Using the UCLA Large Eddy Simulation code - The Speedy Version

Thijs Heus

Max Planck Institute for Meteorology

March 15, 2013

Max-Planck-Institut
für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst

Cascade of Models

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

Cascade of Models

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

Cascade of Models

Other

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

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

Anelastic continuity

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

Ideal gas law equation of state

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

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.

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.

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

Thijs Heus

Max Planck Institute for Meteorology

March 15, 2013

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

- 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!

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 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!

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

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=TRUE ..`
- `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`)
- Run it: `mpirun -np 2 ./uclales`
- Wait...

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.

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`

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)?

Model Options

Thijs Heus

Max Planck Institute for Meteorology

March 15, 2013

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

- 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.

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 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.

Data files

nudge_in

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)

Hands-On

Thijs Heus

Max Planck Institute for Meteorology

March 15, 2013

Max-Planck-Institut für Meteorologie

Hans-Ertel-Zentrum für Wetterforschung
Deutscher Wetterdienst
DWD

Based on dry CBL I

- 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 616 and 674. Look at those in depth.
- The function `get_avg` creates an average over the 2 horizontal direction out of a 3D array.

Based on dry CBL II

- 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 1007 on as an example to add your variables.
- That should be all: Try and compile. Now it gets time to debug.

ALTERNATIVE: Make a Setup for BOMEX Shallow Cumulus

- Check `m̃300063/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?