to the International Hydrological Programme (IHP), a short course on numerical prediction of high-impact weather systems will be conducted for participants from the Asia-Pacific regions. The course will be held for the period from 2 to 15 December 2007 at Hydrospheric Atmospheric Research Center (HyARC) and Information Technology Center, Nagoya University, Nagoya, Japan. The course includes a series of lectures in English, practical sessions and technical tours.

Objectives

High-impact weather systems are most significant phenomena in the atmosphere and occasionally cause huge disasters to human society. Understanding their mechanisms and structures is necessary for prediction and prevention/reduction of disasters. One of the most important objectives of numerical models is high-resolution simulation of high-impact weather systems for detailed studies and accurate predictions. In the 17th IHP training course, the participants will study characteristics and performance of the state-of-the-art numerical models such as regional and global cloud resolving models.

In order to perform simulations and numerical experiments of the high-impact weather systems, HyARC is developing a cloud-resolving numerical model named “the Cloud Resolving Storm Simulator” (CReSS). The participants will have practice to perform some simulation experiments of weather systems to understand how to use the cloud-resolving model. High-resolution numerical simulation using the CReSS model will clarify a detailed structure of the high-impact weather systems and make a quantitative prediction of the associated precipitation.

Course Contents (convener: K. Tsuboki)

Lecturers

K. Tsuboki (Hydrospheric Atmospheric Research Center, Nagoya University)
T. Shinoda (Hydrospheric Atmospheric Research Center, Nagoya University)
T. Nasuno (Frontier Research Center for Global Change)
T. Maesaka (National Research Institute for Earth Science and Disaster Prevention)
M. Yoshizaki (Institute of Observational Research for Global Change)
M. Sato (Center for Climate System Research, University of Tokyo)
T. Nagai (Information Technology Center, Nagoya University)
K. Saito (Meteorological Research Institute, Japan Meteorological Agency)
T. Aoki (Tokyo Institute of Technology)
K. Tanaka (Disaster Prevention Research Institute, Kyoto University)
W. Ofuchi (The Earth Simulator Center)
A. Sakakibara (Chuden CTI Co.,Ltd.)

List of Participants

1) UNESCO-IHP Participants

Mr. Fierra Setyawan Sastropawiro (Indonesia)

Affiliation: Meteorological and Geophysical Agency, Research and Development Center

Address: Jl. Angkasa I/2, Kemayoran, Jakarta 10720, Indonesia

Ms. Elza Surmaini (Indonesia)

Affiliation: Indonesian Agroclimate & Hydrology Research Institute, Indonesian Center for Agricultural Land Resources Research and Developmet, Ministry of Agriculture

Address: Jl. Tentara Pelajar no. 1A, Bogor 16111, West Java, Indonesia

Ms. Sinthaly Chanthana (Lao. P.D.R.)

Affiliation: Ministry of Agriculture and Forestry, Department of Meteorology and Hydrology (DMH)

Address: Bane Arkarth, Luangprabang Road, Sikhottabong District, Vientiane Capital, Lao PDR

Mr. Sazali Bin Osman (Malaysia)

Affiliation: Division of Hydrology and Water Resources, Department of Irrigation & Drainage

Address: KM 7, Jl. Ampang, 68000 Ampang, Kuala Lumpur, Malaysia

Mr. Tirtha Raj Adhikari (Nepal)

Affiliation: Central Department of Hydrology and Meteorology, Tribhuvan University

Address: Kirtipur, Kathmandu, Nepal

Ms. Thanh Thi Nguyen (Vietnam)

Affiliation: Vietnam Institute of Meteorology, Hydrology and Environment; Center for Marine and Ocean - Atmosphere Interaction Research

Address: 5/62 Nguyen Chi Than, Hanoi, Viet Nam

Ms. Josefina Calapan Argete (Philippines)

Affiliation: Institute of Environmental Science & Meteorology College of Science, University of the Philippines

Address: Diliman, Quezon City, Philippines

Ms. Sirithorn Noiduang (Thailand)

Affiliation: Dpartment of Water Resources, Bureau of Research Development and Hydrology

Address: 180/3 Rama VIRd., Samsanenai, Payathai District, Bangkok, 10400, Thailand

2) Participants supported by other funds

Mr. Liew Ju Neng (Malaysia)

Affiliation: School of Environmental and Natural Resource Sciences, Faculty of Science and Technology, University Kevangsaan Malaysia

Address: 43600 Ukm Bangi, Selangor , Malaysia

Mr. Palikone Thalongsengchanh (Lao. P.D.R.)

Affiliation: Graduate School of Science Kyoto University

Address: Oiwake-Cho, Kitashirakawa, Sakyo-Ku, Kyoto 606-8502 Japan

3) Participants from Graduate School of Environmental Studies, Nagoya University

Mr. Uddin Md. Rafi (Bangladesh)

Affiliation: Graduate School of Environmental Studies, Nagoya University, Japan

Address: Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

Ms. Singh Prasamsa (Nepal)

Affiliation: Graduate School of Environmental Studies, Nagoya University, Japan

Address: Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

Ms. Nasreen Akter (Bangladesh)

Affiliation: Graduate School of Environmental Studies, Nagoya University, Japan

Address: Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

Mr. Ningning Zhang (China)

Affiliation: Graduate School of Environmental Studies, Nagoya University, Japan

Address: Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

Lectures

L0 Introduction ····· K. Tsuboki
- Guidance and outline of the IHP short course

L1 Mesoscale meteorology and numerical models ····· M. Yoshizaki
- Introductory lecture of meso-scale meteorology and overview of non-hydrostatic numerical model

L2 Computational Fluid Dynamics ····· T. Aoki
- Basics of computational fluid dynamics and environmental simulation
- Visualization of computational fluid dynamics

L3 Main frame computer in Nagoya University ····· T. Nagai
- Computer system at the Information Technology Center, Nagoya University
- Basics of UNIX and usage of the main frame computer

L4 Cloud resolving model ····· K. Tsuboki
- Dynamic process and physical process of a cloud resolving model
- Introduction of the Cloud Resolving Storm Simulator (CReSS)

L5 Parallel processing of the cloud resolving model ····· A. Sakakibara
- Parallel computation using the Massage Passing Interface
- OpenMP

L6 The non-hydrostatic model of the Japan Meteorological Agency ····· K. Saito
- Introduction of the JMAMRI non-hydrostatic model
- Experimental and operational utilization of the NHM

L7 Surface process ····· K. Tanaka
- Surface process model, SiBUC
- Coupling the surface model and the cloud resolving mode

L8 Planetary boundary layer and convective clouds ····· T. Shinoda
- Simulation and observation of the planetary boundary layer
- Interaction of PBL and convective clouds

L9 AGCM for the Earth Simulator ····· W. Ofuchi
- Introduction of the AFES (the AGCM for the Earth Simulator)
- Simulation experiments using the AFES model

L10 Global non-hydrostatic model ····· M. Sato
- Introduction of the global non-hydrostatic model, NICAM
- Simulation experiments using NICAM

L11 Typhoon simulation ····· T. Nasuno
- Overview and recent researches of typhoon simulations

L 12 Radar observation and cloud resolving model····· T. Maesaka

- Radar observation of severe storms
- Data assimilation of radar to the cloud resolving model

Practices

P1 Basic usage of the CReSS model

- Numerical experiments of cats-eye, mountain waves and supercells

P2 Simulation experiments of high impact weather systems

- Numerical experiments of typhoons, heavy rainfall, snow storms and so on

Technical Tours

T1 The Earth Simulator Center in Yokohama

T2 The Japan Meteorological Agency in Tokyo

Schedule (2-15 December 2007)

Date	Description	City
2 (Sunday) December 2007	Arrival at Nagoya	Nagoya
3 (Monday)	10:00-12:00: Guidance (Lecture 0) by K. Tsuboki and Lecture 1 by M. Yoshizaki 14:00-16:00: Lecture 3 by T.Nagai Reception at Nagoya University in the eveneing	Nagoya
4 (Tuesday)	10:00-12:00: Lecture 7 by K. Tanaka 14:00-16:00: Lecture 8 by T. Shinoda	Nagoya
5 (Wednesday)	10:00-12:00: Lecture 4 by K. Tsuboki 14:00-16:00: Lecture 5 by A. Sakakibara	Nagoya
6 (Thursday)	10:00-12:00 Lecture 9 by W. Ofuchi 14:00 -: Practice 1	Nagoya
7 (Friday)	10:00-12:00: Lecture 6 by K. Saito 14:00-: Practice 2	Nagoya
8 (Saturday)	Move to Tokyo	Tokyo
9 (Sunday)	Japanese Culture Introduction and Free Time	Tokyo
10 (Monday)	Technical Tour 1: The Japan Meteorologycal Agency in Tokyo (Move to Yokohama) Technical Tour 2: The Earth Simulation Center in Yokohama (Move to Nagoya)	Nagoya
11 (Tuesday)	10:00-12:00: Lecture 2 by T. Aoki 14:00-: Practice 2 (continued)	Nagoya
12 (Wednesday)	10:00-12:00: Lecture 11 by T. Nasuno 14:00-: Practice 2 (continued)	Nagoya
13 (Thursday)	10:00-12:00: Lecture 10 by M. Sato 14:00-: Practice 2 (continued)	Nagoya
14 (Friday)	10:00-12:00: Lecture 12 by T. Maesaka 14:00-: Closing Ceremony	Nagoya
15 (Saturday)	Departure from Nagoya	