Using the UCLA Large Eddy Simulation code

Thijs Heus, Bjorn Stevens, Axel Seifert, Cathy Hohenegger

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Overview of UCLA LES
UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut
für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

This Week

<table>
<tr><th></th><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th></tr>
<tr><td>9.30</td><td>ClimateServiceCenter 112
Bjorn - Introduction
Thijs - Setting Up</td><td>ZMAW 301
Cathy – Surface
Thijs - Statistics</td><td rowspan="3">Geomatikum 13.35
Thijs – Dynamics I
Practical: Own topics</td><td rowspan="2">Geomatikum 13.35
Axel - Microphysics
Practical: Own topics</td><td>ZMAW 301
Thijs – Dynamics II
Topics of interest</td></tr>
<tr><td>13.00</td><td>ClimateServiceCenter 112
Thijs – Code Structure</td><td rowspan="2">ClimateServiceCenter 112
Executing the code and building a case.</td><td rowspan="2">ZMAW 024
Practical: Own topics</td></tr>
<tr><td>15.00</td><td>ZMAW 301
Thijs – In- and Output</td><td>Geomatikum 13.35
Bjorn - Radiation</td></tr>
<tr><td>18.00</td><td>Icebreaker/Beer/...</td><td></td><td></td><td>Dinner</td><td></td></tr>
</table>

Our Group

- Hans-Ertel Zentrum for research on Clouds and Convection
- Led by Cathy Hohenegger and Axel Seifert
- Funded by Deutscher Wetter Dienst
- Hunt for knowledge on convective clouds in various conditions
- Large Eddy Simulations are our primary (but not only) tool

Cascade of Models

- General Circulation Models
- Regional Models
- Large-Eddy Simulations
- Direct Numerical Simulations

Cascade of Models

General Circulation Models

- Domain size: Entire Earth
- Horizontal Boundary conditions: None
- Horizontal grid spacing: $50\mathrm{km}$
- Total number of points: about $400 \times 400 \times 100$
- Simulation duration: Weeks - millennia
- Resolved: Hadley Circulation, fronts, ...
- Parameterized: Clouds, Boundary layers, Surface, Microphysics

- Domain size: Continental scale or smaller
- Studies of organization, deep systems,...
- Horizontal Boundary conditions: Nested/forced by GCM
- Horizontal grid spacing: $5\mathrm{km}$
- Total number of points: about $400 \times 400 \times 100$
- Simulation duration: Weeks
- Resolved: Deep clouds
- Parameterized: Shallow Clouds, Boundary layers, Surface, Microphysics

Large-Eddy Simulations

- Domain size: $1 - 100\mathrm{km}$
- Studies of boundary layer processes, idealized (and not so idealized) clouds
- Horizontal Boundary conditions: Periodic
- Horizontal grid spacing: $50\mathrm{m}$
- Total number of points: about $400 \times 400 \times 100$
- Simulation duration: Hours/Days
- Resolved: Shallow Clouds, Boundary layers
- Parameterized: Turbulence, Surface, Microphysics

- Domain size:$1\mathrm{m}$
- Studies of turbulence, possibly with interactions of other processes
- Horizontal Boundary conditions: Periodic
- Horizontal grid spacing: $1\mathrm{mm}$
- Total number of points: about $1000 \times 1000 \times 1000$
- Simulation duration: Minutes
- Resolved: Turbulence, surface (?)
- Parameterized: Microphysics

Focus of LES is on Geophysical *Fluid Dynamics*

Many processes are still unresolved or beyond the scope of LES:

- Radiation - At best, 2D radiation is available
- Chemistry, aerosols and microphysics
- Near-Surface processes

Large-Eddy Simulations

Principle

- Spatially filter (smooth) the Navier Stokes Equations
- Ensure that the width of this spatial filter lies in the inertial subrange of the turbulent field
- Explicitly solve the most energetic scales
- Model the Sub Filter Scale (SFS) turbulence. The details of this SFS model should not matter.

We violate these principles on a daily basis. But still, over 90% of the energy in the bulk of the convective boundary layer is usually resolved.

Filtering

$$\bar{u} = \int G(r) u \mathrm{dr}$$

With G the filter (could be a (grid-)box, a gaussian, a spectral filter,....)

Navier Stokes Equations

$$\frac{\partial u_i}{\partial t} = \qquad -u_j\frac{\partial u_i}{\partial x_j} - c_p\Theta_0\frac{\partial \pi}{\partial x_i} \qquad + \nu\frac{\partial^2 u_i}{\partial x_j^2} + \mathcal{F}_i$$

Large-Eddy Equations

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i$$

$$\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \gamma_{\phi j})}{\partial x_j} + \mathcal{S}_\phi$$

Anelastic continuity

$$\frac{\partial(\rho_0 u_i)}{\partial x_i} = 0$$

Ideal gas law equation of state

$$\theta_v = \theta \left(1 + (R_v/R_d - 1)q_t - (R_v/R_d)q_l\right).$$

Closure

- $\tau_{ij} \equiv \overline{u_i u_j} - \bar{u}_i \bar{u}_j$ is the Sub Filter Scale flux and needs to be modeled
- Can be done by
  - Smagorinsky diagnostic closure
  - Deardorff prognostic TKE
  - Higher order closures
  - Nothing at all (Numerical diffusion)
- All models start off with models for homogeneous isotropic turbulence
- Empirical modifications are nearly always done to match stable turbulence and condensation gradients.

- Dry LES: Smagorinsky (1963), Lilly(1967), Deardorff(1972)
- Cloudy LES: Sommeria(1976)
- 'Big breakthrough LES': Schmidt and Schumann (1989)
- 'Huge breakthrough LES': Earth Simulator Global LES (2001)

History I

Intercomparisons

- Dry CBL: Nieuwstadt et al. (1986, 1993) and Andren et al. (1994)
- Non-Precip Stratocumlus: Moeng et al. (1996)
- Radiative Smoke: Bretherton et al. (1999)
- Non-Precip Shallow Cu: Siebesma et al. (2003)
- Non-Precip Stratocumlus: Stevens et al. (2001)
- Diurnal Cycle Cu: Brown et al. (2001)
- Sheared and Stable BLs: Holtslag(2006), Beare(2006)
- Precip Stratocumlus: Ackerman et al. (2008)
- Precip Cumlus: van Zanten et al. (2011)
- Precip Stratocumlus: Ackerman et al. (2008)
- Radiative, transition runs: Sandu, de Roode, Blossey (2012)

History
UCLALES

- Based on a meso-scale modeling code by prof. Cotton and prof. Pielke at Colorado State University (eighties, nineties)
- Started as LES by Bjorn in the nineties
- Blossomed with him at UCLA (hence the name)
- Parallelized by Jim Edwards, Microphysics with help of Graham Feingold and Axel Seifert, dynamics by Verica Savic Jovcic
- Participated in all GCSS intercomparisons, and in many process studies

When *not* to use LES

When your problem has ...

- ... nothing to do with turbulence
- ... exclusively to do with turbulence (use DNS!)
- ... is dominated by larger scales (e.g. frontal systems)

Or when you don't have sufficient computer power to do high resolution simulations. In which case, start doing theory.

When not (yet) to use *UCLA LES*

When your problem has ...

- ... strong pressure fluctuations (anelastic approximation is used)
- ... orography, heterogeneous surface conditions or land-atmosphere interactions
- ... has an important lateral component to it (Periodic boundary conditins)

Or when you're not willing to look into the code.

What *can* be done with (UCLA) LES

Classical studies

- Clear convective boundary layers
- Shallow cumulus clouds
- Stratocumulus clouds

What *can* be done with (UCLA) LES

Modern studies

- Precipitation and microphysics
- Cloud and parcel tracking
- Deep convection
- Stable boundary layers
- Surface interaction
- Day-to-day runs like in the KNMI Testbed

Model Philosophy

Why use stand-alone LES models at all?

- Research desires ad-hoc changes
- Big model structures (WRF, ECHAM, ICON...) tend to be cluttered, lots of unnecessary additions, hard to run and compile, unreadable,...
- UCLALES is just small enough to understand (more or less)
- It is easy to code any forcing/output you want, and use it for 1 study
- Optimized for user/developer time, not CPU Time

Course Aim

After this course, you should...

- Be able to run and tweak the model
- Know where to look up scripts and examples (including in these handouts)
- Understand the (im-)possibilities and sensitivities of UCLA LES
- Have a feel for what resolution should be used when, and what model setting is necessary.

Drinks tonight?

Hadley's, 18.00?

Setting up the code: Obtaining, compiling, running (and version management)

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Git version management

- Git is a distributed version management system
- All history of all branches is captured
- Easy to create branches for some project (like the course)
- Easy to merge fixes and features from branch to branch
- The main repository sits on `www.gitorious.org/uclales`
- The `master` branch should always be the most stable, up-to-date branch

Gitorious.org

- Register on `www.gitorious.org` (already done?)
- Tell me your username there, to give you (write) access to UCLALES
- Login at `www.gitorious.org`
- Go to “Manage SSH keys”
- Go to “Add SSH key”
- Add the contents of `key/id_rsa.pub` (or `~/.ssh/id_rsa.pub`) and click OK
- Take some time to browse through the website

- In your course directory, download the code with `git clone git@gitorious.org:uclales/uclales.git`
- `cd uclales; ls`
- The entire history is now local in your folder
- `git branch -a` shows all branches
- By default, you are on the master branch

Using Git

Switching branches

- The start off point for your code is the master branch, so go there if you're not already on it: `git checkout master`
- Now make your personal branch, based on the course branch: `git checkout -b yourname`
- Here you can play whatever you like

- Open the file `test1`
- Write something in it
- See what is different: `git status` and `git diff`
- If you are happy with you change, commit: `git commit test1` or `git commit -a` for all changes
- Write a commit message and save
- See what is different now: `git diff`
- Nothing!

- Open the new file `test2`
- Write something in it
- See what is different: `git status` and `git diff`
- You have to add the file with `git add test2`
- If you are happy with you change, commit: `git commit test1` or `git commit -a` for all changes
- Write a commit message and save
- See what is different now: `git diff`
- Nothing!

- On `gitorious.org`, nothing has changed yet
- To update: `git push origin yourname`
- Refresh `gitorious.org`; many new branches
- To get them all: `git pull`
- `git branch -a` has more branches now

Other commands

- `git rm filename` and `git mv filename` (Re)move files
- `git merge branchname` merges `branchname` into the current branch
- `git checkout -f filename` resets a single file to whatever was committed
- `git reset` is the panic button and reverts everything to the previous state
- See `uclales/doc/git_uclales.pdf` for longer explanation

UCLALES requires almost no outside libraries.

- NetCDF (v3 or later) for input and output
- MPI (Only if you want to do Parallel runs)
- A Fortran 95 compiler (IFort, gfortran, xlf work)
- Git for keeping up to date with the source code
- CMake (optional) for easier/faster compilation

On thunder, load cmake, Ifort and mpi with:

```
module load cmake
module load intel13.0.0
module load openmpi_ib1.6.2-static-intel13
```

There are two ways of compiling the code.

- CMake does its best to create a Makefile automatically.
  - Allows for parallel compilation
  - Easier to maintain
  - Not on every system
- A bunch of predefined Makefiles are available in the `misc/makefiles` directory.

Compilation I

CMake

- The `CMakeLists.txt` file in the `uclales` dir sets all the options, searches for libraries etc.
- Overrides can be set on the commandline or in a configuration file
- Choose/edit a configuration file in `uclales/config`. This sets paths to libraries
- For now, just copy the thunder one to default:
  `cp thunder.cmake default.cmake`
- Create a build directory
  `mkdir build; cd build` from the `uclales` dir
- Run CMake to create the makefile: `cmake -D MPI=FALSE ..`
- `make -j4` to build the binary `uclales`
- Executing `./uclales` gives an error now: Missing `NAMELIST`

CMake responds to a number of commandline options, case sensitive, always with -D as a flag

Variable	Values	
MPI	TRUE, FALSE	Switch between parallel and serial
CMAKE_BUILD_TYPE	DEBUG, RELEASE	Switch between debug settings and optimized
PROFILER	GPROF, SCALASCA, MARMOT	Switch on profiler (to assess speed bottleneck)

Executing

- Copy the executable `uclales` to the `run` directory
- We need a runscript (`uclales/misc/jobscripts/runscript_course_seq`)
- We need a `NAMELIST` (`uclales/misc/initfiles/namelist_drycbl`)
- Submit it: `qsub runscript_course_seq`
- Wait...
- See what happens with: `tail -f output`

Model Options

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Starting a model run

There are four ingredients that feed into the model

- Hardcoded options
- Restart files (in NetCDF format)
- Data files (in text format)
- An option file: NAMELIST

Available runs I

In `misc/initfiles` the following cases are provided by default:

- **namelist_astex**: The Astex case.
- **namelist_cumulus**: Namelist to reproduce the idealized cumulus cases reported in Stevens, JAS (2007). Requires the generation of a sound_in file with bstate.f95.
- **namelist_drycbl**: Idealized dry CBL consisting of a layer with initially uniform stratification and constant forcing.
- **namelist_dycm01**: The DYCOMS GCSS RF01 case, requires the generation of a sound_in file with bstate.f95.
- **namelist_dycm02**: The DYCOMS GCSS RF02 case, requires the generation of a sound_in file with bstate.f95, as well as the generation of zm_grid_in and zt_grid_in files uzing zgrid.gcss9.f.
- **namelist_rico**: The RICO GCSS composite case.
- **namelist_smoke**: The GCSS smoke case.

Data files

*_grid_in

- `zm_grid_in`, `zt_grid_in` Input files for vertical non-equidistant grids that are not possible with the namelist options. A single column of values, needs to have at least nzp-2 points

sound_in

- A completely flexible input file for the initial profiles of the mean quantities
- Textfile with a bunch of rows:
  - height in meters or in pressure (depending on `ipsflg`) The first number is the surface pressure
  - Temperature. Depending on `itsflg`, the absolute temperature, potential temperature or liquid water potential temperature.
  - Humidity. Depending on `irsflg`, the relative humidity or total humidity
  - Horizontal velocity fields, u and v.

The file contents should cover the entire domain. Between anchor points, linear interpolation happens.

Time dependent fluxes and large scale forcings.

- The first block sets the surface values, with columns:
  - Time in seconds
  - Surface heat flux in Wm^{-2}
  - Surface moisture flux in Wm^{-2}
  - Surface liquid water potential temperature
  - Surface pressure
- From the second block on, every block starts with: `# time`
- Within each block, the following columns show up:
  - Large scale subsidence w_s, gives the tendency $-w_s\frac{\partial\phi}{\partial z}$
  - Large scale tendency for θ_l
  - Large scale tendency for q_t

The block contents should cover the entire domain. Between anchor points, linear interpolation happens.

Nudges the average fields to a preset value:

$$\left.\frac{\partial \phi}{\partial t}\right| = \frac{\phi_{nudge} - \bar{\phi}}{\tau}$$

With τ^{-1} the nudging strength.
The columns depict:

- height in meters
- Nudging strength
- The nudging value of u, v, θ_l and q_t

The nudging can be time dependent, so each block shows the nudging at a specific time, set by the number that starts the block just after the $\#$

Data files

datafiles directory

- `dmin_wetgrowth_lookup.dat` Only for level=5 microphysics: Look up table for growth ice hydrometeors
- `*.lay`: To be copied to the run dirs and named `backrad_in`. It describes the radiative background state of the atmosphere, including pressure, temperature, humidity and ozone profiles. Only used for `iradtyp = 4` and between the top of the domain and the top of the atmosphere.
- `*.dat` Internal lookup tables for iradtyp=4 radiation

The Namelist

- The only obligatory input file
- Has to be named: `NAMELIST` (in capitals)
- All input is being put in a single namelist, read at `LES.f90`

Grid and Time setup I

Variable	Default	
expnme	Default	experiment name
filprf	x	file prefix for use in constructing output files
nxp	132	total number of x points $(N_y + 4)$
nyp	132	total number of y points $(N_y + 4)$
nzp	105	total number of z points
deltax	35.0 m	grid spacing in x-direction
deltay	35.0 m	grid spacing in y-direction

Grid and Time setup II

deltaz	17.5 m	grid spacing in *z*-direction
dzrat	1.02	grid stretching ration (default 2% per interval)
dzmax	1200 m	height at which grid-stretching begins
igrdtyp	1	control parameter for selecting vertical grid
dtlong	10 s	maximum timestep
hfilin	test.	name of input history file for HISTORY starts (xxx.)
timmax	18000 s	final time of simulation

Grid and Time setup III

wctime		Wall clock time to break off the simulation
nfpt	5	number of levels in upper sponge layer
distim	300 s	minimum relaxation time in sponge layer
naddsc	0	number of additional scalars
runtyp	INITIAL	type of run ('INITIAL' or 'HISTORY')

Physics I

Variable	Default	
iradtyp	0	control parameter for selecting radiation model
CCN	150×10^6	cloud droplet mixing ratio
level	0	0=thermodynamic level, 1=dry cbl, 2=moist cbl (no rain), 3=moist cbl (with rain), 4, 5= ice microphysics
corflg	false	coriolis acceleration (true/false)
radfrq	0	radiation update interval
strtim	0	GMT of model time

Physics II

cntlat	31.5° N	model central latitude
case_name	astex	specify case name (rico,astex,bomex)
lsvarflg	false	reads large scale forcings from the file lscale_in
div	3.75e-6 s^{-1}	divergence
umean	0.	Mean U velocity (subtracted during the calculations
vmean	0.	Mean V velocity (subtracted during the calculations
th00	288	Basic state temperature (subtracted during the calculations

Physics III

sst	292 K	sea surface temperature
isfctyp	0	surface parameterization type (0: specified fluxes; 1: specified surface layer gradients; 2: fixed lower boundary of water, 3-5: Specific variations. See the surface lecture for more information.
ubmin	0.20	minimum u for u_* computation
zrough	0.1	momentum roughness height (if less than zero use Charnock relation)
dthcon	100 Wm^{-2}	surface temperature gradient (isfcflg=1) or surface heat flux (itsflg=0)
drtcon	0 Wm^{-2}	surface humidity (mixing ratio) gradient (isfcflg=1) or surface latent heat flux (itsflg=0)
csx	0.23	Smagorinsky Coefficient

Physics IV

prndtl	1/3	Prandtl Number (if less than zero no sgs for scalars)
sfc_albedo		Albedo of the surface
lnudge		Switching on/off nudging
tnudgefac		Factor to strengthen the nudging
ltimedep		Switch for time depend fluxes and large scale forcings
SolarConstant		Top of Atmosphere radiation

Initial profiles I

Variable	Default	
ipsflg	1	control parameter for input sounding (0: pressure in hPa; 1: height in meters with ps(1)= p_{sfc})
itsflg	1	control parameter for input sounding (0: ts = θ; 1: $ts = \theta_l$)
irsflg	1	control parameter for input sounding (0: rs = Rel. Hum) 1: ($rs = q_t$)
us	n/a	input zonal wind souding
vs	n/a	input meridional wind souding
ts	n/a	input temperature souding

Initial profiles II

rts	n/a	input humidity souding
ps	n/a	input pressure sounding
hs	n/a	vertical position
iseed	0	random seed
zrand	200 m	height below which random perturbations are added

Statistics and output I

Variable	Default	
outflg	true	output flag (true/false)
lsync	false	Synchronize the crosssection output (true/false)
frqhis	9000 s	history write interval
frqanl	3600 s	analysis write interval
slcflg	false	write slice output (true/false)
istpfl	1	print interval for timestep info

Statistics and output II

ssam_intvl	30 s	statistics sampling interval
savg_intvl	1800 s	statistics averaging interval
lcross	false	Crosssection output (true/false)
frqcross	3600 s	crosssection write interval
lxy	false	Crosssection output in xy plane (true/false)
zcross	0	Crosssection location of xy plane (true/false)
lxz	false	Crosssection output in xz plane (true/false)

Statistics and output III

ycross	0	Crosssection location of xy plane (true/false)
lyz	false	Crosssection output in yz plane (true/false)
xcross	0	Crosssection location of xy plane (true/false)
lwaterbudget	false	Crosssection of (costly) waterbudget (true/false)

Structure of the code

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Files and Modules I

LES	Main program which calls a timing routine and the driver, as well as the driver subroutine and the subroutine which defines and reads the model NAMELIST file.
advf	Calculates the tendencies associated with scalar advection.
advl	Calculates the tendencies associated with momentum advection.
defs	Defines physical constants.
forc	Case specific forcings (radiation, subsidence, *etc.*).

Files and Modules II

grid	Definition of grid, allocation of memory and I/O management
init	Routines for processing input (either from a file or the NAMELIST), definition of basic state, initialization of fields, and definition of initial random perturbations.
lsvar	computes sst, div and winds for astex case (only when lsvar=true in NAMELIST)
ncio	Defines structure of ncdf output files.
icemcrp	Bulk microphysical routines.
mpi_interface	Definition of MPI parameters and MPI routines for the domain decomposition (only when using MPI mode else seq_interface).
prss	Poisson solver, calculates the velocity tendencies associated with pressure gradients, also implements time-filter for Runge Kutta scheme and updates velocity.

Files and Modules III

rad_cldwtr	Calculates radiation properties from cloud water and effective radius.
rad_corkds	Reads gas concentrations and calculates radiative properties such as optical depth and absorption coefficients.
rad_d4strm	Computes radiative fluxes and optical properties for Rayleigh scattering.
rad_driver	Includes background soundings for atmospheric gases.
rad_gcss	Simple radiative parametrization for SW and LW fluxes (Delta-Eddigton approximation).
rad_rndnmb	Contains a random number generator.
rad_solver	Radiation solver.

Files and Modules IV

sgsm	Subgrid scale solver.
srfc	Surface boundary condition routines.
stat	Routines for calculating, accumulating and outputting model statistics. Statistical output is provided through the course of a simulation and tends to be problem specific.
step	Time stepper. Also includes several routines for computing tendencies due to physical processes (Coriolis force, buoyancy) or boundary conditions (Rayleigh friction for sponge layer near lid). Updating of scalars is done here. CFL computations and timestep-regridding are also here.

Files and Modules V

thrm	Thermodynamic routines for calculating quantities like temperature, and cloud water, given the thermodynamic state of the model, i.e., $\theta_l, q_t, \rho_0, \pi_0, \Theta_0$.
util	A collection of basic utilities including boundary conditions, FFT calls, explicit array operations such as domain or slab averaging or covariances, the tridiagonal solver, and some NetCDF utilities. Many of the routines in this module make active MPI calls.

Main Variables I

a_xp,a_xt1,a_xt2	4D	Data arrays used to summarize variables
a_up,a_vp,a_wp	3D	u^n, v^n, w^n
a_ut,a_vt,a_wt	"	$\partial_t u$, $\partial_t v$, $\partial_t u$
a_tp,a_tt	"	Liquid water potential temperature, ${\theta'_l}^n$, $\partial_t \theta_l$
a_rp,a_rt	"	Total water mixing ratio r_t^n, $\partial_t r_t$
a_rpp,a_rpt	"	Rain mass mixing ratio r_r^n, $\partial_t r_r$ (for level 3)
a_npp,a_npt	"	Rain number mixing ratio, n_r^n, $\partial_t n_r$ (for level 3)

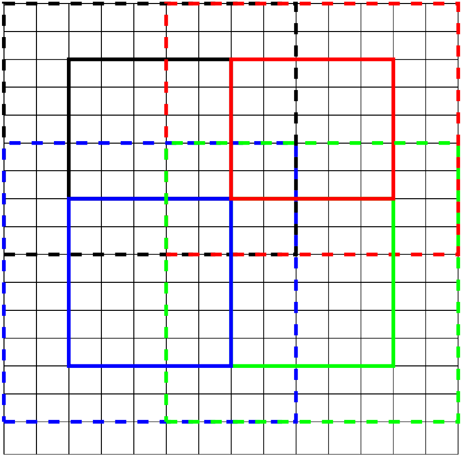
Main Variables II

a_theta	"	Potential temperature, θ (diagnosed from model state)
rc,rv	"	Condensate and vapor mixing ratio r_c, r_v (note that r_c can be either the cloud or total condensate mixing ratio depending on when it is accessed)
press, a_pexnr	"	Pressure and Exner function (p, π respectively)
a_scr1, a_scr2	"	Three dimensional scratch arrays
a_ustar, a_tstar, a_rstar	2D	Surface scales, u_*, θ_*, r_* respectively
uw_sfc, vw_sfc, ww_sfc	"	Surface momentum fluxes, $\overline{u'w'}$, $\overline{v'w'}$, $\overline{w'w'}$ respectively.
wt_sfc, wq_sfc	"	Surface thermodynamic fluxes, $\overline{w'\theta'}$, $\overline{w'r'}$ respectively.

Main Variables III

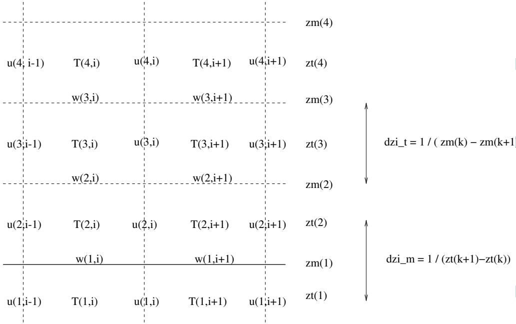
precip	"	Precipitation flux
dn0	1D	Basic state density, $\rho_0(z)$.
xt, yt, zt	"	Position of thermodynamic points
xm, ym, zm	"	position of momentum points
dzi_t	"	$1/(z_m(k) - z_m(k-1)$
dzi_m	"	$1/(z_t(k+1) - z_t(k)$

The Horizontal Grid



- The grid is equidistant in the 2 horizontal directions
- 1 processor covers a certain part of the grid
- And has 2 ghost cells around it on all sides
- All processors together show a big amount of overlap
- Parallelization remains efficient with $> 16 \times 16$ points per processor

The Vertical Grid



- The grid is staggered as a Arakawa C-grid
- Pressure and scalars are defined at cell center
- The velocities are defined at the cell faces to avoid decoupling between pressure and velocity
- The upper/right cell face has the same index as the cell center

Statistics and output

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Output files I

- Restart files `*.rst` only for internal model use. Output every `frqhis` seconds
- 3D output files `name.nc`: 3D output of the main quantities. Output done every `frqanl` seconds. Bulky!
- 2D Crosssections `name.out.cross*nc`: Crosssections of the data in th xy, xz, yz planes, as well as 2D integrated quantities like Liquid Water Path. Output done every `frqcross` seconds, governed by `lcross`, `lxy`, `lxz`, `lyz`
- 1D Profiles `name.ps*nc`. Profiles as a function of height. Output every `savg_intvl`, sampling every `ssam_intvl`. Need to be post processed for MPI runs.
- Timeseries `name.ts.*nc`. Domain averaged surface values, liquid water paths, cloud fraction etc. Output and sampling done every `ssam_intvl`. Needs to be post processed for MPI runs.

Statistics and output I

Variable	Default	
outflg	true	output flag (true/false)
lsync	false	Synchronize the crosssection output (true/false)
frqhis	9000 s	history write interval
frqanl	3600 s	analysis write interval
slcflg	false	write slice output (true/false)
istpfl	1	print interval for timestep info

Statistics and output II

ssam_intvl	30 s	statistics sampling interval
savg_intvl	1800 s	statistics averaging interval
lcross	false	Crosssection output (true/false)
frqcross	3600 s	crosssection write interval
lxy	false	Crosssection output in xy plane (true/false)
zcross	0	Crosssection location of xy plane (true/false)
lxz	false	Crosssection output in xz plane (true/false)

Statistics and output III

ycross	0	Crosssection location of xy plane (true/false)
lyz	false	Crosssection output in yz plane (true/false)
xcross	0	Crosssection location of xy plane (true/false)
lwaterbudget	false	Crosssection of (costly) waterbudget (true/false)

Timeseries

- Postprocessing to make 1 file out of all the files per processor
- Build tool in `uclales/misc/synthesis`:
- `ifort reducets.f90 -o reducets `/path/to/netcdflib/bin/nc-config --fflags --flibs ``
- NOTE: The quotation marks are accent graves (Under the tilde at a US International keyboard
- Use it to gather your timeseries statistics with: `reducets name nx ny`
  - `name` is the *stem* of the filename (so everything before `.ts.00....`
  - `nx` is the number of processes in the x-direction
  - `ny` is the number of processes in the y-direction

Profiles

- Postprocessing to make 1 file out of all the files per processor
- Build tool in `uclales/misc/synthesis`:
- `ifort reduceps.f90 -o reduceps` `` `/path/to/netcdflib/bin/nc-config --fflags --flibs ` ``
- NOTE: The quotation marks are accent graves (Under the tilde at a US International keyboard
- Use it to gather your profile statistics with: `reduceps name nx ny`
  - `name` is the *stem* of the filename (so everything before `.ps.00....`
  - `nx` is the number of processes in the x-direction
  - `ny` is the number of processes in the y-direction

Adding to Profiles and Timeseries

- Both profiles and timeseries are written from `ncio.f90` and `stat.f90`
- They are known to change over time.

Plot

- You're completely free to do what you want :)
- Depending on who you are and what you want for a plot, you could use NCL, Matlab, Python, Ferret, NCView,...
- We'd like to build up a tools database, so feel even more free to submit scripts over git
- As a starter, copy the 2 `plotfld.*` scripts from `uclales/misc/analysis/`
- Explore `plotfld.csh`, and put in the right variable names and time frame.
- Run it!
- Output sits in two pdf files `t1.pdf` and `p1.pdf`

Crosssections

- Postprocessing to make 1 file out of all the files per processor:
- `cdo gather name.out.cross*nc name.out.cross.nc`
- Watch the file quickly with (for instance) `ncview`

Advection, diffusion and subgrid

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Solving velocity $\bar{u}_j$ and scalars $\bar{\phi}$ includes Advection, Diffusion, Pressure and other forces and sources.

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i$$

$$\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \gamma_{\phi j})}{\partial x_j} + \mathcal{S}_\phi$$

Anelastic continuity

$$\frac{\partial (\rho_0 u_i)}{\partial x_i} = 0$$

Ideal gas law equation of state

$$\theta_v = \theta \left[1 + (\frac{R_v}{R_d} - 1) q_t - \frac{R_v}{R_d} q_l \right].$$

Timestepping I

Time stepping is based on a Runge-Kutta third order method.
The tendencies are calculated through 3 iterations:

$$
\begin{aligned}
\phi_*^n &= \phi^n + \alpha_1 \frac{\partial \phi^n}{\partial t}\Delta t \\
\phi_{**}^n &= \phi_*^n + \alpha_2 \frac{\partial \phi^n}{\partial t}\Delta t + \beta_2 \frac{\partial \phi_*^n}{\partial t}\Delta t \\
\phi^{n+1} &= \phi_{**}^n + \alpha_3 \frac{\partial \phi_{**}^n}{\partial t}\Delta t + \beta_3 \frac{\partial \phi_*^n}{\partial t}\Delta t
\end{aligned}
$$

With $\alpha_i = (\frac{8}{15}, -\frac{17}{60}, \frac{3}{4})$ and $\beta_i = (0, -\frac{15}{12}, -\frac{15}{12})$.

Timestepping II

- The timestep Δt (or `dt`.) in the code is variable
- Bounded by the Courant criterion ($CFL = 0.5$)
- Bounded by `dt_long` in `NAMELIST`. Use it for:
  - Unstabilties not in advection
  - Unstable spin ups
  - Circumventing bugs (but fix them later!)
- *Not* bounded by e.g. statistical timesteps. First step after t_{samp} is taken for statistics; faster but slightly imprecise.

Solving velocity $\bar{u}_j$ and scalars $\bar{\phi}$ includes Advection, Diffusion, Pressure and other forces and sources.

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i$$

$$\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \gamma_{\phi j})}{\partial x_j} + \mathcal{S}_\phi$$

Anelastic continuity

$$\frac{\partial(\rho_0 u_i)}{\partial x_i} = 0$$

Ideal gas law equation of state

$$\theta_v = \theta \left[1 + (\frac{R_v}{R_d} - 1) q_t - \frac{R_v}{R_d} q_l\right].$$

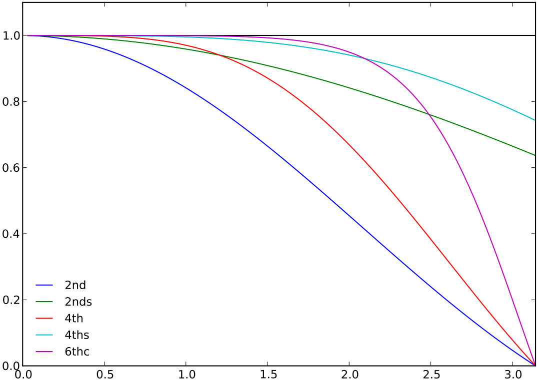
Advection

Advection can be best thought of flux through the boundaries of the cell:

$$\frac{\partial \bar{u}_i \phi_i}{\partial x} = \qquad \frac{F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}}{\Delta x}$$

with $F_{i+\frac{1}{2}}$ the flux through the cell boundary at $i + \frac{1}{2}$.

Advection



Advection

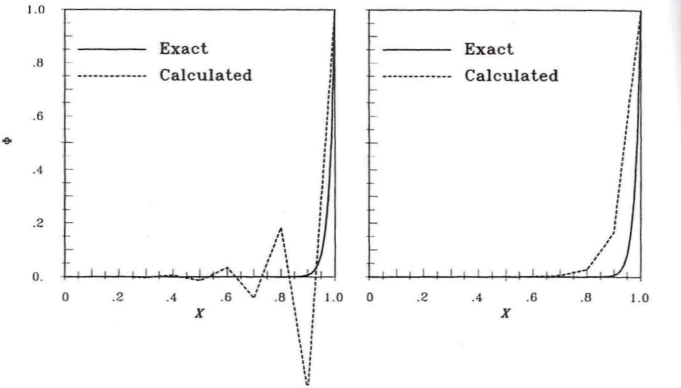


Fig. 3.8. Solution of the 1D convection/diffusion equation at Pe = 50 using CDS (left) and UDS (right) for convection terms and a uniform grid with 11 nodes

In UCLALES, we do 4th order Central Differencing for momentum advection, and flux-limited advection for scalars to guarantee positive values

$$F^{4th}_{i+\frac{1}{2}} = \frac{u_{i+\frac{1}{2}}}{12}\left[-\phi_{i-1} + 7\phi_i + 7\phi_{i+1} - \phi_{i+2}\right]$$

$$F^{\kappa}_{i+\frac{1}{2}} = \bar{u}_{i+\frac{1}{2}}\left[\phi_i + \frac{1}{2}\kappa_{i+\frac{1}{2}}\left(\phi_i - \phi_{i-1}\right)\right]$$

With $\kappa_{i+\frac{1}{2}} > 0$ and a function of consecutive gradients (assuming $u_{i+\frac{1}{2}}$):
$r = \frac{\phi_{i+1}-\phi_i}{\phi_i-\phi_{i-1}}$

Flux limiter schemes

Depending on the setting `lmtr` in `advf.f90`, we use:

$$
\begin{aligned}
\textbf{minmod} \quad & \min(r, 1) \\
\textbf{superbee} \quad & \max(\min(2r, 1), \min(r, 2)) \\
\textbf{MC} \quad & \min(2r, \frac{1+r}{2}, 2) \\
\textbf{vanLeer} \quad & \frac{r+|r|}{1+|r|}
\end{aligned}
$$

By default, it is set to MC.
Effectively, limiter schemes switch back to low order upwind schemes whenever the local gradient is to steep.
This happens a lot in turbulent fields. This can cause so much diffusion that the SFS scheme is rendered useless

Solving velocity $\bar{u}_j$ and scalars $\bar{\phi}$ includes Advection, Diffusion, Pressure and other forces and sources.

$$
\begin{aligned}
\frac{\partial \bar{u}_i}{\partial t} &= -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i \\
\frac{\partial \bar{\phi}}{\partial t} &= -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \gamma_{\phi j})}{\partial x_j} + \mathcal{S}_\phi
\end{aligned}
$$

Anelastic continuity

$$\frac{\partial (\rho_0 u_i)}{\partial x_i} = 0$$

Ideal gas law equation of state

$$\theta_v = \theta \left[1 + (\frac{R_v}{R_d} - 1) q_t - \frac{R_v}{R_d} q_l \right].$$

Diffusion I

The sub-grid fluxes τ_{ij} and $\gamma_{\phi j}$ are not known explicitly and thus must be modeled. This constitutes the model closure. The basic or default form of the closure makes use of the Smagorinsky model, wherein

$$\tau_{ij} = -\rho_0 K_m D_{ij} \quad \text{and} \quad \gamma_{\phi j} = -\frac{K_m}{Pr}\frac{\partial \bar{\phi}}{\partial x_j},$$

where

$$D_{ij} = \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}$$

is the resolved deformation, K_m is the eddy viscosity, and Pr is an eddy Prandtl number. The Smagorinsky model calculates the eddy viscosity as

$$K_m = (C_s \ell)^2 S \sqrt{1 - \frac{Ri}{Pr}} \quad \text{where} \quad Ri = \frac{S^2}{N^2}$$

Diffusion II

and

$$S^2 \equiv \frac{\partial \bar{u}_i}{\partial x_j} D_{ij} \quad \text{and} \quad N^2 = \frac{g}{\Theta_0} \frac{\partial \bar{\theta}_v}{\partial z}.$$

In the above C_s is the Smagorinsky constant and takes on values near 0.2, and

$$\ell^{-2} = (\Delta x \Delta y \Delta z)^{-2/3} + (z\kappa/C_s)^{-2},$$

where $\kappa = 0.35$ is the von Kármán constant in the model.

Solving velocity $\bar{u}_j$ and scalars $\bar{\phi}$ includes Advection, Diffusion, Pressure and other forces and sources.

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i$$

$$\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \gamma_{\phi j})}{\partial x_j} + \mathcal{S}_\phi$$

Anelastic continuity

$$\frac{\partial(\rho_0 u_i)}{\partial x_i} = 0$$

Ideal gas law equation of state

$$\theta_v = \theta \left[1 + (\frac{R_v}{R_d} - 1) q_t - \frac{R_v}{R_d} q_l \right].$$

Pressure(s) I

Exner function: $\bar{\pi} = (\bar{p}/p_{00})^{R_d/c_p}$
The anelastic approximation solves for perturbations about a hydrostatic basic state of constant potential temperature, i.e.,

$$\frac{d\pi_0}{dz} = -\frac{g}{c_p\Theta_0},$$

where subscript 0 denotes a basic state value, which depend only on z (Θ_0 being constant).
For gravity, we use buoyancy deviations from the slab average (not the basic state). For consistency, introduce a second exner π_1:

$$\frac{d}{dz}(\pi_0 + \pi_1) = -\frac{g}{c_p\bar{\theta}_v},$$

Start with continuity:

$$\frac{\partial(\rho_0 u_i)}{\partial x_i} = 0$$

And the momentum equation:

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \mathcal{F}_i$$

Fill them in in each other:

$$\frac{\partial}{\partial x_i}\left(\rho_0 \frac{\partial \bar{u}_i}{\partial t}\right) = \frac{\partial}{\partial x_i}\left[-\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - \rho_0 c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \rho_0 \mathcal{F}_i\right] = 0$$

Calculating Pressure II

$$\frac{\partial}{\partial x_i}\left[-\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - \rho_0 c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \rho_0 \mathcal{F}_i\right] = 0$$

Bring the pressure to the other side:

$$\frac{\partial}{\partial x_i}\left(\rho_0 c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i}\right) = \frac{\partial}{\partial x_i}\left[-\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \rho_0 \mathcal{F}_i\right]$$

And we end up with a Poisson equation:

$$\frac{\partial}{\partial x_i}\left(\rho_0 \frac{\partial \bar{\pi}}{\partial x_i}\right) = \frac{1}{c_p \Theta_0}\frac{\partial}{\partial x_i}\left[-\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \rho_0 \mathcal{F}_i\right]$$

that can be solved efficiently (but globally!) in Fourier space.

The surface model

UCLALES Tutorial

Cathy Hohenegger

Max Planck Institute for Meteorology

November 7 - 11, 2011

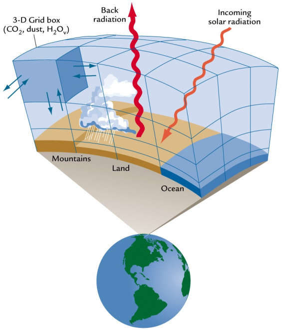
Outline

- The problem
- Physical basis
- The surface subroutine in the UCLALES
- Future development

Why caring about the surface ?

From a very pragmatic point of view:

Need to know the conditions at the bottom boundary to be able to integrate the relevant equations



Why caring about the surface ?

From a less pragmatic point of view:

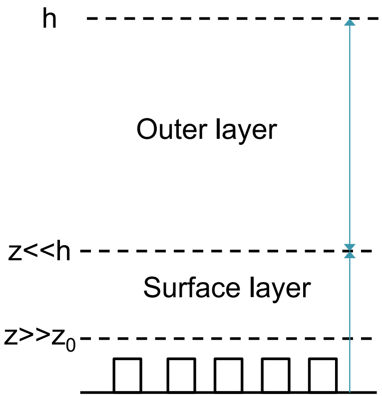
Surface influences the structure and evolution of the planetary boundary layer

- Friction: momentum flux: slows down wind
- Solar absorption: sensible heat flux: warms/cools overlying air
- Solar absorption: latent heat flux: water source for precipitation
- Introduces diurnal cycle

Partitions available energy between sensible and latent heat fluxes

Physical basis

The atmospheric boundary layer, J.R. Garratt



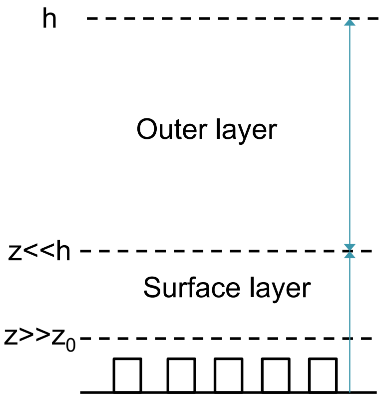
Physical basis

The atmospheric boundary layer, J.R. Garratt

Flow in the PBL is turbulent

$$\chi = \bar{\chi} + \chi'$$

$$\frac{\partial \bar{u}}{\partial t} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial z} + f\bar{v} - \frac{\partial \overline{u'w'}}{\partial z}$$



Variables to be determined

Surface stress in x	$\tau_{x0} = -\rho\overline{u'w'}_0$
Surface stress in y	$\tau_{y0} = -\rho\overline{v'w'}_0$
Surface sensible heat flux	$H_0 = \rho c_p\overline{w'\theta'}_0$
Surface buoyancy flux	$H_{v0} = \rho c_p\overline{w'\theta'_v}_0$
Surface latent heat flux	$LE_0 = \rho L_v\overline{w'q'}_0$

How do we compute the fluxes?

Use of surface similarity theory:

- Only valid for the surface layer $z_0 < z << h$ where fluxes remain constant
- Isolate the relevant scales that can fully characterize the flow in the surface layer
- Arrange them in dimensionless group to form appropriate relationships
- Use data for fitting

How do we compute the fluxes?

Use of surface similarity theory:

- Only valid for the surface layer $z_0 < z << h$ where fluxes remain constant
- Isolate the relevant scales that can fully characterize the flow in the surface layer
- Arrange them in dimensionless group to form appropriate relationships
- Use data for fitting

In general: determine e.g. appropriate velocity and length scales to scale the wind profile to derive not only the wind profile law but also to use this to formulate a suitable drag law

Characteristic scales

Friction velocity	$u_{*0}^2 = \sqrt{(\overline{u'w'}_0)^2 + (\overline{v'w'}_0)^2} = -\overline{u'w'}_0$
Temperature scale	$\theta_{*0} = \dfrac{-\overline{w'\theta'}_0}{u_{*0}}$
Temperature scale	$\theta_{v*0} = \dfrac{-\overline{w'\theta'_v}_0}{u_{*0}}$
Humidity scale	$q_{*0} = \dfrac{-\overline{w'q'}_0}{u_{*0}}$
Obukov stability length	$L = \dfrac{u_{*0}^2 \overline{\theta_v}}{kg\theta_{v*0}}$

Characteristic scales

Friction velocity	$u_{*0}^2 = \sqrt{(\overline{u'w'}_0)^2 + (\overline{v'w'}_0)^2} = -\overline{u'w'}_0$
Temperature scale	$\theta_{*0} = \dfrac{-\overline{w'\theta'}_0}{u_{*0}}$
Temperature scale	$\theta_{v*0} = \dfrac{-\overline{w'\theta'_v}_0}{u_{*0}}$
Humidity scale	$q_{*0} = \dfrac{-\overline{w'q'}_0}{u_{*0}}$
Obukov stability length	$L = \dfrac{u_{*0}^2 \overline{\theta_v}}{kg\theta_{v*0}}$

Relations: First-order closure

Assume neutral and homogeneous atmosphere:

$$\overline{u'w'}_0 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$K_M \sim kzu_{*0} \qquad\qquad k = 0.4$$

$$\overline{u'w'}_0 = -u_{*0}^2$$

$$\frac{kz}{u_{*0}} \frac{\partial \bar{u}}{\partial z} = 1$$

Integrate:

$$\frac{k\bar{u}}{u_{*0}} = \ln \frac{z}{z_0}$$

Relations: First-order closure

Assume neutral and homogeneous atmosphere:

$$\overline{u'w'}_0 = -K_M \frac{\partial \bar{u}}{\partial z}$$

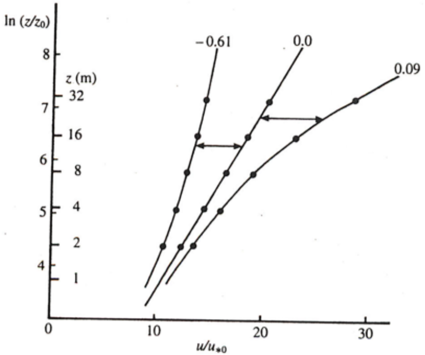
$$K_M \sim kzu_{*0}$$

$$\overline{u'w'}_0 = -u_{*0}^2$$

$$\frac{kz}{u_{*0}} \frac{\partial \bar{u}}{\partial z} = 1$$

Integrate:

$$\frac{k\bar{u}}{u_{*0}} = \ln \frac{z}{z_0}$$



Relations: First-order closure

In general:

$$\overline{u'w'}_0 = -u_{*0}^2 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$\frac{kz}{u_{*0}} \frac{\partial \bar{u}}{\partial z} = \Phi_M(\zeta) \quad \zeta = z/L$$

Integrate:

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \int (1 - \Phi_M(\zeta'))\mathrm{d}(\ln\zeta')$$

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

Relations: First-order closure

In general:

$$\overline{u'w'}_0 = -u_{*0}^2 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$\frac{kz}{u_{*0}} \frac{\partial \bar{u}}{\partial z} = \Phi_M(\zeta) \quad \zeta = z/L$$

$$\overline{w'\theta'_{v0}} = -u_{*0}\theta_{v*0} = -K_H \frac{\partial \bar{\theta}_v}{\partial z}$$

$$\frac{kz}{\theta_{v*0}} \frac{\partial \bar{\theta}_v}{\partial z} = \Phi_H(\zeta)$$

Integrate:

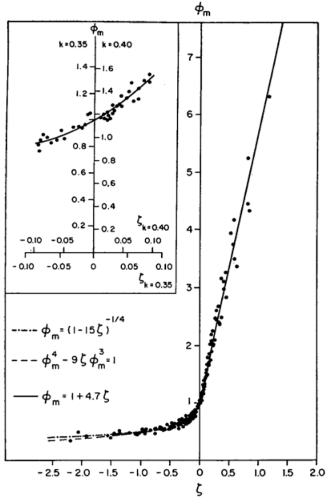
$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \int (1 - \Phi_M(\zeta'))\mathrm{d}(\ln\zeta')$$

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \int (1 - \Phi_H(\zeta'))\mathrm{d}(\ln\zeta')$$

$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$$

Form of the functions



Busigner et al. (1971)

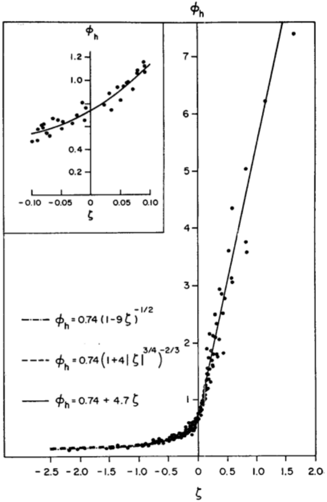
Form of the functions

$$\frac{kz}{u_{*0}}\frac{\partial \bar{u}}{\partial z} = \Phi_M(\zeta)$$
$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

Following Garrat:

$$\zeta < 0 : \; \Phi_M(\zeta) = (1 - 16\zeta)^{-1/4}$$
$$\zeta > 0 : \; \Phi_M(\zeta) = (1 + 5\zeta)$$
$$\zeta < 0 : \; \Psi_M(\zeta) = 2\ln(\frac{1+x}{2}) + \ln(\frac{1+x^2}{2}) - 2\tan^{-1}(x) + \pi/2, \; x = (1 - 16\zeta)^{1/4}$$
$$\zeta > 0 : \; \Psi_M(\zeta) = -5\zeta$$

Form of the functions



Businger et al. (1971)

Form of the functions

$$\frac{kz}{\theta_{v*0}}\frac{\partial\bar{\theta}_v}{\partial z} = \Phi_H(\zeta)$$
$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$$

Following Garrat:

$$\zeta < 0 : \; \Phi_H(\zeta) = (1 - 16\zeta)^{-1/2}$$
$$\zeta > 0 : \; \Phi_H(\zeta) = (1 + 5\zeta)$$
$$\zeta < 0 : \; \Psi_H(\zeta) = 2\ln(\frac{1+x}{2}),\; x = (1 - 16\zeta)^{1/2}$$
$$\zeta > 0 : \; \Psi_H(\zeta) = -5\zeta$$

Alternative: bulk transfer relations

$$\overline{u'w'}_0 = -u_{*0}^2 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

$$\overline{w'\theta'_{v}}_0 = -u_{*0}\theta_{v*0} = -K_H \frac{\partial \bar{\theta}_v}{\partial z}$$

$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$$

Alternative: bulk transfer relations

$$\overline{u'w'}_0 = -u_{*0}^2 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

$$\overline{w'\theta'_{v}}_0 = -u_{*0}\theta_{v*0} = -K_H \frac{\partial \bar{\theta}_v}{\partial z}$$

$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$$

$$\overline{u'w'}_0 = -u_{*0}^2 = -C_D \bar{u}^2$$

$$C_D = \frac{k^2}{(\ln(\frac{z}{z_0}) - \Psi_M(\zeta))^2}$$

$$\overline{w'\theta'_{v}}_0 = -u_{*0}\theta_{v*0} = -C_H \bar{u}(\bar{\theta}_v - \bar{\theta}_{v0})$$

$$C_H = \frac{k^2}{(\ln(\frac{z}{z_0}) - \Psi_M(\zeta))(\ln(\frac{z}{z_T}) - \Psi_H(\zeta))}$$

Alternative: aerodynamic resistance

$$\overline{u'w'}_0 = -u_{*0}^2 = -K_M \frac{\partial \bar{u}}{\partial z}$$

$$\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$$

$$\overline{w'\theta'_v}_0 = -u_{*0}\theta_{v*0} = -K_H \frac{\partial \bar{\theta}_v}{\partial z}$$

$$\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$$

$$\overline{u'w'}_0 = -u_{*0}^2 = -C_D \bar{u}^2$$

$$C_D = \frac{k^2}{(\ln(\frac{z}{z_0}) - \Psi_M(\zeta))^2}$$

$$\overline{w'\theta'_v}_0 = -u_{*0}\theta_{v*0} = -C_H \bar{u}(\bar{\theta}_v - \bar{\theta}_{v0})$$

$$C_H = \frac{k^2}{(\ln(\frac{z}{z_0}) - \Psi_M(\zeta))(\ln(\frac{z}{z_T}) - \Psi_H(\zeta))}$$

$$\overline{u'w'}_0 = \frac{-\bar{u}}{r_{aM}}$$

$$r_{aM} = \frac{1}{C_D \bar{u}}$$

$$\overline{w'\theta'_v}_0 = \frac{\bar{\theta}_{v0} - \bar{\theta}_v}{r_{aH}}$$

$$r_{aH} = \frac{1}{C_H \bar{u}}$$

The surface subroutine in UCLALES

- Case default:
  sensible and latent heat fluxes prescribed,
  moment fluxes diagnosed from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$

The surface subroutine in UCLALES

- Case default:

  sensible and latent heat fluxes prescribed,

  moment fluxes diagnosed from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$

- Case 1:

  gradient in temperature and moisture prescribed,

  sensible and latent heat fluxes from $\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$

  momentum fluxes from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$

The surface subroutine in UCLALES

- Case default:

  sensible and latent heat fluxes prescribed,

  moment fluxes diagnosed from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$
- Case 1:

  gradient in temperature and moisture prescribed,

  sensible and latent heat fluxes from $\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$

  momentum fluxes from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$
- Case 2:

  surface temperature and moisture prescribed

  sensible and latent heat fluxes from $\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$

  momentum fluxes from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$

The surface subroutine in UCLALES

- Case default:
  sensible and latent heat fluxes prescribed,
  moment fluxes diagnosed from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$
- Case 1:
  gradient in temperature and moisture prescribed,
  sensible and latent heat fluxes from $\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$
  momentum fluxes from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$
- Case 2:
  surface temperature and moisture prescribed
  sensible and latent heat fluxes from $\frac{k(\bar{\theta}_v - \bar{\theta}_{v0})}{\theta_{v*0}} = \ln(\frac{z}{z_T}) - \Psi_H(\zeta)$
  momentum fluxes from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$
- Case 3:
  C_D, $C_{H,}$ surface temperature and moisture prescribed
  sensible and latent heat fluxes from $u_{*0}\theta_{v*0} = C_H\bar{u}(\bar{\theta}_v - \bar{\theta}_{v0})$
  momentum fluxes from $u_{*0}^2 = C_D\bar{u}^2$

The surface subroutine in UCLALES

- Case 4:

  Buoyancy flux prescribed

  moment fluxes diagnosed from $\frac{k\bar{u}}{u_{*0}} = \ln(\frac{z}{z_0}) - \Psi_M(\zeta)$

To note:

- Need to know z_0 and z_T . Currently z_0=z_T =zrough. If zrough sets to a value smaller or equal to zero, then

  $z_0 = \frac{0.016}{g} u_{*0}^2$ true for ocean

- Cases 2, 3 and 4 assume to be over the ocean, i.e. q_o=q_{sat}(SST)
- First point should be in the surface layer but higher than z_0

Namelist options with the cases

	isfctyp	dthcon	drtcon	zrough	sst
Default	0	Sensible heat flux Wm^{-2}	Latent heat flux Wm^{-2}	Roughness length	Not needed
Case 1	1	Temperature gradient $K\ m^{-1}$	Moisture gradient $kg\ kg^{-1}\ m^{-1}$	Roughness length	Not needed
Case 2	2	Not needed	Not needed	Roughness length	Sea surface temperature
Case 3	3	C_H	C_q	C_D	Sea surface temperature
Case 4	4	Buoyancy flux Wm^{-2}	Not needed	Roughness length	Sea surface temperature

Future development

At the moment, we cannot compute the fluxes interactively for land points

- Latent heat flux is the sum of evaporation and transpiration. Evaporation and transpiration are computed according to:

$$\frac{\rho L_v}{r_{aH} + r_s}(q_{sat}(\bar{T}_0) - \bar{q})$$

- A land surface model provides surface and vegetation resistances as well as surface temperature

Microphysics and Thermodynamics

UCLALES Tutorial

Axel Seifert

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst
DWD

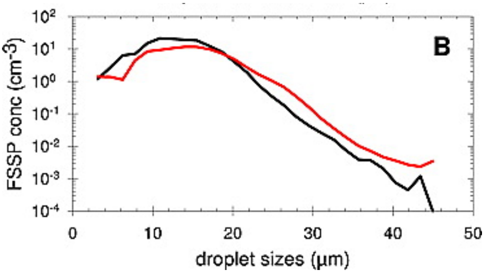
Overview

- → microstructure of clouds
- → cloud processes
- → bulk parameterization
- → warm rain: autoconversion / accretion
- → sedimentation
- → more details on sedimentation and evaporation
- → turbulence effects on warm rain
- → ice particle fall speeds
- → ice nucleation
- → glacitation of clouds
- → UCLA-LES microphysics schemes

Microstructure of liquid clouds

Liquid clouds are characterised by small micrometer sized droplet. Typical drops sizes range from 1-2 µm and a few tens of micrometers.

Drop size distributions in maritime shallow clouds

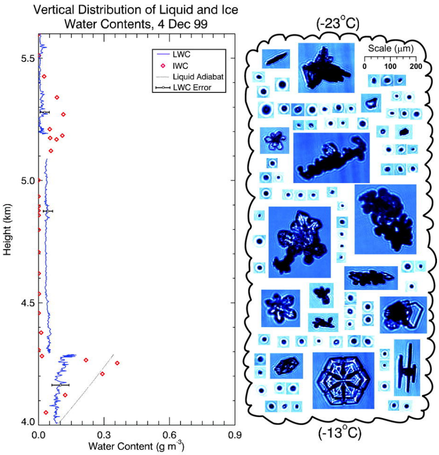


(from Hudson and Noble, 2009, JGR)

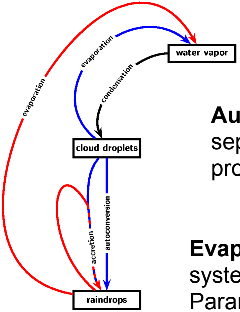
Microstructure of mixed-phase clouds

In mixed-phase clouds we find small liquid droplet coexisting with ice particles of different shapes and sizes.

Here an example of measurements with a Cloud Particle Imager (CPI) by Fleishhauer et al. (2002).



Cloud microphysical processes



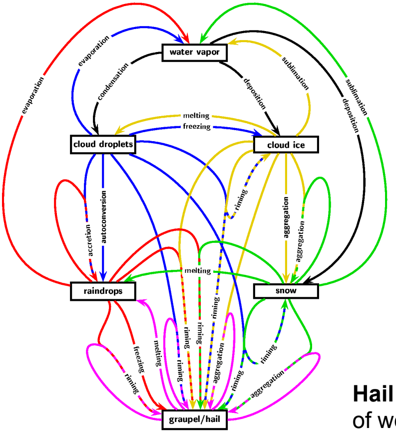
Evaporation and condensation of cloud droplets are usually parameterized by a saturation adjustment scheme.

Autoconversion is an artificial process introduced by the separation of cloud droplets and rain. Parameterization of the process is quite difficult and many different schemes are available.

Evaporation of raindrops can be very important in convective systems, since it determines the strength of the cold pool. Parameterization is not easy, since evaporation is very size dependent.

Even for the warm rain processes a lot of things are unknown or in discussion for decades, like effects of **mixing / entrainment** on the cloud droplet distribution, effects of **turbulence** on coalescence, **coalescence efficiencies, collisional breakup** or the details of the **nucleation** process. In most cloud models these problems are neglected or parameterized in a quite simple and ad-hoc way.

Cloud microphysical processes



Conversion processes, like snow to graupel conversion by riming, are very difficult to parameterize but very important in convective clouds.

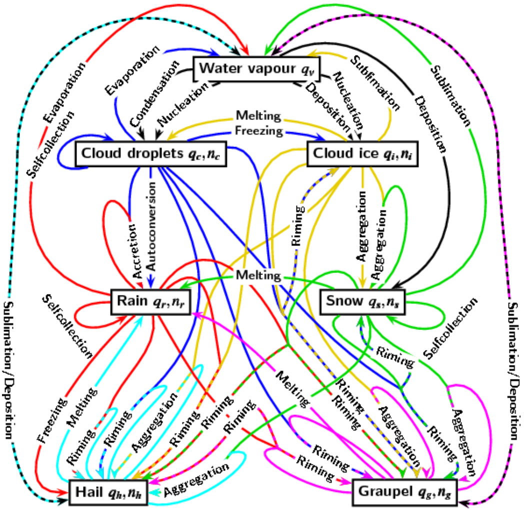
Especially for snow and graupel the particle properties like **particle density** and **fall speeds** are important parameters. The assumption of a constant particle density is questionable.

Aggregation processes assume certain collision and sticking efficiencies, which are not well known.

Hail processes is especially complicated because of wet growth, partial melting or shedding.

The so-called **ice multiplication** (or Hallet-Mossop process) may be very important, but is still not well understood

Cloud microphysical processes



This is the level=5 scheme in UCLA-LES

... but secondary processes, like Hallet-Mossop, are not included in the diagram.

Spectral formulation of cloud microphysics

The particle size distribution ***f(x),*** with some measure of particle size ***x***, is explicity calculated from

$$\frac{\partial f(x,\vec{r},t)}{\partial t} + \nabla \cdot [\vec{v}(\vec{r},t) f(x,\vec{r},t)] + \frac{\partial}{\partial z}[v_s(x) f(x,\vec{r},t)] + \frac{\partial}{\partial x}[\dot{x} f(x,\vec{r},t)] = \sigma_{coal} + \sigma_{break}$$

with

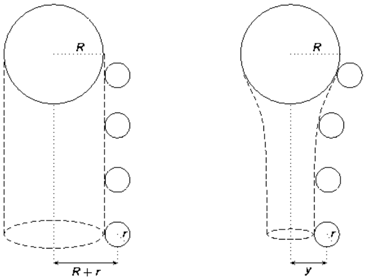
$$\sigma_{coal} = \frac{1}{2}\int_0^x f(x-x',\vec{r},t)\, f(x',\vec{r},t)\, K(x-x',x')\, dx' - \int_0^\infty f(x,\vec{r},t)\, f(x',\vec{r},t)\, K(x,x')\, dx'$$

and

$$\sigma_{break} = \frac{1}{2}\int_0^\infty \int_0^x f(x',\vec{r},t)\, f(x'',\vec{r},t)\, B(x',x'')\, P(x;x',x'')\, dx'\, dx'' - \int_0^\infty f(x,\vec{r},t)\, f(x',\vec{r},t)\, B(x,x')\, dx'.$$

The gravitational collision-coalescence kernel

$$K(x,y) = \pi\,[r(x)+r(y)]^2\,|v(x)-v(y)|\,E_{coll}(x,y)\,E_{coal}(x,y)$$

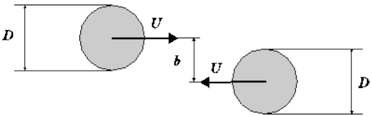


collision efficiency:

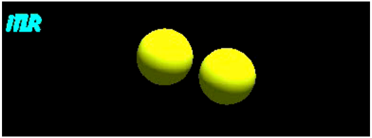
$$E_{coll} = \frac{y^2}{(R+r)^2}$$

The effects of in-cloud turbulence on the collision frequency is a current research topic. Recent results indicate that turbulence can significantly enhance the rain formation process.

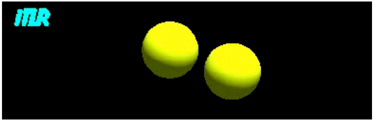
Collisional breakup



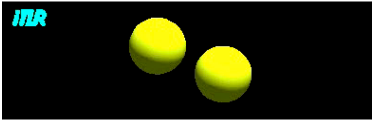
DNS by University Stuttgart



coalescence



no coalescence
(rebound)



collisional breakup
(filament type)

Bulk microphysical schemes

Instead of ***f(x)*** only moments of the size distribution are explicitly predicted like the liquid water content:

$$L = \frac{\pi \rho_w}{6} \int_0^\infty D^3 f(D) dD$$

or the number concentration of particles:

$$N = \int_0^\infty f(D) dD$$

maybe even a third one, like the sixth moment (reflectivity)

Bin vs. bulk microphysics

Spectral bin model (100-500 variables):

$$\frac{\partial f(x)}{\partial t} + \nabla \cdot [\,\mathbf{v}\, f(x)\,] + \frac{\partial}{\partial z}[\,v_T(x)\, f(x)\,] = \mathcal{F}(x)$$

Two-moment bulk model (8-12 variables):

$$\frac{\partial N}{\partial t} + \nabla \cdot [\,\mathbf{v}\, N\,] + \frac{\partial}{\partial z}[\,v_N(\bar{x})\, N\,] = N\,\mathcal{G}(\bar{x})$$

$$\frac{\partial L}{\partial t} + \nabla \cdot [\,\mathbf{v}\, L\,] + \frac{\partial}{\partial z}[\,v_L(\bar{x})\, L\,] = L\,\mathcal{H}(\bar{x}), \qquad \bar{x} = L/N$$

UCLA-LES level=3 and level=5 are a two-moment schemes

One-moment bulk model (3-5 variables):

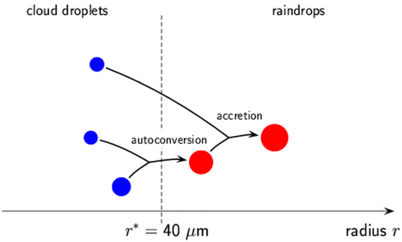
$$\frac{\partial L}{\partial t} + \nabla \cdot [\,\mathbf{v}\, L\,] + \frac{\partial}{\partial z}[\,\tilde{v}_L(L)\, L\,] = \mathcal{S}(L)$$

UCLA-LES level=4 is a mix of one- and two-moment scheme

Note: cloud droplets are single moment in all UCLA-LES schemes, number is prescribed.

Kessler's warm phase scheme

In 1969 Kessler published a very simple warm rain parameterization which is still used in many bulk schemes.



autoconversion rate:

$$\left.\frac{\partial L_r}{\partial t}\right|_{au} = \begin{cases} k\,(L_c - L_0), & \text{if } L_c > L_0 = 0.5 \text{ g m}^{-3} \\ 0, & \text{else} \end{cases}$$

„As we know, water clouds sometimes persist for a long time without evidence of precipitation, but various measurements show that cloud amounts > 1 g/m³ are usually associated with production of precipitation. It seems reasonable to model nature in a system where the rate of cloud autoconversion increases with the cloud content but is zero for amounts below some threshold."

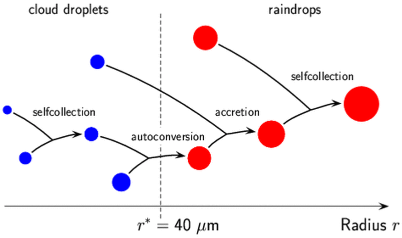
(E. Kessler: *On the Distribution and Continuity of Water Substance in Atmospheric Circulation*, *Meteor. Monogr.* , 1969)

A two-moment warm phase scheme

Assuming a Gamma distribution for cloud droplets

$$f_c(x) = A x^{\nu} e^{-Bx}$$

the following autoconversion can be derived from the spectral collection equation



$$\left.\frac{\partial L_r}{\partial t}\right|_{au} = \frac{k_c}{20\, x^*} \frac{(\nu+2)(\nu+4)}{(\nu+1)^2} L_c^2\, \bar{x}_c^2 \left[1 + \frac{\Phi_{au}(\tau)}{(1-\tau)^2}\right]$$

with a universal function.

$$\Phi_{au}(\tau) = 600 \tau^{0.68} (1 - \tau^{0.68})^3$$

The universal function parameterizes the time evolution, i.e. the broadening, of the cloud droplet distribution during the rain formation process.

Seifert and Beheng (2001), Atmos. Res.

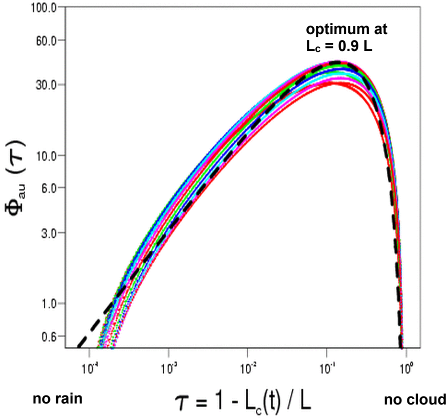
A two-moment warm phase scheme

The colored lines represent solutions of the spectral collection equation for various initial conditions.

The dashed line is the fit:

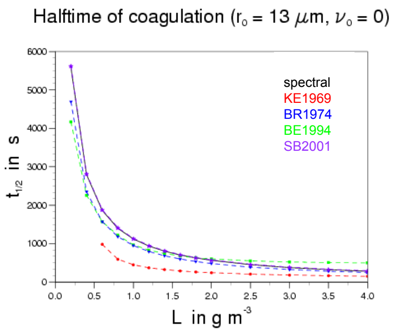
$$\Phi_{\mathrm{au}}(\tau) = 600\tau^{0.68}(1-\tau^{0.68})^3$$

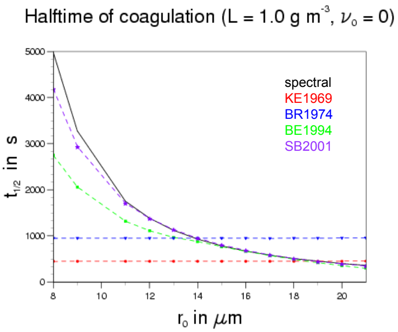
This function describes the **broadening of the cloud droplet size distribution** by collisions between cloud droplets.



Seifert and Beheng (2001), Atmos. Res.

A comparison of warm phase autoconversion schemes





- For high LWC the differences between the schemes are usually small
- For low LWC the differences are larger and the effects of drop size or cloud droplet number concentration on coalescence, can be important.

Sedimentation as an example for bulk process schemes

$$\frac{\partial f(D)}{\partial t} + \frac{\partial}{\partial z}\left[v(D)f(D)\right] = 0$$

with f(D) number density size distribution (unit m^{-4}).

Now we integrate for the (bulk) mass density (liquid water content)

$$L = \frac{\pi \rho_W}{6} \int\limits_0^\infty D^3 f(D) dD$$

and find

$$\frac{\partial L}{\partial t} + \frac{\partial}{\partial z}\left[v_L L\right] = 0$$

with the mass weighted fall velocity

$$v_L = \frac{\int_0^\infty D^3 f(D) v(D) dD}{\int_0^\infty D^3 f(D) dD}$$

… use the fundamental parameterization assumption

Now we assume that $f(D)$ can be described by an exponential distribution

$$f(D) = N_0 \ \exp(-\lambda D) \text{ with } N_0 = \text{const.}$$

All moments of this distribution are then given by

$$\mathcal{M}_n = \int_0^\infty D^n f(D) dD = \frac{\Gamma(n+1)}{\lambda^{n+1}}$$

or, more specific, for the liquid water content we find

$$L = \frac{\pi \rho_{\mathcal{W}}}{6} \int_0^\infty D^3 f(D) dD = \pi \rho_{\mathcal{W}} \lambda^4$$

… and specify a fall speed....

A power-law for the particle fall speed

$$v(D) = \alpha \left(\frac{D}{D_0} \right)^{\frac{1}{2}}$$

leads to the following sedimentation velocity:

$$v_L = \frac{\int_0^\infty D^3 f(D) v(D) dD}{\int_0^\infty D^3 f(D) dD} = \frac{N_0\ \alpha}{6} \Gamma \left(\frac{9}{2} \right) \left(\frac{L}{\pi \rho_{\mathcal{W}}} \right)^{\frac{1}{8}} = \tilde{\alpha} L^{\frac{1}{8}}$$

Note: This was just a one-moment scheme!

An interesting result for sedimentation:

Spectral microphysics:

$$\frac{\partial f(D)}{\partial t} + \frac{\partial}{\partial z}\left[v(D) f(D)\right] = 0$$

One-moment scheme:

No gravitational sorting!

$$\frac{\partial L}{\partial t} + \frac{\partial}{\partial z}\left[v_L(L) L\right] = 0$$

Two-moment scheme:

Has gravitational sorting!

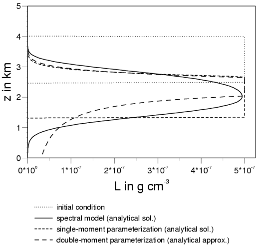
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial z}\left[v_1(\bar{x})\, N\right] = 0$$

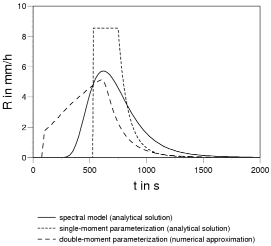
$$\frac{\partial L}{\partial t} + \frac{\partial}{\partial z}\left[v_2(\bar{x})\, L\right] = 0$$

Note: A linear PDE is parameterized by a nonlinear PDE!!

An idealized rainfall experiment

Sedimentation of a layer of raindrops as described by the spectral equation, a one-moment scheme and a two-moment scheme.





Both parameterizations have serious problems with this simple test!

Wacker and Seifert (2001), Atmos. Res.

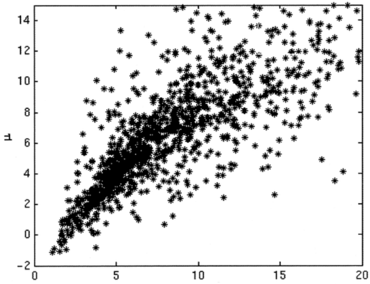
When you are stuck: Look at the real thing!

Especially in convective precipitation the raindrop size distribution f(D) is highly variable and not necessarily exponential. A better description is a Gamma distribution:

$$f(D) = N_0\, D^{\mu}\, \exp(-\lambda D)$$

Problem: μ and N_0 are highly variable and have a strong impact on evaporation and sedimentation

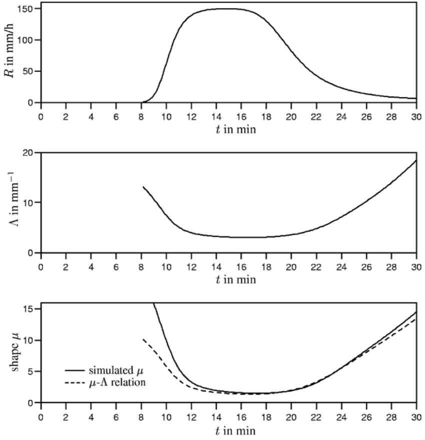
Zhang et al. (2001) measured μ vs. λ



$\lambda = \lambda(q_r)$ in mm^{-1}

Two-moment schemes do not necessarily solve (all) our problems!
.... but we can help them out.

Another idealized rainfall experiment



Simulation using a 1D rainshaft model with a homogenous cloud as initial condition.

The shape of the raindrop size distribution can be parameterized as a function of the slope parameter

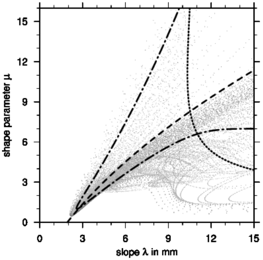
$$\mathbf{f(D) = N_0\, D^{\mu}\, exp(-\lambda D)}$$

with $\mathbf{\mu = \mu(\lambda)}$

YES! We can simulate the empirical relationship with a quit simple bin model.

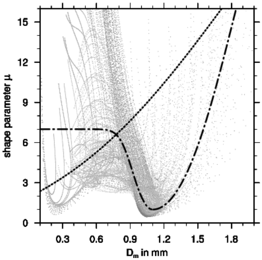
Seifert (2005), JAM

The shape parameter of the raindrop distribution



Adding evaporation to the problem leads to more scatter in the μ-λ-relation.
Using a μ-D-relation instead of μ-λ allows to distinguish large and small mean diameters

$$\mu = \begin{cases} 6\tanh\{[c_1\,(D_m - D_{eq})]^2\} + 1, & D_m \leq D_{eq} \\ 30\tanh\{[c_2\,(D_m - D_{eq})]^2\} + 1, & D_m > D_{eq} \end{cases}$$

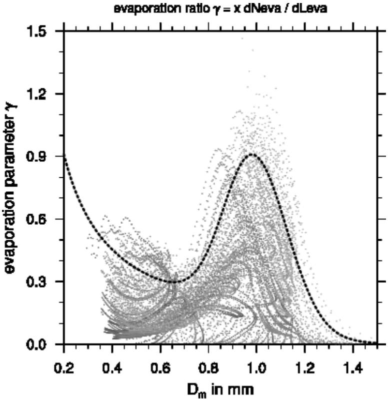


- Low μ for D ≈ 1 mm: „breakup/coalescence regime“
- Large μ for D >> 1 mm: „gravitational sorting regime“
- Large uncertainty for small mean diameters: evaporation, gravitational sorting,…

Not yet in UCLA-LES level=3 or 4, only level=5

Seifert (2008), JAS

The size effect of evaporation



Using the spectral bin model, an empirical parameterization of the size effect of evaporation can be derived:

$$\left.\frac{\partial N_r}{\partial t}\right|_{\mathrm{eva}} = \gamma \, \frac{N_r}{L_r} \, \left.\frac{\partial L_r}{\partial t}\right|_{\mathrm{eva}}$$

with

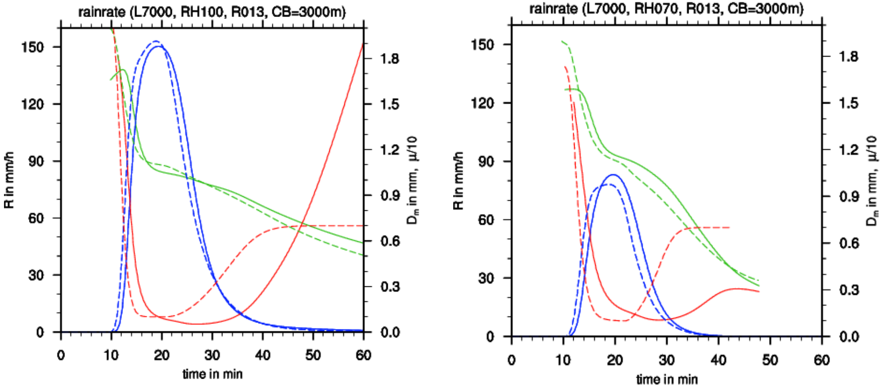
$$\gamma = \frac{D_{eq}}{D_m} \exp(-0.2\mu)$$

Not yet in UCLA-LES level=3 or 4, only level=5

Seifert (2008), JAS

Results of the revised two-moment scheme in a 1D rainshaft model

Comparison of the spectral bin and the two-moment bulk model for an strong rain event (rain rate **R**, mean diameter $\mathbf{D_m}$ and shape parameter $\boldsymbol{\mu}$)



Note: No overshooting or any other artifacts in rainrate

Seifert (2008), JAS

Microphysics sensitivities in UCLA-LES

Columns are cloud droplet number concentration, liquid water path, rain water path, cloud cover, inversion height, surface rain rate, max. rain rate and number of raindrops

N_c	Microphysics	$\mathcal{L}$	$\mathcal{R}$	C	z_i	R	R_{mx}	N_R
35	SB	13.0	16.8	0.13	2183	36.5	53.1	25.9
70		17.4	17.3	0.14	2368	42.3	50.1	16.9
105		20.0	6.5	0.17	2477	11.6	19.1	10.4
140		19.8	3.9	0.18	2494	8.1	11.4	7.5
35	KK	14.7	30.5	0.11	2271	37.5	86.7	16.9
70		20.3	3.1	0.18	2506	2.3	9.0	5.0
105		20.5	1.4	0.18	2527	1.8	4.4	3.4
140		20.9	0.4	0.19	2508	0.3	1.1	3.0

Name	Microphysics	$\mathcal{L}$	$\mathcal{R}$	C	z_i	R	R_{mx}	N_R
S01	SB	17.4	17.3	0.14	2368	42.3	50.1	16.9
S02	SB-$\mu = 0$	16.6	6.8	0.16	2357	18.0	24.9	14.9
S03	SB-$\mu = 5$	18.9	19.4	0.15	2368	31.0	53.6	18.4
S04	SB-$\mu = 10$	18.3	22.7	0.15	2401	42.4	60.2	16.7
S05	(SB, SB, KK)	16.8	20.8	0.13	2431	40.3	56.4	16.8
S06	SB (no SC)	22.8	38.1	0.17	2452	27.6	85.9	64.0
S07	(SB, KK, SB)	20.4	20.3	0.15	2273	42.9	99.3	57.0
S08	KK	20.3	3.1	0.18	2506	2.3	9.0	5.0
S09	(SB, KK, KK)	22.4	55.6	0.16	2348	18.1	95.5	58.3
S10	(KK, SB, SB)	18.9	1.6	0.18	2505	2.7	7.0	3.9
S11	S03 with Breakup	16.2	19.6	0.13	2335	36.5	53.0	16.7
S12	S11 with Ventilation	14.7	15.6	0.11	2336	23.0	62.9	21.6

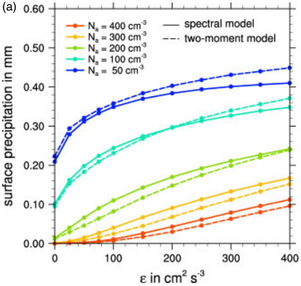
Read the paper!

Stevens and Seifert (2008), Journal of the Meteorological Society of Japan

Turbulence effect on warm rain

Turbulence can enhance the collision frequency of droplet. This can be included in the SB warm rain scheme and is included in UCLA-LES.

rainshaft model



UCLA-LES RICO simulations

Table II. Sensitivity to turbulence-enhanced coalescence (T), versus no turbulence enhancement (NT), for cloud droplet number concentrations N_c = 70, 140 and 300 mg^{-1}. NT-140-hr and T-140-hr represent simulations with doubled horizontal resolution (grid spacing of 50 m).

Run	L (gm^{-2})	R (gm^{-2})	z_i (m)	C	R_{sfc} (W m^{-2})	R_{max} (W m^{-2})	N_r (dm^{-3})
NT-70	18.6	7.0	2418	0.17	8.6	16.6	19.7
T-70	19.3	22.2	2358	0.15	43.3	51.6	26.6
NT-140	18.9	0.8	2449	0.17	0.8	2.0	8.7
T-140	19.7	8.3	2422	0.17	13.2	18.8	14.9
NT-140-hr	21.1	1.0	2422	0.21	1.1	2.6	8.9
T-140-hr	21.9	3.9	2399	0.21	4.9	9.9	10.9
NT-300	20.2	0.0	2442	0.17	0.0	0.0	4.7
T-300	18.3	0.4	2438	0.16	0.4	0.9	6.4

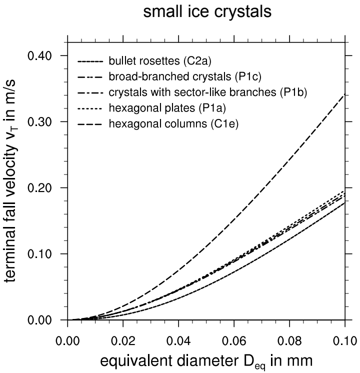
Variables are cloud (liquid) water path L, rain water path R, inversion height z_i, fraction of cloudy columns C, rain-drop number concentrations averaged over raining regions N_r, surface rain rate R_{sfc}, and the maximum rain-rate R_{max} within the (domain-averaged) profile of rain-rate.
All variables are averaged over the last four hours of each simulation.
A rain rate of 29 W m^{-2} corresponds to 1 mm day^{-1}.

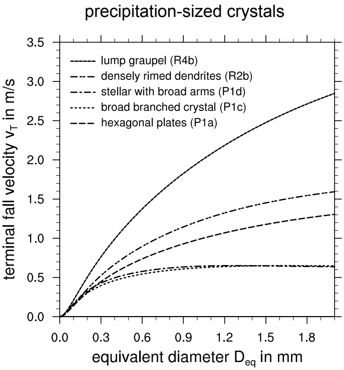
Again, read the paper!

Seifert, Nuijens and Stevens (2010), QJ

Ice particle fall speeds

➔ For the parameterization of sedimentation and growth rates the terminal fall velocity of the particles is of greatest importance





➔ Often used for ‚tuning', because actual particles habits for a case are usually not known and can hardly be predicted.

Ice nucleation

- ➔ Homogeneous nucleation from vapor: Does not occur in the atmosphere!
- ➔ Homogeneous freezing of cloud droplets: at about -37 C, can occur in stron deep convective updrafts
- ➔ Homogeneous freezing of liquid aerosols: colder than -37, but below RH=100 %, may be the main mechanism for cirrus formation. Does still need high ice supersaturation of 140-170 %.
- ➔ Heterogeneous freezing: needs ice nuclei (IN), for each specific IN strongly temperature and RH dependent. Usually we don‘t know the IN distribution and have to make ad-hoc choices, e.g., climatology.
- ➔ Different modes of heterogeneous nucleation: immersion freezing, deposition nucleation, contact nucleation, etc.
- ➔ The importance of different substances, e.g., soot, dust, or organics is still under debate. Aerosol age, and aerosol processing does play a role (also for CCN).

My personal advice: Stay away from ice clouds, if you can!

If you can not for some reason, then make at least sure that your results are not overly dependent on your IN choices, and get some f**** observations.**

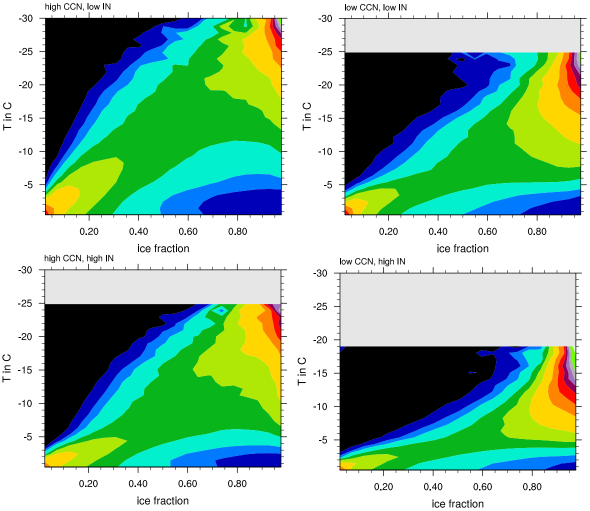
Glaciation of clouds

➔ more IN lead to a more efficient glaciation

➔ less CCN lead to a more efficient glaciation, because large drizzle drops or rain have a higher freezing probability

➔ Therefore your choices of CCN and IN matter for the cloud dynamics

PDF of IWC/TWC as a function of temperature



Seifert et al. (2011), ACPD

UCLA-LES microphysics schemes

level=2: Pure condensation

level=3: Bjorn‘s warm rain scheme based on SB 2001

- two-moment rain as described in Stevens and Seifert (2008).
- *Code is short and easy to understand.*

level=4: Thijs‘ mixed-phase scheme

- one-moment snow and graupel, two-moment rain and ice,
- works fine for bubble convection, but not yet tuned for other cases.
- *Code is well organized, but not documented*

level=5: Axel‘s two-moment mixed-phase scheme with hail,

- everything two-moment, well tested on 1-3 km grids, i.e. COSMO model,
- but not much experience with LES cases.
- Scheme is very modular, many choices, more extensions, i.e. process parameterizations, are available, e.g. more ice nucleation or CCN activation schemes.
- *Code got quite messy recently and may be a bit confusing, but the structure is still okay. Several published papers that describe the schemes.*

Radiation

UCLALES Tutorial

Bjorn Stevens

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Excercises

UCLALES Tutorial

Bjorn Stevens

Max Planck Institute for Meteorology

November 7 - 11, 2011

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

The Dry Convective Boundary Layer

- Run the code with `uclales/misc/initfiles/namelist_drycbl`
- Process the statistics with `reduceps` and `reducets`
- Stitch the crosssections together with `cdo gather`
- Plot with `ncview`, `ncl`, the scripts in `uclales/misc/analysis`, or your program of choice

Questions I

- What are the profiles of the 3 velocity components? Do you understand that?
- There are 3 different ways of defining the boundary layer height `zi`:
  - The maximum gradient in θ_l
  - The maximum variance in θ_l
  - The minimum buoyancy flux
- What are the differences?
- The encroachment rate is equal to:

$$z_{enc}(t) = \sqrt{\frac{2Ft}{\Gamma}}$$

with F the surface heat flux and Γ the temperature lapse rate. How does z_i compare with z_{enc}? What is the difference?

Questions II

- Look at the variances: `u2, w2, t2`. What do they look like? What is/is not with what you expect from Boundary Layer theory?
- Look at the vertical flux profiles, and in particular `tot_tw` and `sfs_tw`.
- Finally, compare the advective tendency (`adv_u`) with the diffusion(`dff_u`). What do you notice? Would you say that the LES is well resolved? Where / why (not)?

Questions III

- **Optional, to be done after the Statistics class:** It would be very useful to have conditional sampling of the thermal updrafts. Unfortunately, they are not in the `.ps` file at the moment. As a (lengthier) exercise, we are going to do that here.
- Open the files `ncio.f90` and `stat.f90`. First, have a look at `stat.f90`
- The name of a `ps` variable is defined in `s2` from line 52 on. This includes the `cs2` variables for buoyant cloud conditional sampling. Append `cs3` variables for (at least) w and tv at the end of the array. Raise `nvar2` at l.33 accordingly.
- Make sure you know the number of your new variables.
- The conditional sampling for cloud water is done in subroutine `accum_lvl2` between lines 604 and 658. Look at those in depth.

Questions IV

- The function `get_avg` creates an average over the 2 horizontal direction out of a 3D array.
- The function `get_csum` creates a conditional sum over an array, on places where the final array is 1
- Use these lines for a conditional sampling of dry thermals. Put it in subroutine `accum_lvl1`
- In `ncio.f90` the variable output names, longnames and units are provided. Use the code from line 989 on as an example to add your variables.
- That should be all: Try and compile. Now it gets time to debug.

BOMEX Shallow Cumulus

- Check `articles/siebesma2003.pdf` for the initial settings of BOMEX
- Build a `NAMELIST` based on it. Hint: the `RICO` Namelist should be a good starting point
- Run the run, postprocess like the Dry CBL run
- If successful, commit your NAMELIST to git
- Rerun your run with a different name, but with `level=3` for microphysics in the NAMELIST

Questions I

- Plot the cloud fraction and the cloud cover. What is the difference between the two?
- What are cloud base and cloud top? There are several cloud bases/tops in the `.ts` file. What is the difference between them? What can we (implicitly) learn about these clouds based upon these numbers?
- One classical way of parametrizing (shallow) cumuli in large scale models, is to model the transport through the cloud layer with a mass flux approach. If necessary, read up on it in `siebesma1995.pdf`. They found that entrainment and detrainment rates in the large scale models were off by an order of magnitude.
- Try and reproduce figures 6 and on from that study using the output of the `.ps` file. `_cs1` is the conditional sampling over the cloud. `_cs2` is the conditional sampling for the buoyant part of the cloud.

Questions II

- BOMEX was an intercomparison case of non-precipitating cumulus clouds. Is the non-precipitating really true, or just because of a lack of microphysical models a decade ago?
- If precipitation is present, does it matter?

DYCOMS RF02 Stratocumulus

- The Dycoms Stratocumulus case is described in `ackerman2009.pdf`
- Done with $70cm^{-3}$ CCN and prescribed radiation.
- Is the cloud layer sensitive to these kind of choices?
- The autoconversion rate can be switched to the Khairoutdinov/Kogan scheme (optimized for Stratocumulus) or Seifert Beheng (more general). Any difference?
- Compare with the results from the paper