Using the UCLA Large Eddy Simulation code

Thijs Heus, Bjorn Stevens, Axel Seifert, Cathy Hohenegger

Max Planck Institute for Meteorology

November 7 - 11, 2011

Overview of UCLA LES

UCLALES Tutorial

Thijs Heus

Max Planck Institute for Meteorology

November 7 - 11, 2011

This Week

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:30</td>
<td>ClimateServiceCenter 112</td>
<td>Bjorn - Introduction</td>
<td>Thijs - Setting Up</td>
<td>Geomatikum 11.35</td>
<td>Assembly</td>
</tr>
<tr>
<td>10:00</td>
<td>ClimateServiceCenter 112</td>
<td>Thijs - Code Structure</td>
<td>ClimateServiceCenter 112</td>
<td>Geomatikum 11.35</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>11:00</td>
<td>ZMAW 301</td>
<td>Thijs - Code Structure</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>12:00</td>
<td>ClimateServiceCenter 112</td>
<td>Executing the code and building a case.</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>13:00</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>14:00</td>
<td>Thijs - Code Structure</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>15:00</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>16:00</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>17:00</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
<td>SMASH 001</td>
</tr>
<tr>
<td>18:00</td>
<td>Icebreaker/Beer/...</td>
<td>Dinner</td>
<td>Dinner</td>
<td>Dinner</td>
<td>Dinner</td>
</tr>
</tbody>
</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

#### General Circulation Models
- **Domain size:** Entire Earth
- **Horizontal Boundary conditions:** None
- **Horizontal grid spacing:** 50 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

#### Regional Models
- **Domain size:** Continental scale or smaller
- **Studies of organization, deep systems,...**
- **Horizontal Boundary conditions:** Nested/forced by GCM
- **Horizontal grid spacing:** 5 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 km
- **Studies of boundary layer processes, idealized (and not so idealized) clouds**
- **Horizontal Boundary conditions:** Periodic
- **Horizontal grid spacing:** 50 m
- **Total number of points:** about $400 \times 400 \times 100$
- **Simulation duration:** Hours/Days
- **Resolved:** Shallow Clouds, Boundary layers
- **Parameterized:** Turbulence, Surface, Microphysics
Cascade of Models
Direct Numerical Simulations

- Domain size: 1m
- Studies of turbulence, possibly with interactions of other processes
- Horizontal Boundary conditions: Periodic
- Horizontal grid spacing: 1mm
- Total number of points: about 1000 × 1000 × 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

Overview of UCLA LES

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

\[ \bar{u} = \int G(r) u(r) dr \]

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

\[
\frac{\partial u_i}{\partial t} = -u_j \frac{\partial u_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \pi}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i
\]

Large-Eddy Equations

\[
\begin{align*}
\frac{\partial \bar{u}_i}{\partial t} &= -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + F_i \\
\frac{\partial \bar{\rho}}{\partial t} &= -\bar{u}_j \frac{\partial \bar{\rho}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \gamma_{ij})}{\partial x_j} + S_{\phi} \\
\end{align*}
\]

Anelastic continuity

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

Ideal gas law equation of state

\[
\theta_v = \theta (1 + (R_v/R_d - 1)q_t - (R_v/R_d)q_l).
\]

Closure

- \( \tau_{ij} \equiv \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j \) 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.

History

- 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 Stratocumulus: Moeng et al. (1996)
- Radiative Smoke: Bretherton et al. (1999)
- Non-Precip Shallow Cu: Siebesma et al. (2003)
- Non-Precip Stratocumulus: Stevens et al. (2001)
- Diurnal Cycle Cu: Brown et al. (2001)
- Precip Stratocumulus: Ackerman et al. (2008)

History II

Intercomparisons

- Precip Cumulus: van Zanten et al. (2011)
- Precip Stratocumulus: Ackerman et al. (2008)
- Radiative, transition runs: Sandu, de Roode, Blossey (2012)

Overview of UCLA LES

Large-Eddy Simulations

History

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 conditions)

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

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
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.

Overview of UCLA LES

Large-Eddy Simulations

History

Overview of UCLA LES

Max Planck Institute for Meteorology

November 7 - 11, 2011

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

UCLALES Tutorial

Thijs Heus

Contents:
- A public SSH key (for gitorious.org)
- A directory with the lectures (will be updated)
- A directory with supplementary material (e.g. articles to read)
- A directory to run your runs

Do not overwrite these files - they will be updated

Not yet here: The source code

Feel free to do the course on your own account/machine!
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

Using Git

Obtaining the code

- 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 course work will be done based on the course branch, so change:
git checkout course

- Some differences appear there

- Now make your personal branch, based on the course branch: git
  checkout -b yourname

- Here you can play whatever you like
Using Git
Changing something

- Open the file test1
- Write something in it
- See what is different: `git status` and `git diff`
- If you are happy with your 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!

Using Git
Creating a new file

- 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 your 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!

Using Git
Updating the remote repository

- 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

Using Git
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
Compilation Requirements

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

Setting up

Git

Compilation

There are two ways of compiling the code.

- CMake does its best to create a Makefile automatically.
- There are a bunch of predefined Makefiles available in the misc/makefiles directory.

CMake

- The CMakeLists.txt file in the uclales dir sets all the options.
- 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.
- The CMakeLists.txt file in the uclales dir sets all the options.
- 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.

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

There are two ways of compiling the code.

- CMake does its best to create a Makefile automatically.
- There are a bunch of predefined Makefiles available in the misc/makefiles directory.

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales

CMake

- Run CMake to create the makefile: cmake -D MPI=FALSE ..
- make -j4 to build the binary uclales
Compilation

CMake options

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI</td>
<td>TRUE, FALSE</td>
<td>Switch between parallel and serial</td>
</tr>
<tr>
<td>CMAKE_BUILD_TYPE</td>
<td>DEBUG, RELEASE</td>
<td>Switch between debug settings and optimized</td>
</tr>
<tr>
<td>PROFILER</td>
<td>GPROF, SCALASCA, MARMOT</td>
<td>Switch on profiler (to assess speed bottleneck)</td>
</tr>
</tbody>
</table>

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

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
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.

**Available runs II**

- **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 using zgrid.gcsc9.f.
- **namelist_rico**: The RICO GCSS composite case.
- **namelist_smoke**: The GCSS smoke case.

**Data files**

- **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.
Data files

Data files

ls_flux_in

Time dependent fluxes and large scale forcings.

- The first block sets the surface values, with columns:
  - Time in seconds
  - Surface heat flux in $W/m^2$
  - Surface moisture flux in $W/m^2$
  - 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 $\theta_l$
  - Large scale tendency for $q_t$

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

Data files

nudge_in

Nudges the average fields to a preset value:

$$\frac{\partial \phi}{\partial t} = \frac{\phi_{\text{nudge}} - \bar{\phi}}{\tau}$$

With $\tau^{-1}$ the nudging strength.

The columns depict:

- height in meters
- Nudging strength
- The nudging value of $u$, $v$, $\theta_l$ 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
filprf | x
nwp | 132
nyp | 132
nzp | 105
deltax | 35.0 m
deltay | 35.0 m
deltaz | 17.5 m
dzrat | 1.02
dzmax | 1200 m
digrdtyp | 1
dtlong | 10 s
hfilin | test.
timmax | 18000 s

Model Options

Data files

Namelist

---

Grid and Time setup II

**Variable** | **Default**
--- | ---
deltaz | 17.5 m
dzrat | 1.02
dzmax | 1200 m
igrdtyp | 1
dtlong | 10 s
hfilin | test.
timmax | 18000 s

Model Options

Data files

Namelist

---

Grid and Time setup III

**Variable** | **Default**
--- | ---
wctime | Wall clock time to break off the simulation
nfpt | 5
distim | 300 s
naddsc | 0
runtyp | INITIAL

Model Options

Data files

Namelist

---

Physics I

**Variable** | **Default**
--- | ---
iradtyp | 0
CCN | cloud droplet mixing ratio
level | 0
Corflg | coriolis acceleration (true/false)
radfrq | radiation update interval
strtim | GMT of model time

Model Options

Data files

Namelist

---
### Physics II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cntlat</td>
<td>31.5° N</td>
<td>model central latitude</td>
</tr>
<tr>
<td>case_name</td>
<td>astex</td>
<td>specify case name (rico, astex, bomex)</td>
</tr>
<tr>
<td>lsvarflg</td>
<td>false</td>
<td>reads large scale forcings from the file lscale.in</td>
</tr>
<tr>
<td>div</td>
<td>3.75e-6</td>
<td>divergence</td>
</tr>
<tr>
<td>umean</td>
<td>0.</td>
<td>Mean $U$ velocity (subtracted during the calculations)</td>
</tr>
<tr>
<td>vmean</td>
<td>0.</td>
<td>Mean $V$ velocity (subtracted during the calculations)</td>
</tr>
<tr>
<td>th00</td>
<td>288</td>
<td>Basic state temperature (subtracted during the calculations)</td>
</tr>
</tbody>
</table>

### Physics III

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sst</td>
<td>292 K</td>
<td>sea surface temperature</td>
</tr>
<tr>
<td>isfctyp</td>
<td>0</td>
<td>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.</td>
</tr>
<tr>
<td>ubmin</td>
<td>0.20</td>
<td>minimum $u$ for $u_*$ computation</td>
</tr>
<tr>
<td>zrough</td>
<td>0.1</td>
<td>momentum roughness height (if less than zero use Charnock relation)</td>
</tr>
<tr>
<td>dthcon</td>
<td>100 Wm$^{-2}$</td>
<td>surface temperature gradient (isfcflg=1) or surface heat flux (itsflg=0)</td>
</tr>
<tr>
<td>drtcon</td>
<td>0 Wm$^{-2}$</td>
<td>surface humidity (mixing ratio) gradient (isfcflg=1) or surface latent heat flux (itsflg=0)</td>
</tr>
<tr>
<td>csx</td>
<td>0.23</td>
<td>Smagorinsky Coefficient</td>
</tr>
</tbody>
</table>

### Physics IV

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prndtl</td>
<td>1/3</td>
<td>Prandtl Number (if less than zero no sgs for scalars)</td>
</tr>
<tr>
<td>sfc_albedo</td>
<td></td>
<td>Albedo of the surface</td>
</tr>
<tr>
<td>lnudge</td>
<td></td>
<td>Switching on/off nudging</td>
</tr>
<tr>
<td>tnudgefac</td>
<td></td>
<td>Factor to strengthen the nudging</td>
</tr>
<tr>
<td>ltimedep</td>
<td></td>
<td>Switch for time depend fluxes and large scale forcings</td>
</tr>
<tr>
<td>SolarConstant</td>
<td></td>
<td>Top of Atmosphere radiation</td>
</tr>
</tbody>
</table>

### Initial profiles I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipsflg</td>
<td>1</td>
<td>control parameter for input sounding (0: pressure in hPa; 1: height in meters with ps(1)=psfc)</td>
</tr>
<tr>
<td>itsflg</td>
<td>1</td>
<td>control parameter for input sounding (0: $ts = \theta$; 1: $ts = \theta_i$)</td>
</tr>
<tr>
<td>irsflg</td>
<td>1</td>
<td>control parameter for input sounding (0: $rs = \text{Rel. Hum}$; 1: $(rs = q_i)$)</td>
</tr>
<tr>
<td>us</td>
<td>n/a</td>
<td>input zonal wind sounding</td>
</tr>
<tr>
<td>vs</td>
<td>n/a</td>
<td>input meridional wind sounding</td>
</tr>
<tr>
<td>ts</td>
<td>n/a</td>
<td>input temperature sounding</td>
</tr>
</tbody>
</table>
### Initial profiles II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rts</td>
<td>n/a</td>
<td>n/a input humidity sounding</td>
</tr>
<tr>
<td>ps</td>
<td>n/a</td>
<td>n/a input pressure sounding</td>
</tr>
<tr>
<td>hs</td>
<td>n/a</td>
<td>n/a vertical position</td>
</tr>
<tr>
<td>iseed</td>
<td>0</td>
<td>random seed</td>
</tr>
<tr>
<td>zrand</td>
<td>200 m</td>
<td>height below which random perturbations are added</td>
</tr>
</tbody>
</table>

### Statistics and output I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outflg</td>
<td>true</td>
<td>output flag (true/false)</td>
</tr>
<tr>
<td>lsync</td>
<td>false</td>
<td>Synchronize the crosssection output (true/false)</td>
</tr>
<tr>
<td>frqhis</td>
<td>9000 s</td>
<td>history write interval</td>
</tr>
<tr>
<td>frqanl</td>
<td>3600 s</td>
<td>analysis write interval</td>
</tr>
<tr>
<td>slcll</td>
<td>false</td>
<td>write slice output (true/false)</td>
</tr>
<tr>
<td>istpfl</td>
<td>1</td>
<td>print interval for timestep info</td>
</tr>
</tbody>
</table>

### Statistics and output II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssam_intvl</td>
<td>30 s</td>
<td>statistics sampling interval</td>
</tr>
<tr>
<td>savg_intvl</td>
<td>1800 s</td>
<td>statistics averaging interval</td>
</tr>
<tr>
<td>lcross</td>
<td>false</td>
<td>Crosssection output (true/false)</td>
</tr>
<tr>
<td>frqcross</td>
<td>3600 s</td>
<td>Crosssection write interval</td>
</tr>
<tr>
<td>lxy</td>
<td>false</td>
<td>Crosssection output in xy plane (true/false)</td>
</tr>
<tr>
<td>zcross</td>
<td>0</td>
<td>Crosssection location of xy plane (true/false)</td>
</tr>
<tr>
<td>lxz</td>
<td>false</td>
<td>Crosssection output in xz plane (true/false)</td>
</tr>
<tr>
<td>ycross</td>
<td>0</td>
<td>Crosssection location of xy plane (true/false)</td>
</tr>
<tr>
<td>lyz</td>
<td>false</td>
<td>Crosssection output in yz plane (true/false)</td>
</tr>
<tr>
<td>xcross</td>
<td>0</td>
<td>Crosssection location of xy plane (true/false)</td>
</tr>
<tr>
<td>lwaterbudget</td>
<td>false</td>
<td>Crosssection of (costly) waterbudget (true/false)</td>
</tr>
</tbody>
</table>
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-Eddington 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., \( \theta_l^l, q_t^l, \rho_0, \pi_0, \Theta_0 \).
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 \( r_l^n, \partial_t r_l \)
a_rpp,a_rpt  " Rain mass mixing ratio \( r_l^n, \partial_t r_l \) (for level 3)
a_npp,a_npt  " Rain number mixing ratio, \( n_l^n, \partial_t n_r \) (for level 3)

Main Variables II

a_theta  " Potential temperature, \( \theta \) (diagnosed from model state)
r_c,r_v  " 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 (\( p, \pi \) respectively)
a_scr1,a_scr2  " Three dimensional scratch arrays
a_ustar,a_tstar,a_rstar  2D Surface scales, \( u_s, \theta_s, r_s \) respectively
uw_sfc, vv_sfc, ww_sfc  " Surface momentum fluxes, \( \overline{w'}, \overline{v'w'}, \overline{w'w'} \) respectively.
ww_sfc  " Surface thermodynamic fluxes, \( \overline{w'\theta'}, \overline{w'r'} \) respectively.
Main Variables III

- precip: Precipitation flux
- $d n_0$: 1D Basic state density, $\rho_0(z)$.
- $x_t, y_t, z_t$: Position of thermodynamic points
- $x_m, y_m, z_m$: Position of momentum points
- $d z_i$:  
  - $d z_i(t) = 1/(z_t(k + 1) - z_t(k))$
  - $d z_i(m) = 1/(z_m(k) - z_m(k - 1))$

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 an 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 the 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.

Output files II

- 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outflg</td>
<td>true</td>
<td>output flag (true/false)</td>
</tr>
<tr>
<td>lsync</td>
<td>false</td>
<td>Synchronize the crosssection output (true/false)</td>
</tr>
<tr>
<td>frqhis</td>
<td>9000 s</td>
<td>history write interval</td>
</tr>
<tr>
<td>frqanl</td>
<td>3600 s</td>
<td>analysis write interval</td>
</tr>
<tr>
<td>slcflg</td>
<td>false</td>
<td>write slice output (true/false)</td>
</tr>
<tr>
<td>istpfl</td>
<td>1</td>
<td>print interval for timestep info</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssam_intvl</td>
<td>30 s</td>
<td>statistics sampling interval</td>
</tr>
<tr>
<td>savg_intvl</td>
<td>1800 s</td>
<td>statistics averaging interval</td>
</tr>
<tr>
<td>lcross</td>
<td>false</td>
<td>Crosssection output (true/false)</td>
</tr>
<tr>
<td>frqcross</td>
<td>3600 s</td>
<td>crosssection write interval</td>
</tr>
<tr>
<td>lxy</td>
<td>false</td>
<td>Crosssection output in xy plane (true/false)</td>
</tr>
<tr>
<td>zcross</td>
<td>0</td>
<td>Crosssection location of xy plane (true/false)</td>
</tr>
<tr>
<td>lxz</td>
<td>false</td>
<td>Crosssection output in xz plane (true/false)</td>
</tr>
</tbody>
</table>

Statistics and output II
Statistics and output III

<table>
<thead>
<tr>
<th>ycross</th>
<th>0</th>
<th>Crosssection location of xy plane (true/false)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lyz</td>
<td>false</td>
<td>Crosssection output in yz plane (true/false)</td>
</tr>
<tr>
<td>xcross</td>
<td>0</td>
<td>Crosssection location of xy plane (true/false)</td>
</tr>
<tr>
<td>lwaterbudget</td>
<td>false</td>
<td>Crosssection of (costly) waterbudget (true/false)</td>
</tr>
</tbody>
</table>

### 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/netcdf/lib/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.
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

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

---

**The LES Equations**

Other forces

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

\[
\begin{align*}
\frac{\partial \bar{u}_i}{\partial t} &= -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \pi}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + F_i \\
\frac{\partial \bar{\phi}}{\partial t} &= -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \gamma_{ij})}{\partial x_j} + S_{\phi}
\end{align*}
\]

Anelastic continuity

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

Ideal gas law equation of state

\[
\theta_v = \theta \left[ 1 + \left( \frac{R_v}{R_d} - 1 \right) q_t - \frac{R_v}{R_d} q_t \right].
\]
Time stepping is based on a Runge-Kutta third order method. The tendencies are calculated through 3 iterations:

\[
\begin{align*}
\phi^n_* &= \phi^n + \alpha_1 \frac{\partial \phi^n}{\partial t} \Delta t \\
\phi^{n*} &= \phi^n_* + \alpha_2 \frac{\partial \phi^n}{\partial t} \Delta t + \beta_2 \frac{\partial \phi^n}{\partial t} \Delta t \\
\phi^{n+1} &= \phi^{n*} + \alpha_3 \frac{\partial \phi^{n*}}{\partial t} \Delta t + \beta_3 \frac{\partial \phi^n}{\partial t} \Delta t
\end{align*}
\]

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

The timestep \( \Delta t \) (or \( dt \)) in the code is variable:
- Bounded by the Courant criterion (\( CFL = 0.5 \))
- Bounded by \( dt_{long} \) in NAMELIST. Use it for:
  - Unstabilities 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.

### The LES Equations

**Other forces**

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

\[
\begin{align*}
\frac{\partial \bar{u}_i}{\partial t} &= -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + F_i \\
\frac{\partial \bar{\phi}}{\partial t} &= -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \gamma \phi_j)}{\partial x_j} + S_{\phi}
\end{align*}
\]

Anelastic continuity

\[
\frac{\partial (\rho_0 u_i)}{\partial x_j} = 0
\]

Ideal gas law equation of state

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

**Advection**

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

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

with \( F_{i+\frac{1}{2}} \) the flux through the cell boundary at \( i + \frac{1}{2} \).
In UCLALES, we do 4th order Central Differencing for momentum advection, and flux-limited advection for scalars to guarantee positive values:

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

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

With \( \kappa_{i+\frac{1}{2}} > 0 \) and a function of consecutive gradients (assuming \( u_{i+\frac{1}{2}} \)):

\[ r = \frac{\phi_{i+1} - \phi_i}{\phi_j - \phi_{j-1}} \]

Flux limiter schemes

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

- **minmod** \( \min(r, 1) \)
- **superbee** \( \max(\min(2r, 1), \min(r, 2)) \)
- **MC** \( \min(2r, \frac{1 + r}{2}, 2) \)
- **vanLeer** \( \frac{r + |r|}{1 + |r|} \)

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

**Other forces**

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

\[
\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_P \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + F_i
\]

\[
\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j}
\]

Anelastic continuity

\[
\frac{\partial (\rho_0 u_i)}{\partial x_j} = 0
\]

Ideal gas law equation of state

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

### Diffusion I

The sub-grid fluxes \( \tau_{ij} \) and \( \gamma_{ij} \) 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

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

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

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

### Diffusion II

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

\[
\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_P \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \tau_{ij})}{\partial x_j} + F_i
\]

\[
\frac{\partial \bar{\phi}}{\partial t} = -\bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j}
\]

Anelastic continuity

\[
\frac{\partial (\rho_0 u_i)}{\partial x_j} = 0
\]

Ideal gas law equation of state

\[
\theta_v = \theta \left[ 1 + \left( \frac{R_v}{R_d} - 1 \right) q_t - \frac{R_v}{R_d} q_i \right].
\]
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 \( z \) (\( \Theta_0 \) being constant).

For gravity, we use buoyancy deviations from the slab average (not the basic state). For consistency, introduce a second exner \( \pi_1 \):

\[
\frac{d}{dz}(\pi_0 + \pi_1) = -\frac{g}{c_p \Theta_0}.
\]

### Calculating Pressure I

Start with continuity:

\[
\frac{\partial(\rho_0 \bar{u}_i)}{\partial x_i} = 0
\]

And the momentum equation:

\[
\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial (\rho_0 \bar{\tau}_{ij})}{\partial x_j} + F_i
\]

Fill them in in each other:

\[
\frac{\partial}{\partial x_i} \left( \rho_0 \frac{\partial \bar{u}_i}{\partial t} \right) = \frac{\partial}{\partial x_i} \left[ -\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - \rho_0 c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} + \frac{\partial (\rho_0 \bar{\tau}_{ij})}{\partial x_j} + \rho_0 F_i \right] = 0
\]

Bring the pressure to the other side:

\[
\frac{\partial}{\partial x_i} \left( \rho_0 c_p \Theta_0 \frac{\partial \bar{\pi}}{\partial x_i} \right) = \frac{\partial}{\partial x_i} \left[ -\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial (\rho_0 \bar{\tau}_{ij})}{\partial x_j} + \rho_0 F_i \right]
\]

And we end up with a Poisson equation:

\[
\frac{\partial}{\partial x_i} \left( \rho_0 \frac{\partial \bar{\pi}}{\partial x_i} \right) = \frac{1}{c_p \Theta_0} \frac{\partial}{\partial x_i} \left[ -\rho_0 \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial (\rho_0 \bar{\tau}_{ij})}{\partial x_j} + \rho_0 F_i \right]
\]

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

\[ h \]

\[ z \ll h \]

\[ z \gg z_0 \]

Variables to be determined

Surface stress in \( x \)

\[ \tau_0 = -\rho \frac{\partial u}{\partial z} \]

Surface stress in \( y \)

\[ \tau_0 = -\rho \frac{\partial v}{\partial z} \]

Surface sensible heat flux

\[ H_0 = \rho c_p \frac{\partial \theta}{\partial z} \]

Surface buoyancy flux

\[ H_v = \rho c_p \frac{\partial \theta}{\partial z} \]

Surface latent heat flux

\[ LE_0 = \rho L \frac{\partial q}{\partial z} \]

How do we compute the fluxes?

Use of surface similarity theory:

• Only valid for the surface layer \( z_0 < z \ll h \) 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 \( z_0 < z \ll h \) 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**

<table>
<thead>
<tr>
<th>Friction velocity</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature scale</td>
<td>θ₀</td>
</tr>
<tr>
<td>Temperature scale</td>
<td>θ₀</td>
</tr>
<tr>
<td>Humidity scale</td>
<td>0</td>
</tr>
<tr>
<td>Obukov stability length</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\frac{u}{\theta} \cdot \frac{\partial \bar{u}}{\partial z} = 1
\]

Integrate:

\[
\ln \left( \frac{z}{z_0} \right) = \frac{n}{0} \cdot 
\]

**Relations: First-order closure**

Assume neutral and homogeneous atmosphere:

\[
\begin{align*}
0 & - \frac{\partial}{\partial z} \\
\sim & 0 \\
0 & - \frac{\partial}{\partial z} \\
\frac{\partial}{\partial z} & 0
\end{align*}
\]

Integrate:

\[
\begin{align*}
0 & - n \\
0 & 0
\end{align*}
\]
Relations: First-order closure

In general:

\[ \frac{\partial}{\partial z} \left( \frac{z}{2} - \frac{\partial \bar{v}}{\partial \bar{v}} \right) \]

Integrate:

\[ \frac{\partial}{\partial z} \left( \frac{z}{2} - \frac{\partial \bar{v}}{\partial \bar{v}} \right) \int - \, d \bar{v} \]

Form of the functions

Following Garrat:

\[ \frac{\partial}{\partial z} \left( \frac{z}{2} - \frac{\partial \bar{v}}{\partial \bar{v}} \right) \]

Businger et al. (1971)
Form of the functions

Following Garrat:

Alternative: bulk transfer relations

Alternative: bulk transfer relations
Alternative: aerodynamic resistance

\[ \frac{\partial^2 \theta}{\partial z^2} = \frac{\theta}{\theta_0} - \frac{\partial \theta}{\partial z} \]

\[ \frac{n}{n_0} = \frac{\left( \frac{\theta}{\theta_0} - \frac{\partial \theta}{\partial z} \right)}{n_0} \]

\[ \frac{\left( \frac{\theta}{\theta_0} - \frac{\partial \theta}{\partial z} \right)}{n_0} = \frac{\left( \frac{\theta}{\theta_0} - \frac{\partial \theta}{\partial z} \right)}{n_0} \]

\[ \frac{\theta_0 - \theta}{\theta_0} = \frac{\theta_0 - \theta}{\theta_0} \]

The surface subroutine in UCLALES

- Case default:
  - sensible and latent heat fluxes prescribed,
  - moment fluxes diagnosed from

- Case 1:
  - gradient in temperature and moisture prescribed,
  - sensible and latent heat fluxes from
  - momentum fluxes from

- Case 2:
  - surface temperature and moisture prescribed
  - sensible and latent heat fluxes from
  - momentum fluxes from
The surface subroutine in UCLALES

- Case default:
  - sensible and latent heat fluxes prescribed, moment fluxes diagnosed from \( \frac{\partial}{\partial z} n = - \)

- Case 1:
  - gradient in temperature and moisture prescribed, sensible and latent heat fluxes from \( \frac{\partial}{\partial z} n = - \)
moment fluxes from \( \frac{\partial}{\partial z} n = - \)

- Case 2:
  - surface temperature and moisture prescribed
  - sensible and latent heat fluxes from \( \frac{\partial}{\partial z} n = - \)
moment fluxes from \( \frac{\partial}{\partial z} n = - \)

- Case 3:
  - \( C_D, C_H \)
  - surface temperature and moisture prescribed
  - sensible and latent heat fluxes from \( \frac{\partial}{\partial z} n = - \)
moment fluxes from \( \frac{\partial}{\partial z} n = - \)

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:

\[
\rho \frac{\partial}{\partial z} n = -
\]

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

Namelist options with the cases

<table>
<thead>
<tr>
<th>isfctyp</th>
<th>dthcon</th>
<th>drtcon</th>
<th>zrough</th>
<th>sst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0</td>
<td>Sensible heat flux Wm(^{-2})</td>
<td>Latent heat flux Wm(^{-2})</td>
<td>Roughness length</td>
</tr>
<tr>
<td>Case 1</td>
<td>1</td>
<td>Temperature gradient K m(^{-1})</td>
<td>Moisture gradient kg kg(^{-1}) m(^{-1})</td>
<td>Roughness length</td>
</tr>
<tr>
<td>Case 2</td>
<td>2</td>
<td>Not needed</td>
<td>Not needed</td>
<td>Roughness length</td>
</tr>
<tr>
<td>Case 3</td>
<td>3</td>
<td>( C_H )</td>
<td>( C_H )</td>
<td>( C_D )</td>
</tr>
<tr>
<td>Case 4</td>
<td>4</td>
<td>Buoyancy flux Wm(^{-2})</td>
<td>Not needed</td>
<td>Roughness length</td>
</tr>
</tbody>
</table>

To note:

- Need to know \( z_0 \) and \( z_T \). Currently \( z_0 = z_T = z_{\text{rough}} \). If \( z_{\text{rough}} \) sets to a value smaller or equal to zero, then

\[
\rho \frac{\partial}{\partial z} n = -
\]
true for ocean

- Cases 2, 3 and 4 assume to be over the ocean, i.e. \( q_0 = q_{\text{sat}}(\text{SST}) \)
- First point should be in the surface layer but higher than \( z_0 \)
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 droplets. Typical drop 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.

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.

Spectral formulation of cloud microphysics

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

$$\frac{\partial f(x, R, t)}{\partial t} + \nabla \cdot [\mathbf{v}(x, R, t) f(x, R, t)] + \frac{\partial}{\partial x} \left[ v_0(x) f(x, R, t) \right]$$

$$= \frac{\partial}{\partial x} \left[ x f(x, R, t) \right] = \sigma_{\text{coll}} + \sigma_{\text{break}}$$

with

$$\sigma_{\text{coll}} = \frac{1}{2} \int_0^x f(x - x', R, t) f(x', R, t) K(x - x', x') \, dx'$$

$$- \frac{1}{2} \int_0^x f(x, R, t) f(x', R, t) K(x, x') \, dx'$$

and

$$\sigma_{\text{break}} = \frac{1}{2} \int_0^x f(x', R, t) f(x', R, t) B(x', x'') P(x, x'') \, dx''$$

$$- \frac{1}{2} \int_0^x f(x, R, t) f(x', R, t) B(x, x') \, dx'$$.

This is the level=5 scheme in UCLA-LES

... but secondary processes, like Hallet-Mossop, are not included in the diagram.
The gravitational collision-coalescence kernel

\[ K(x, y) = \pi \left[ r(x) + r(y) \right]^2 |v(x) - v(y)| E_{\text{coll}}(x, y) E_{\text{coa}}(x, y) \]

collision efficiency:

\[ E_{\text{coll}} = \frac{y^2}{(R + r)^2} \]

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

Bulk microphysical schemes

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

\[ L = \frac{\pi p_{\text{w}}}{6} \int_0^\infty D^3 f(D) dD \]

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):

\[ \frac{\partial f(x)}{\partial t} + \nabla \cdot [v f(x)] + \frac{\partial}{\partial z} \left[ v_N(x) f(x) \right] = F(x) \]

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

\[ \frac{\partial N}{\partial t} + \nabla \cdot [v N] + \frac{\partial}{\partial z} \left[ v_N(x) N \right] = N G(x) \]
\[ \frac{\partial L}{\partial t} + \nabla \cdot [v L] + \frac{\partial}{\partial z} \left[ v_L(x) L \right] = L H(x), \quad \bar{x} = L/N \]

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

\[ \frac{\partial L}{\partial t} + \nabla \cdot [v L] + \frac{\partial}{\partial z} \left[ \bar{v}_L(L) L \right] = S(L) \]

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

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.

Assuming a Gamma distribution for cloud droplets the following autoconversion can be derived from the spectral collection equation

\[ f_c(x) = Ax^\nu e^{-Bx} \]

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

\[ \Phi_{\text{lin}}(\tau) = 600\tau^{0.68}(1 - \tau^{0.68})^3 \]

The colored lines represent solutions of the spectral collection equation for various initial conditions. The dashed line is the fit:

\[ \Phi_{\text{lin}}(\tau) = 600\tau^{0.68}(1 - \tau^{0.68})^3 \]

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

A two-moment warm phase scheme

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

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

with \( f(D) \) number density size distribution (unit m\(^{-1}\)).

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

\[
L = \frac{\pi D_w}{6} \int_0^\infty D^3 f(D) dD
\]

and find

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

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

\[
M_n = \int_0^\infty D^n f(D) dD = \frac{\Gamma(n + 1)}{\lambda^{n+1}}
\]

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

\[
L = \frac{\pi D_w}{6} \int_0^\infty D^3 f(D) dD = \pi \rho_w \lambda^4
\]

... 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:

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

Note: This was just a one-moment scheme!

An interesting result for sedimentation:

**Spectral microphysics:**

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

**One-moment scheme:**

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

**Two-moment scheme:**

\[
\frac{\partial N}{\partial t} + \frac{\partial}{\partial z} [v_L(\bar{z}) N] = 0
\]

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

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!


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

\[ f(D) = N_0 D^\mu \exp(-\lambda D) \]

with \( \mu = \mu(\lambda) \)

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

Zhang et al. (2001) measured \( \mu \) vs. \( \lambda \)

\[ \lambda = \lambda(q_r) \text{ in mm}^{-1} \]

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: \( \mu \) and \( N_0 \) are highly variable and have a strong impact on evaporation and sedimentation

Two-moment schemes do not necessarily solve (all) our problems!

.... but we can help them out.

Seifert (2005), JAM

Adding evaporation to the problem leads to more scatter in the \( \mu-\lambda \)-relation.

Using a \( \mu-D \)-relation instead of \( \mu-\lambda \) allows to distinguish large and small mean diameters

- Low \( \mu \) for \( D \approx 1 \text{ mm} \): “breakup/coalescence regime”
- Large \( \mu \) for \( D >> 1 \text{ mm} \): “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 shape parameter of the raindrop distribution
The size effect of evaporation

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

\[
\frac{\partial N_r}{\partial t}|_{\text{evap}} = \gamma N_r \frac{\partial L_r}{\partial t}|_{\text{evap}}
\]

with

\[
\gamma = \frac{D_0}{D_m} \exp(-0.2 \mu)
\]

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

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

<table>
<thead>
<tr>
<th>N,</th>
<th>Microphysics</th>
<th>L</th>
<th>R</th>
<th>C</th>
<th>k</th>
<th>α</th>
<th>Rm</th>
<th>Nc</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>SB</td>
<td>17.4</td>
<td>17.3</td>
<td>0.14</td>
<td>2338</td>
<td>42.3</td>
<td>50.1</td>
<td>16.9</td>
</tr>
<tr>
<td>70</td>
<td>SB</td>
<td>17.2</td>
<td>17.1</td>
<td>0.14</td>
<td>2359</td>
<td>50.3</td>
<td>50.1</td>
<td>16.9</td>
</tr>
<tr>
<td>105</td>
<td>SB</td>
<td>20.6</td>
<td>6.5</td>
<td>0.17</td>
<td>2497</td>
<td>14.1</td>
<td>50.1</td>
<td>10.4</td>
</tr>
<tr>
<td>140</td>
<td>SB</td>
<td>20.6</td>
<td>6.5</td>
<td>0.17</td>
<td>2497</td>
<td>14.1</td>
<td>50.1</td>
<td>10.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N,</th>
<th>Microphysics</th>
<th>L</th>
<th>R</th>
<th>C</th>
<th>k</th>
<th>α</th>
<th>Rm</th>
<th>Nc</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>SB</td>
<td>17.4</td>
<td>17.3</td>
<td>0.14</td>
<td>2338</td>
<td>42.3</td>
<td>50.1</td>
<td>16.9</td>
</tr>
<tr>
<td>70</td>
<td>SB</td>
<td>17.2</td>
<td>17.1</td>
<td>0.14</td>
<td>2359</td>
<td>50.3</td>
<td>50.1</td>
<td>16.9</td>
</tr>
<tr>
<td>105</td>
<td>SB</td>
<td>20.6</td>
<td>6.5</td>
<td>0.17</td>
<td>2497</td>
<td>14.1</td>
<td>50.1</td>
<td>10.4</td>
</tr>
<tr>
<td>140</td>
<td>SB</td>
<td>20.6</td>
<td>6.5</td>
<td>0.17</td>
<td>2497</td>
<td>14.1</td>
<td>50.1</td>
<td>10.4</td>
</tr>
</tbody>
</table>

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

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.

<table>
<thead>
<tr>
<th>Rainshaft model</th>
<th>Rain rate (L, R100, R013, CB=3000m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rainrate (L7000, RH100, R013, CB=3000m)</td>
</tr>
<tr>
<td></td>
<td>rainrate (L7000, RH100, R013, CB=3000m)</td>
</tr>
</tbody>
</table>

Read the paper!

Seifert and Stevens (2005), Journal of the Meteorological Society of Japan

UCLA-LES RICO simulations

<table>
<thead>
<tr>
<th>Rainshaft model</th>
<th>Rain rate (L, R100, R013, CB=3000m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rainrate (L7000, RH100, R013, CB=3000m)</td>
</tr>
<tr>
<td></td>
<td>rainrate (L7000, RH100, R013, CB=3000m)</td>
</tr>
</tbody>
</table>

Read the paper!

Seifert, Nuissier and Stevens (2010), CJ
### Ice particle fall speeds

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

**small ice crystals**

- Ice nucleation
  - Homogeneous nucleation from vapor: Does not occur in the atmosphere!
  - Homogeneous freezing of cloud droplets: at about -37 C, can occur in strong 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.

**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: Thijjs’ 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.

### 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
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 $z_i$:
  - The maximum gradient in $\theta_i$
  - The maximum variance in $\theta_i$
  - 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 $\Gamma$ the temperature lapse rate. How does $z_i$ compare with $z_{enc}$? What is the difference?
Questions II

- Look at the variances: $u^2$, $w^2$, $t^2$. 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 $\text{tot} \_tw$ and $\text{sf} \_tw$.
- Finally, compare the advective tendency ($\text{adv} \_u$) with the diffusion ($\text{dff} \_u$). 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.

Questions IV

- The conditional sampling for cloud water is done in subroutine `accum lvl2` between lines 604 and 658. Look at those in depth.
- 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.

Questions V

- That should be all: Try and compile. Now it gets time to debug.
BOMEX Shallow Cumulus

Questions I

- 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

- 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.

Questions II

- 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.
- 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 70 cm$^{-3}$ 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