Manual for DALES

Stephan de Roode, Erwin de Beus, Pim van Dorp,
Pier Siebesma and Antoon van Hooft

January 16, 2018

1 Introduction

This guide starts by providing all essential steps to get started with the Dutch Atmospheric Large Eddy Simulation (DALES) model version 4.1 on your own computer, from compiling the source code to running your first case. This description has been adapted from an early description of DALES 3.1 by Thijs Heus, Chiel van Heerwaarden and Johan van der Dussen.
DALES is written in Fortran 90. This language is still frequently used in the scientific communities of geophysics and astronomy basically because in the 1950s many early scientific programs were developed in Fortran. For example, numerical weather prediction (NWP) was one of the first intensive computer applications, and many current state-of-the-art NWP models have their roots in these early Fortran versions. DALES can be run using multiple CPUs, and it has the option to write output in the so-called NCDF format. This however requires quite some software packages, which are often conveniently available in Linux environments. To perform numerical simulations you will have to install the VirtualBox[1] and the virtual Linux environment including DALES4.1[2]. Sometimes you want to install a package which requires the root password: in this application it has been set to `dales`. Some convenient Linux commands will be summarized in Appendix A. In case you happen to operate Ubuntu on your computer you can also find a recipe to install Dales on Ubuntu 16.04 in Appendix B.

[1] available for downloading from `https://www.virtualbox.org`. In case you are already using Ubuntu or OpenSuSe you can install VirtualBox from `zypper install virtualbox` or `sudo apt-get install virtualbox`, respectively.

[2] available for downloading from
`http://www.srderoode.nl/Teaching/LES_course/Dales-4.1_4.ova`

2 Running DALES in the VirtualBox

In the virtual Linux environment you will see eight icons at the top left of the screen. Open a terminal by clicking on the terminal icon. Type `ls` to see which files and directories are present. The source code of DALES is stored in `dales/src`. You can go to this directory by typing `cd dales/src`. If you want to go back to your home folder simply type `cd`. If you want to read the contents of a file type `atom filename.f90`. If you open the file `program.f90`[3] you will learn about the skeleton of the DALES model in the sense that it calls many subroutines. Note that every `*f90` routine starts with a brief description. Since DALES is a collaboration between different institutions (Heus et al., 2010) it also states the authors of the subroutines, which can be handy in case one has specific questions about the code. For example, subroutines including the effect of plants on evaporation and atmospheric chemistry have been developed by our colleagues from Wageningen University. The bulk of the routines relevant to clouds have been written (e.g. precipitation, moist thermodynamics) or incorporated (radiative transfer) by team members from the TU Delft and KNMI.

2.1 Make an executable of the code

To make an executable of the model code create a new directory by typing

```
mkdir dales.build
```

Go to this directory and type

`cmake ~/dales`[4].

You will see that the directory contains some directories. In addition the script `Makefile` has been generated. It compiles the Fortran file and makes an executable. Execute this file by typing

`make`[5].

The `dales4` model has been put in the directory `~\dales.build\src`.

2.2 Perform a simulation with DALES

2.2.1 Input files

Some examples of input files have been posted on the internet. As you will often use the same input files for different experiments it is therefore convenient to put them in a separate directory. To this end create the directory `~/Cases/CBL`, go to this subdirectory by typing `cd ~/Cases/CBL` and download the files `prof.inp.cbl_fixed_grad`, `lscale.inp.cbl_fixed_grad` and `namoptions.coarse` by typing, as an example,

`wget www.srderoode.nl/Teaching/LES_course/CBL/namoptions.coarse`[6]

[3] Linux tip: type `atom pr` followed by a click on the 'tab' of your keyboard. You will see that the system automatically finishes the filename.

[4] At some computers, like Macs, the $\sim$ symbol can be obtained from the Shift-§-button

[5] If your computer has multiple CPUs you can type `make -j n` to use a number of n CPUs, with $1 \leq n \leq N$ an integer that should be smaller than the number N of CPUs on your machine. If you type `make -j` the number of processors is automatically set to N. Be careful if you do this on a computer cluster as your task will use all CPUs which may lead to a significant speed reduction of the other running processes. You may loose friends if you do so.

[6] You can obtain the history of your previous commands by using the upwards pointing arrow at your keyboard.

We will perform various experiments of the Convective Boundary Layer (CBL) and we will store the results in the directory `~/Experiments/CBL` (create it if it is not present). We wish to collect the output from our first experiment in the subdirectory `~/Experiments/CBL/H001`. Before we can do a run, we have to copy some obligatory input files to this subdirectory. Go to the subdirectory `H001` (`cd ~/Experiments/CBL/H001`) and copy the following files

```
cp ~/Cases/CBL/prof.inp.cbl_fixed_grad prof.inp.001
cp ~/Cases/CBL/lscale.inp.cbl_fixed_grad lscale.inp.001
cp ~/Cases/CBL/namoptions.coarse namoptions
cp ~/dales.build/src/dales4 .
```

In the last command the dot indicates that the name of the copied file remains the same. Always check if the experiment number `iexpnr` in the second line of `namoptions` is the same as the number of the subdirectory, otherwise change to `iexpnr = 001`, for example with aid of `atom namoptions`. The `namoptions` file tells the model to do a simulation on a horizontal domain size of `xsize=ysize`=1.6 km using `itot=jtot`=32 grid points that lasts only `runtime`= 300 s with a time step of `dtmax`=1 s . Except for some testing purposes such a small number of grid points in the horizontal plane (`itot=jtot`=32) is actually never used in practice. However, here we find it convenient to quickly see whether the model runs smoothly.

2.2.2 Execute a simulation

Start a model run by typing

`mpirun -np 2 dales4 |tee output.001`.

The command tells the computer to use 2 CPUs and to send information about the run both to screen and an output file. The last line presents the total duration of the simulation (W =`TOTAL wall time`). The model experiment can be sped up if you have the opportunity to use more than 2 CPUs.

Exercise 1: Applying LES to the entire Earth?

1a. Use the information given above to calculate the computation time for one grid point per computational time step per CPU. The total number of grid points N can be found in `namoptions` and is equal to
N_p = `itot`$\cdot$ `jtot`$\cdot$ `kmax`
The number of time steps is equal to the ratio N_t = `runtime`/`dtmax`. With aid of the Wall clock time W the computational time per time step can be calculated from $t^* = \frac{W N_{\mathrm{CPU}}}{N_t N_p}$. We have to multiply by N_{CPU} as the wall clock time is related to its reciprocal (as we assume that a number of n CPUs reduce the wall clock time by about a factor n).
1b. Use the radius of the Earth (6400 km) to compute the global surface area. Compute the total time needed to repeat simulation `H001` for the entire Earth and for a period of 24 hours using only one CPU.
1c. Suppose that you can make a network consisting of the same CPUs as you are using, and that the calculation time will be reduced inversely proportional to the number of CPUs. How many CPUs are needed to finish the calculation within 24 hours?

Let us now perform a second experiment that will simulate a period of 10 hours. Get the appropriate `namoptions` file,

```
wget www.srderoode.nl/Teaching/LES_course/CBL/namoptions
```

Create a new directory `H002`, copy the other necessary files to this subdirectory, and execute dales4. At this point it will be convenient to open a text document in which you briefly describe some key details about the different runs, for example

`atom ~/Experiments/CBL/cbl_run_info.txt`.

We are now in a position to run the model and obtain model output. However, we have not yet explained what is actually included in the input files. So before digging into the details of the model output we will first explain the structure of the input files.

2.2.3 Input file structure

```
Dry Convective Boundary Layer LES Course Class 1
 height(m) thl(K) qt(kg/kg) u(m/s) v(m/s) tke(m2/s2)
 12.5000 293.075 0.00000 1.00000 0.00000 0.000800000
 37.5000 293.225 0.00000 1.00000 0.00000 0.000266667
 62.5000 293.375 0.00000 1.00000 0.00000 0.000160000
 ......
```

Table 1: Example of a `prof.inp` file.

Table 1 shows the first few lines of `prof.inp`[7]. The LES model expects two header lines. If you have only one, or more than two, the model will crash immediately. The first header line contains a brief description of the case, and the second one the meaning of the columns. The number of data lines should correspond to the number `kmax` in `namoptions`.
The example applies a constant vertical grid resolution $\Delta z = 25$ m. Note that the first level is located at $z_\mathrm{f}(1) = \Delta z/2 = 12.5$ m. This has to do with the fact that DALES uses a staggered grid, in which all variables are located at the so-called full levels (z_f) except for the vertical velocity which is defined at the half levels, $z_\mathrm{h}(1) = 0, z_\mathrm{h}(2) = \Delta z, z_\mathrm{h}(3) = 2\Delta z$, etc. Note that a non-equidistant vertical grid structure is allowed. The key point here is that the heights of the full vertical levels z_f are defined in `prof.inp`.
The values of the liquid water potential temperature (`thl`, θ_l), the total water specific humidity (`qt`, q_t), and the horizontal wind velocity components (`u`, u) and (`v`, v) represent the initial vertical profiles at $t = 0$. They need to be obtained from observations. The last quantity `tke` indicates the initial subgrid Turbulent Kinetic Energy (TKE). To promote the development of turbulence at the resolved scales it usually has a nonzero value in the lower part of the atmosphere.
Table 2 shows the first few lines of `lscale.inp`. Its values indicate large-scale forcing quantities that are assumed to be constant in time. If the forcings are time dependent another input file needs to be built which will be discussed at a later stage. The

[7] The source file that produced the *.inp files can be obtained from `www.srderoode.nl/Teaching/LES_course/CBL/make_input.pro`. You can run it after installing `GDL` by `apt install gnudatalanguage`, then type `gdl` and write the *inp files with aid of the command `.r make_inp`

```
Dry Convective Boundary Layer LES Course Class 1
 height(m) ugeo(m/s) vgeo(m/s) wfls(m/s) not_used not_used dqtdtls(kg/kg/s) dthldt(K/s)
 12.500 1.000 0.000 0.000000 0.0 0.0 0.0 0.0
 37.500 1.000 0.000 0.000000 0.0 0.0 0.0 0.0
 62.500 1.000 0.000 0.000000 0.0 0.0 0.0 0.0
 ......
```

Table 2: Example of a `lscale.inp` file.

geostrophic wind components are indicated by `ugeo` and `vgeo`[8], and the large-scale subsidence is given by `wfls`. There are two columns with the weird names `not_used` that should be present but are actually not used anymore in Dales. Because we wish to keep input files backwards compatible (meaning that files can be used in any arbitrary Dales version) such ghost data need to be included.
Table 3 shows the `namoptions` file used for the second experiment.

2.3 File transfer

It is possible to move files from the Linux environment to your laptop and vice versa. This allows to process LES output fields or to prepare LES input files in your own familiar laptop environment. First you have to go to Machine → Settings (see Fig. 1). Then go to Shared Folders and select an arbitrary directory on your laptop. Take care that you switch on Auto-mount and Make Permanent. You can now access your laptop folder via `cd /media`. Copying a file can be done by

```
cp file /media/your_laptop_folder/file
```

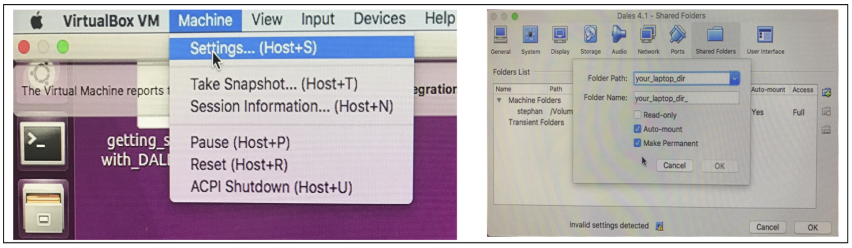

Figure 1: Connecting the virtual Linux environment to your laptop for file sharing.

[8]Recall that the geostrophic wind is the wind velocity that would occur if the horizontal pressure force balances the Coriolis force, which implies that it ignores variations in time and turbulence. In fact, if we talk about prescribing a geostrophic wind, this is actually similar to stating that the define the horizontal pressure gradient since $u_{\text{geo}} \equiv -\frac{1}{\rho f}\frac{\partial p}{\partial y}$ and $v_{\text{geo}} \equiv \frac{1}{\rho f}\frac{\partial p}{\partial x}$.

&RUN	
iexpnr = 002	experiment number
runtime = 36000	total simulation time (secs)
dtmax = 10.	numerical time step (secs), applied if ladaptive=.false.
ladaptive = .true.	max time step is determined by the CFL criterion
irandom = 43	number used in randomizer function
randthl = 0.1	amplitude of initial θ_l perturbations
randqt = 0e-5	amplitude of initial q_t perturbations
/	
&DOMAIN	
itot = 64	nr of points in the horizontal $x-$direction
jtot = 64	nr of points in the horizontal $y-$direction
kmax = 128	nr of points in the vertical $z-$direction
xsize = 3200.	domain size in the $x-$direction (m)
ysize = 3200.	domain size in the $y-$direction (m)
xlat = 32.	latitude, needed for Coriolis force
xlon = 129.	longitude, only relevant for solar radiation
/	
&PHYSICS	
ps = 101900.00	surface pressure (Pa)
thls = 293.	surface value of θ_l (K), not needed if fluxes are prescribed
lcoriol = .true.	include the Coriolis force
iradiation = 0	0=no radiation
z0 = 0.1	roughness length (m)
/	
&NAMSURFACE	
wtsurf = 0.1	surface value of $\overline{w'\theta_l'}$ (mK/s)
wqsurf = 0e-5	surface value of $\overline{w'q_t'}$ ((kg/kg)(m/s))
isurf = 4	4: use z_0 and prescribed surface values for $\overline{w'\theta_l'}$, $\overline{w'q_t'}$
/	
&DYNAMICS	
cu = 1.	Galilean transformation velocity in the $x-$direction (trick to reduce time step)
cv = 0.	Galilean transformation velocity in the $y-$direction
iadv_mom = 2	2:2nd order (Central Difference) advection scheme for momentum
iadv_tke = 2	
iadv_thl = 2	
iadv_qt = 2	
/	
&NAMSUBGRID	
ldelta = .true.	length scale in TKE scheme $\ell = \Delta = (\Delta x \Delta y \Delta z)^{1/3}$
/	
&NAMCHECKSIM	
tcheck = 5	computes and writes the max value of the divergence of the velocity field every 5 s
/	
&NAMTIMESTAT	
ltimestat = .true.	compute statistics of scalars (tmser and tmsurf)
dtav = 60	write this output to file every 60 s
/	
&NAMGENSTAT	
lstat = .true.	computes horizontal slab mean statistics
dtav = 60	compute slab average every 60 s
timeav = 600	write average during a simulation period of 600 s
/	
&NAMNETCDFSTATS	
lnetcdf = .true.	provide output in NCDF format
/	

Table 3: Example of a minimal namoptions file. Many more switches can be added, or the value of switches can be changed, which will be discussed in the remainder of the course.

2.4 The Convective Boundary Layer

The Convective Boundary Layer (CBL) is one of the first boundary-layer regimes that was investigated with LES (Nieuwstadt et al., 1993)[9]. To study how its time evolution depends on the surface forcing and atmospheric stability we will now perform some sensitivity experiments. For each experiment use a different experiment number.

Exercise 2: Evolution of the mean state.

Use the results of the second experiment to make a plot of the temperature T and θ_l at different times. These variables are written to the ASCII file `field.iexpnr`[a]. Discuss their vertical gradients. Can you see the development of a thermal inversion layer? Use the file `tmser1` to make a plot of the quantity `zi` as a function of time. Do you find a relation between this quantity and the structure of θ_l? We will call this simulation the Reference case.

Exercise 3: Sensitivity experiments.

3a. We will now use DALES to study the effect of the slope of θ_l and the surface heat flux on the evolution of the vertical temperature profile. To this end perform a new experiment in which you change the value of the surface heat flux in `namoptions`. For example, you may increase or reduce it by a factor of 2. In addition, download the file `prof.inp.steep_thl_grad` from `www.srderoode.nl/Teaching/LES_course/CBL/` to investigate the effect of the atmospheric stability. Compare your results with the reference case.
3b. Download the file `prof.inp.steep_thl_grad` or create a new `prof.inp` file yourself with the `make_input.pro`. As you will perform many more different case in this course you may also start writing a script that creates the input files with use or your favorite software package. Perform a run and compare your results with the reference case.
3c. Investigate to which extent the results change if you modify, for example, the horizontal grid size or the number of horizontal grid points.
3d. Investigate the effect of changing the vertical resolution. In the latter case note that you will have to modify the `*.inp` files and adapt the value of `kmax` in `namoptions`.

[a] It is also possible to read the same data set saved in NCDF format with aid of the matlab routine `netcdfplotexample.m` available from `www.srderoode.nl/Teaching/LES_course/Software`

[9] As this paper is difficult to find we have posted it at `www.srderoode.nl/Teaching/LES_course/Literature/Nieuwstadt_etal_LES_CBL.pdf`.

3 Formulation of the LES model

Here we will briefly summarize the governing LES equations that apply to an atmosphere free of clouds. For a detailed description of an LES model like DALES we refer the reader to Heus et al. (2010) and Böing (2014) who explain the updated anelastic version for simulations of deep convection.

3.1 Prognostic budget equations

LES models solve the budget equations for filtered variables including momentum and thermodynamic state variables, such as heat, entropy, or the total water specific humidity. After application of the LES filter the prognostic equation for an arbitrary scalar φ can be written as

$$\frac{\partial \tilde{\varphi}}{\partial t} + \tilde{u}_j \frac{\partial \tilde{\varphi}}{\partial x_j} = -\frac{\partial \widetilde{u_j'' \varphi''}}{\partial x_j} + \tilde{S}_\varphi, \tag{1}$$

with t the time, and the velocity vector components $(u_1, u_2, u_3) = (u, v, w)$ in the (x, y, z) direction, respectively. In the absence of clouds $\varphi \in \{\theta, q\}$, with θ representing the potential temperature and q the water vapor specific humidity. A tilde indicates the filtered mean value and the SFS scalar flux is denoted by $\widetilde{u_j'' \varphi''} \equiv \widetilde{u_j \varphi} - \tilde{u}_j \tilde{\varphi}$. The Boussinesq form of the filtered momentum equation reads,

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u_j} \frac{\partial \tilde{u}_i}{\partial x_j} = \frac{g}{\theta_0} \delta_{i3} (\tilde{\theta_\mathrm{v}} - \overline{\theta_\mathrm{v}}) - \frac{\partial \pi}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \tilde{S}_{u_i}, \tag{2}$$

where g is the gravitational acceleration, θ_0 the reference state virtual potential temperature, δ_{ij} the Kronecker-delta function, the virtual potential temperature is defined as,

$$\theta_\mathrm{v} \equiv \theta(1 + \epsilon_I q), \tag{3}$$

with $\epsilon_I \approx 0.608$ a thermodynamic constant, π is the modified pressure (Deardorff, 1973),

$$\pi = \frac{p}{\rho_0} + \frac{2}{3}\tilde{e}, \tag{4}$$

and an overbar is used to indicate a horizontal slab-mean value. For compact notation we have included the mean horizontal pressure gradient and the Coriolis force in the source function S_{u_i}. The deviatoric part of the SFS momentum flux τ_{ij} is computed from (Deardorff, 1980),

$$\tau_{ij} \equiv \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j - \frac{2}{3}\delta_{ij}\tilde{e} = -K_\mathrm{m} \left(\frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j} \right), \tag{5}$$

and

$$\widetilde{u_j'' \varphi''} = -K_\mathrm{h} \frac{\partial \tilde{\varphi}}{\partial x_j}. \tag{6}$$

The factor $\frac{2}{3}\delta_{ij}\tilde{e}$ that is subtracted in (5) does not arise from the filtering procedure. To compensate it has been added to the filtered pressure term to give the modified pressure. Here K_m and K_h represent the eddy viscosity for momentum and the eddy diffusivity for the thermodynamic scalars, respectively. In a TKE closure approach both are taken proportionally to the square root of the SFS TKE (e),

$$K_\mathrm{m} = c_\mathrm{m} \lambda \tilde{e}^{1/2}, \tag{7a}$$

$$K_\mathrm{h} = c_\mathrm{h} \lambda \tilde{e}^{1/2}, \tag{7b}$$

with λ the characteristic length scale of the SFS turbulent eddies and c_m and c_h proportionality constants. By analogy with the molecular Prandtl number, which is defined as the ratio of the viscosity to the thermal diffusivity, the ratio $K_\mathrm{m}/K_\mathrm{h}$ can be interpreted as a turbulent SFS Prandtl number,

$$\mathrm{Pr_T} = \frac{K_\mathrm{m}}{K_\mathrm{h}} = \frac{c_\mathrm{m}}{c_\mathrm{h}}. \tag{8}$$

The budget equation for $\tilde{e}$ reads,

$$\frac{\partial \tilde{e}}{\partial t} + \tilde{u}_j \frac{\partial \tilde{e}}{\partial x_j} = \frac{g}{\theta_0}\widetilde{w''\theta''_\mathrm{v}} - \widetilde{u''_i u''_j}\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \widetilde{u''_j e}}{\partial x_j} - \frac{1}{\rho_0}\frac{\partial \widetilde{u''_j p''}}{\partial x_j} - \epsilon, \tag{9}$$

with ρ_0 a reference density and p the pressure. The SFS flux $\widetilde{u''_i u''_j}$ is computed as τ_{ij} following (5), and (6) is used to calculate the SFS fluxes of the θ and q which in turn are used to calculate the SFS buoyancy flux. The total turbulent transport term is computed following a downgradient diffusion approach,

$$\widetilde{u''_j e} + \frac{\widetilde{u''_j p''}}{\rho_0} = -2K_\mathrm{m}\frac{\partial \tilde{e}}{\partial x_j}, \tag{10}$$

and the viscous dissipation of e by molecular viscosity (ϵ) is calculated as

$$\epsilon = c_\epsilon \frac{\tilde{e}^{3/2}}{\lambda}, \tag{11}$$

with c_ϵ a proportionality constant.
In the remainder of the text we will omit the tildes. With this notation the parameterized equation for the SFS TKE can be written as

$$\underbrace{\frac{\partial e}{\partial t}}_{\text{tendency}} + \underbrace{u_j \frac{\partial e}{\partial x_j}}_{\text{mean advection}} = -\underbrace{K_\mathrm{h} N^2}_{\text{buoyancy}} + \underbrace{K_\mathrm{m} S^2}_{\text{shear production}} + \underbrace{2\frac{\partial}{\partial x_j}\left(K_\mathrm{m}\frac{\partial e}{\partial x_j}\right)}_{\text{turbulent transport}} - \underbrace{\frac{c_\epsilon e^{3/2}}{\lambda}}_{\text{dissipation}}, \tag{12}$$

with

$$S^2 \equiv \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)^2 = \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j}. \tag{13}$$

The classical Smagorinsky model assumes a balance between shear production and dissipation of TKE (Smagorinsky, 1963). Stratification effects can be included by maintaining the buoyancy flux (Mason, 1989),

$$-K_\mathrm{h} N^2 + K_\mathrm{m} S^2 - \frac{c_\epsilon e^{3/2}}{\lambda} = 0. \tag{14}$$

This simplified form of the SFS TKE equation thus neglects the tendency, mean advection and turbulent transport.

3.2 Formulations of the length scale

3.2.1 Constant length scale l_Δ

Deardorff (1973) proposed to use the geometric mean l_Δ of the filter mesh sizes $\Delta x, \Delta y$ and Δz as a representative length scale for SFS eddies,

$$l_\Delta \equiv (\Delta x \Delta y \Delta z)^{1/3}. \tag{15}$$

3.2.2 Stability dependent length scale l_Δ

Deardorff (1980) argued that for a stable stratification the length scale of the eddies may become smaller than the grid size. The vertical stability can be expressed in terms of the Brunt-Väisälä frequency N,

$$N \equiv \left(\frac{g}{\theta_0} \frac{\partial \theta_\mathrm{v}}{\partial z} \right)^{1/2}. \tag{16}$$

The gradient Richardson number $\mathrm{Ri_g}$ is defined by

$$\mathrm{Ri_g} \equiv \frac{N^2}{S^2}. \tag{17}$$

Deardorff proposed the following stability dependent length scale,

$$l_\mathrm{D} = c_\mathrm{n} \frac{e^{1/2}}{N}, \tag{18}$$

to be used only if its magnitude is smaller than l_Δ,

$$\lambda = \min\left(l_\Delta, l_\mathrm{D}\right). \tag{19}$$

For $\lambda = l_\mathrm{D}$ the quantity c_h becomes dependent on the stability,

$$c_\mathrm{h} = \left(c_{\mathrm{h},1} + c_{\mathrm{h},2} \frac{\lambda}{l_\Delta} \right) c_\mathrm{m}, \tag{20}$$

with $c_{\mathrm{h},1} = 1$ and $c_{\mathrm{h},2} = 2$. This approach effectively lets the turbulent Prandtl number depend on the stability, with $\mathrm{Pr_T}$ approaching unity for a very strong stable stratification. The factor c_ϵ is also adapted according to

$$c_\epsilon = c_{\epsilon,1} + c_{\epsilon,2} \frac{\lambda}{l_\Delta}. \tag{21}$$

3.2.3 Mason and Thomson length scale l_M

Last we mention the length scale l_M that was constructed by Mason and Thomson (1992) to let the resulting eddy viscosity better match observed MO similarity relations. Specifically, they proposed

$$\frac{1}{l_\mathrm{M}^n} = \frac{1}{[\kappa(z+z_0)]^n} + \frac{1}{(c_\mathrm{s} l_\Delta)^n}, \tag{22}$$

with z_0 the roughness length. Brown et al. (1994) suggested to use $n = 2$.

3.3 Model constants

In the remainder we will use subscripts 'Δ', 'D' and 'M' to indicate quantities that are derived with the constant length scale l_Δ defined by (15), the stability dependent length scale l_D according to (18) and (19), and the length scale l_M following (22), respectively.
DALES has evolved from the LES code used by Nieuwstadt et al. (1993), and the original setting $c_\mathrm{h} = 3c_\mathrm{m}$ is still used, i.e. $\mathrm{Pr}_\Delta = 1/3$. The factor c_s represents the Smagorinsky constant,

$$c_\mathrm{s} \equiv \left(\frac{c_\mathrm{m}^3}{c_\epsilon} \right)^{1/4} = \frac{c_\mathrm{f}}{2\pi} (\frac{3}{2}\alpha_\mathrm{m})^{-3/4}, \tag{23}$$

with c_f the filter constant and α_m the Kolmogorov constant (see Table 4 for their values used in DALES). We note that there is no general consensus on the optimum values of these quantities, causing differences in the value for the Smagorinsky constant. For example, Lesieur et al. (2005) uses $\alpha_\mathrm{m} = 1.4$, whereas Schumann (1975) and Meneveau and Katz (2000) use a value of 1.5 and 1.6, respectively. As compared to DALES Mason (1989) uses a smaller filter constant of $c_\mathrm{f} = 2$. Kleissl et al. (2003) used an array of sonic anemometers to measure SFS diffusion constants in the atmospheric surface layer. They actually found that c_s is not constant, but is reduced near the ground surface and also tends to become smaller with increasing stability.

c_f	α_m	$c_{\mathrm{h},1}$	$c_{\mathrm{h},1}$	$c_{\epsilon,1}$	$c_{\epsilon,2}$	c_n
2.5	1.5	1	2	0.19	0.51	0.76

Table 4: Summary of constants used in the SFS TKE model. The values are corresponding to the ones used in DALES.

	c_m	c_s	c_h	c_ϵ
l_Δ	0.12	0.22	0.35	0.7

Table 5: Summary of dependent quantities for the SFS TKE model.

4 Instruction notes Class 2

4.1 Introduction

In practice one sometimes wishes to take a detailed look at the 3D LES fields, or to compute some additional statistics, for example during during a particular period of time of a previous simulation. The `initd*` files that are written by Dales allow to start a new simulation on the basis of all relevant variables that have been saved during a previous one. Here we will explain how the 'warm' start can be executed, how one can produce 3D fields, and how one can perform additional statistics for an arbitrary period of time on the basis of fields that were generated in a previous run.

4.2 How to perform a warm start

Here we will assume that you have performed a simulation with experiment number `iexpnr` and which has generated files `initdhhhmmmxxprocnryyprocnr.iexpnr`, for example, `initd08h00mx000y000.001`. Here `hh` is the hour at which the data are written, `mm` is the minute, `xprocnr` and `yprocnr` are the processor numbers that contain a part of the horizontal domain area, and `iexpnr` is the number of the experiment. In the standard settings of the model `init*` files are produced after each hour, but smaller time steps are also possible. The `init*` are binary (non-ascii) files, and the data are saved in exactly the same way as they were present in the computer memory. This allows to redoing a part of the simulation identically to the original one.
Let us perform a new simulation. In the following example we will assume a new CBL experiment with number `iexpnr=002` and which uses input files from hour `hh=08` of a previous simulation with number `iexpnr_old=001`. To this end
• Make the directory `~/Experiments/CBL/H002` and copy the dales executable into this directory
`mkdir ~/Experiments/CBL/H002`
`cp ~/dales.build/src/dales4 ~/Experiments/CBL/H002/.`
• Go to the directory `~/Experiments/CBL`
`cd ~/Experiments/CBL` • Because we will perform a warm start, the settings of the new run need to be the same, except for some quantities like the run time. Copy the namoptions file,
`cp H001/namoptions.001 H002/namoptions.002`
• copy the `lscale`, `prof` and `baseprof` input files from H001 to H002 with the appropriate extensions, i.e. `cp H001/prof.inp.001 H002/prof.inp.002`. The model is coded such that it needs to read the vertical levels from `prof.inp`, but for a warm start it will obviously not use the initial vertical proflies of the (thermo-) dynamic variables.
• for each processor field, copy the file of the eight hour,
`cp H001/initd08* H002/.`
The '`*`' is a wildcard and acts to copy all files which names start with `initd08`. This is convenient as it will copy all the input files of hour 8 that were produced by each processor, and will be needed for the new run.

We will need to adapt `namoptions.iexpnr` to let the model know to use the `init*` files. This is done by adding `lwarmstart` and `startfile` in the data block that has the header `&RUN`. The end of the data block iis indicated by '`/`' symbol:

```
&RUN
iexpnr = 002
```

```
.....
lwarmstart = .true.
startfile = 'initd08h00mx000y000.001'
/
```

The model is coded such that each processor will read its own `init*` files of hour `08`. Before adding new information to `namoptions` run this experiment to see if it works.

4.3 Conditional sampling

To apply a conditional sampling of updraft fields you will need to add the following data block to `namoptions`,

```
 &NAMSAMPLING
dtav = 60
timeav = 600.
lsampup = .true.
/
```

This asks the model to provide time-averaged statistics at each 600 s on the basis of analyses of the 3D fields at time steps of 60 s. Perform a warm start of the CBL case and run the case for one hour (`runtime = 3600`).

4.4 Creating 3D fields (Optional exercise)

Now we wish to obtain a snapshot of the atmospheric structure for H001. To this end we will keep working in H002. Change `runtime = 10` and add the following block of data to namoptions:

```
&NAMFIELDDUMP
lfielddump = .true.
klow = 10
khigh = 10
dtav = 10
/
```

This setting will enforce the model to write the fields at the tenth vertical level. In principle $1 \leq$ `klow` $\leq$ `khigh` $\leq$ `kmax`.
If you run the model with these settings you will find that as many `fielddump*` files as processor numbers have been generated. You can find information about the content of file from[10]

- `ncdump fielddump.000.000.002.nc -h`, gives a summary of the variables that are present,
- `ncdump fielddump.000.000.002.nc -v thl`, gives the values of a variable, in this example θ_l

Another handy tool is `ncview`[11] . If you type `ncview fielddump.000.000.002.nc` you can quickly plot a 2D field. However, one can find in the Dales source file `~/dales/src/modfielddump.f90` that the wind velocities have to be divided by a

[10] first install the `ncdump` package from the command `sudo apt install netcdf-bin`
[11] first install the `ncview` package from the command `sudo apt install ncview`

factor 1000 to get them in units m/s. Also, θ_l first has to be multiplied by a factor 100, and consequently a constant factor of 300 has to be added. This data manipulation allows to write data as compact integers, which keeps the `fielddump` files as small as possible.

5 Instruction notes Class 4

5.1 Introduction

So far we have performed experiments with a constant surface flux forcing. Dales also allows to apply time-dependent forcings, which needs to be prescribed in the file `ls_flux.inp`. We will look at some new values of switches in `namoptions`.

5.2 The GABLS1 stable boundary layer model intercomparison case.

The GABLS1 (**G**lobal Energy and Water cycle Exchanges **A**tmospheric **B**oundary **L**ayer **S**tudy) was set up to test the skill of a wide variety of atmospheric models to represent the turbulence structure of the nocturnal boundary layer (Beare et al., 2006). Create a new directory ∼`/Cases/Gabls1` and download the input files to this directory,

```
wget www.srderoode.nl/Teaching/LES_course/Gabls1.tar
```

A `.tar` files collects multiple files in a single file structure. This is handy if you want to copy files to another platform. Useful comments are

`tar -tf Gabls1.tar`	shows the list of the included files
`tar -xvf Gabls1.tar`	unpacks the files
`tar -cf all_files.tar file1 file2`	collects `file1` and `file2` in `allfiles.tar`
`gzip all_files.tar`	compress `all_files.tar`
`gunzip all_files.tar`	uncompress `all_files.tar`

Note that in general information about Linux commands can be readily obtained from the `man` command, e.g. `man tar`

Because we will perform a new series of runs of Gabls1, it is recommended to create a new directory ∼`/Experiments/Gabls1`. For your first test run create a new subdirectory ∼`/Experiments/Gabls1/H000` and prepare this experiment by

```
cp ∼/Cases/Gabls1/prof.inp.gabls1 ∼/Experiments/Gabls1/H000/prof.inp.000
cp ∼/Cases/Gabls1/lscale.inp.gabls1 ∼/Experiments/Gabls1/H000/lscale.inp.000
cp ∼/Cases/Gabls1/ls_flux.inp.gabls1 ∼/Experiments/Gabls1/H000/ls_flux.inp.000
cp ∼/Cases/Gabls1/namoptions.000 ∼/Experiments/Gabls1/H000/namoptions
```

A summary of some key switches of `namoptions` is presented in Table 6.

5.2.1 Adaptive time step

An important difference with the CBL case is that the horizontal grid resolution is much higher for Gabls1. This choice is motivated by the fact that the turbulent eddies for a turbulent stratification are much smaller Run this case to see whether it runs smoothly. Do a second run in which you change enlarge the domain size by change `xsize` and `ysize` to 800 m. Because of the `namoptions` switch `ladaptive=.true` a minimum value of the numerical time step is evaluated at every time step. Check this by comparing the wall clock times for the two experiments. The reason why the coarse horizontal resolution run is much faster can be explained from the Courant-

& RUN		
iexpnr	= 000	
runtime	= 300	short test run
dtmax	= 10	
ladaptive	= .true.	
/		
&DOMAIN		
itot	= 32	
jtot	= 32	
xsize	= 100.	this value gives a horizontal grid resolution $\Delta x = 3.125$ m
ysize	= 100.	
xlat	= 73.	high latitude
/		
&PHYSICS		
ltimedep	= .true.	to indicate time-dependent large-scale forcing
/		
&NAMSURFACE		
isurf	= 2	to indicate that we prescribe the surface temperature as a lower boundary condition
/		
&NAMBUDGET		
lbudget	= .true.	to compute the budget of the Turbulent Kinetic Energy
dtav	= 60.	computes the budget at intervals of 60 s
timeav	= 600.	computes and saves time averaged values over a period of 600 s
/		

Table 6: Key `namoptions` switches for the Gabls1 case.

Friedrichs-Lewy (CFL) stability criterion which for 1D reads,

$$\mathrm{CFL} = u\frac{\Delta t}{\Delta x} < \mathrm{CFL}_{\mathrm{crit}}. \tag{24}$$

For $\mathrm{CFL}_{\mathrm{crit}} = 1$ This criterion can be interpreted as one that states that properties should not travel to an adjacent grid size within one time step Δt. A second criterion checks the turbulent viscosity K_{m} to let the time step satisfy the following cell Peclet number (Pe) condition,

$$\mathrm{Pe} = K\frac{\Delta t}{\Delta x}^2 < \mathrm{Pe}_{\mathrm{crit}}. \tag{25}$$

Dales computes CFL and Pe as follows, respectively,

$$\mathrm{CFL} = \left(\frac{u^2}{\Delta x^2} + \frac{v^2}{\Delta y^2} + \frac{w^2}{\Delta z^2}\right)\Delta t^2 < 1, \tag{26}$$

$$\mathrm{Pe} = \frac{K_{\mathrm{m}}\Delta t}{\min(\Delta x, \Delta y, \Delta z)^2} < 0.15, \tag{27}$$

Note that in Dales $\mathrm{CFL}_{\mathrm{crit}}$ may be smaller for higher-order advection schemes[12]. In conclusion Dales takes the minimum time step following from the Peclet and CFL criteria, and in general larger time steps are permitted for a coarser grid resolution. It is important to stress that these criteria are necessary but do not guarantee stability.

5.2.2 Input file for time-dependent large-scale forcings

If the switch `ltimedep` is set to `.true.` dales expects the presence of the file `ls_flux.inp` which contains the time-dependent large-scale forcing conditions. In the previous example of the CBL we set `isurf=4` to prescribe the surface heat fluxes

[12] see the source file `tstep.f90`

and the surface roughness length z_0. Any time dependent surface fluxes can be prescribed in the first data block of `ls_flux.inp`. However, in Gabls1 the surface temperature was prescribed, and its use must be enforced by setting `isurf=2`. Except for `isurf=1` the default setting of Dales is that a saturated surface is assumed. The number of times in `ls_flux` can be chosen arbitrarily and Dales will linearly interpolate the values between the given times.
The next part of `ls_flux` contains vertical profiles of the large-scale processes similar to `lscale.inp`. Please notice that the latter file is always required, although for `isurf=2` its values are overwritten by `ls_flux`. This needs to be improved in a future version of Dales.

```
Gabls1.Hres
time       wtsurf        wqsurf        thls        qts         psurf
[s]        [K m/s]       [kg m/s]      [K]         [kg/kg]     [Pa]
0.0        0.0000e+00    0.0000e+00    265.000     0.000       100000.0
3600.0     0.0000e+00    0.0000e+00    264.750     0.000       100000.0
7200.0     0.0000e+00    0.0000e+00    264.500     0.000       100000.0
10800.0    0.0000e+00    0.0000e+00    264.250     0.000       100000.0
14400.0    0.0000e+00    0.0000e+00    264.000     0.000       100000.0
18000.0    0.0000e+00    0.0000e+00    263.750     0.000       100000.0
21600.0    0.0000e+00    0.0000e+00    263.500     0.000       100000.0
25200.0    0.0000e+00    0.0000e+00    263.250     0.000       100000.0
28800.0    0.0000e+00    0.0000e+00    263.000     0.000       100000.0
32400.0    0.0000e+00    0.0000e+00    262.750     0.000       100000.0


large scale forcing terms
height     ug            vg            wfls        not_used    not_used    dqtdt       dthlrad
#          0.00000
1.56250    8.000         0.000         0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.00e+00
4.68750    8.000         0.000         0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.00e+00
7.81250    8.000         0.000         0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.00e+00
...
...

large scale forcing terms
height     ug            vg            wfls        not_used    not_used    dqtdt       dthlrad
#          32400.0
1.56250    8.000         0.000         0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.00e+00
```

Table 7: Snapshot of the Gabls1 `ls_flux.inp` file.

6 Radiative transfer

year	1980	1990	2000	2010
grid points	64^3	128^3	256^3	512^3
simulation time	4 hrs	$4-10$ hrs	$4-10$ hrs	$\sim$ days
hor. domain size (km^2)	3×3	6×6	25×25	200×200
computing	single CPU		multiple CPU	(ultrafast) GPU
code development		advection schemes		anelastic approach
new physics	turbulence	moist thermodynamics	'warm' microphysics	ice microphysics
		simple radiation		detailed radiation
case	CBL	Cumulus	Precipitating Clouds	Cloud Transitions
		Stratocumulus	Stable BL	Continuous runs
				Deep Convection
key questions	turbulence	entrainment	precipitation	mesoscale organization
				weather prediction
				climate research
				wind and solar energy

Table 8: Summary

Table 8 shows a schematic overview of the development of DALES. So far we have been concerned with LES of turbulence in clear air (CBL, SBL), and in (precipitating) shallow cumulus. These runs have in common that they could all be simulated without interactive radiation. The first simulations of stratocumulus requested the use of a radiation scheme to capture the strong longwave radiative cooling at its top. Duynkerke et al. (1999) proposed a parameterization that only contained a few lines of model code. A few years later Stevens et al. (2005) added some new code to capture the effect of longwave warming near the cloud base, in addition to a longwave radiative cooling of the free troposphere. An analytical solution to the solar radiative transfer problem in clouds was used for simulations of the full diurnal cycle of stratocumulus (Duynkerke et al., 2004). These parameterizations are all present in Dales.

Although these parameterizations are very simple they are rather accurate (see for example Larson et al. (2007)). However, for operational use they have a couple of major disadvantages. First of all, they require a specification of the downwelling flux at the top of the LES domain. This may be obtained by running a full radiative transfer model. Secondly, the schemes are insensitive to the presence of other quantities than liquid water. This is a problem since, for example, the downwelling longwave radiation strongly depends on the temperature and water vapor specific humidity. We will therefore not discuss these parameterizations, but the interested reader can find their descriptions in the literature mentioned above.

As part of the CGILS experiment the Rapid Radiative Transfer Model for General Circulation Models (RRTMG) was implemented in Dales (Blossey et al., 2013). The longwave and shortwave bands are divided in smaller subintervals, and since the scheme is applied to the full atmopheric column it is a computationally expensive code. The radiation code may be called at arbitrary time intervals. Often a radiation time step of a minute is used, and the computed radiative tendencies will be be

assumed to be constant in time until the next radiative flux computation. To reduce the computational time up to about ∼20 years ago global weather forecast models called the radiation scheme only at rather large time intervals of a few hours.

To make a step towards more realism we will learn how we can include detailed shortwave and longwave radiative transfer calculations with Dales and study its results.

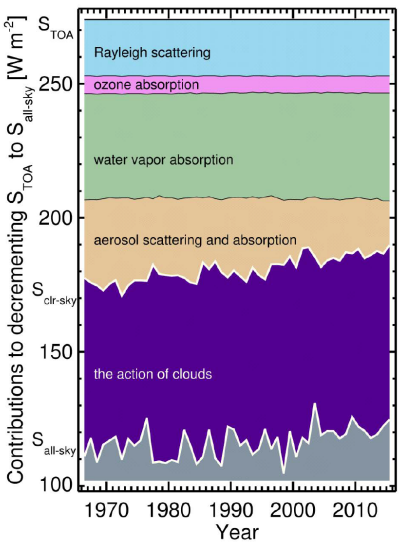

Figure 2: An assessment of the annual variations in the variations of the solar radiative fluxes due to various species (Boers et al., 2017).

6.1 Solar radiation at Cabauw

Fig. 2 shows the contributions to the decrease of the downwelling solar radiative flux by the presence of various species. Interestingly, the results suggest that during the last 40 years the local atmosphere has become cleaner as manifested by a gradual reduction of the absorption of solar radiation by aerosols.

In any case, solar radiation is changing throughout the atmospheric column, with ozone absorption of UV radiation in the stratosphere being a striking example. This implies that in order to make accurate radiation calculations on the one hand the vertical domain should reach the top of the atmosphere. On the other hand, turbulence might only be present in the lower few kilometers, which asks for a high density of grid points in this layer only. In LES models these opposing demands have been nicely compromised by adding a horizontally homogeneous layer on top of the LES domain. This upper layer is used only for radiative transfer calculations and contains vertical profiles of pressure, temperature, humidity and ozone. Although RRTMG contains standard ozone profiles, the user may optionally prescribe a different ozone concentration.

An example of a so-called background profile is shown in Table 9. There are a few notable issues:

```
netcdf backrad.ctl_s11_Nz_128 {
dimensions:
 lev = 64 ;
variables:
 float lev(lev) ;
 lev:units = "Pa" ;
 lev:long_name = "isobaric level" ;
 float T(lev) ;
 T:units = "K" ;
 T:long_name = "Temperature" ;
 float q(lev) ;
 q:units = "kg/kg" ;
 q:long_name = "water vapor content" ;
 float o3(lev) ;
 o3:units = "kg/kg" ;
 o3:long_name = "Ozone mass mixing ratio" ;

// global attributes:
 :Title = "CGILS radiation input" ;
data:

 lev = 101967.2, 101674.7, 101269.1, 100736, 100076.6, 99302.08, 98397.48,
 97336.52, 96116.12, 94736.95, 93468.34, 93468.15, 93198.96, 91502.81,
 89652.63, 87655.99, 85520.44, 83252.66, 80861.59, 78359.75, 75759.24,
 73070.53, 70305.65, 67479.39, 64606.13, 61697.88, 58767.62, 55830.45,
 52908.93, 50031.8, 47227.08, 44520.95, 41937.78, 39487.86, 37167.38,
 34971.04, 32892.95, 30927.47, 29069.2, 27312.95, 25653.76, 24086.83,
 22607.61, 21211.71, 19894.93, 18653.28, 17482.86, 16380.03, 15341.24,
 14342.19, 13351.92, 12362.02, 11372.46, 10383.19, 9394.112, 8405.158,
 7416.284, 6427.467, 5438.624, 4449.765, 3460.937, 2472.099, 1483.253,
 494.4198 ;

 T = 291.1213, 290.8827, 290.551, 290.1137, 289.5704, 288.929, 288.1753,
...
 q = 0.01021351, 0.01021351, 0.01021351, 0.01021351, 0.01021351, 0.01021351,
...
 o3 = 4.673502e-08, 4.682244e-08, 4.695263e-08, 4.712739e-08, 4.734801e-08,
...
}
```

Table 9: Example of an `ncdump` of `backrad.nc`

- Only pressure levels are needed and not height.
- The file requests the temperature.
- Ozone is used only if `usero3 = .true.` in `namoptions`.
- The number of vertical levels can be chosen arbitrarily. More is always better but also more expensive.
- RRTMG will always use the full 3D LES computational domain and data from `backrad.nc` for levels above.
- Each LES subcolumn has an identical thermodynamic profile above its top.
- In the current Dales setting of RRTMG the default aerosol concentration is zero, but it can be set in the variable `ztaua` in the subroutine `rrtmg_sw_rad.f90`.

In conclusion, if one is interested in the effect of clouds on radiation, one must choose the LES domain high enough to capture them all.

6.2 Instructions: How to run Dales as a radiative transfer model

Figure 3: Example of deselected code in `program.f90` in order to basically run it as a radiative transfer tool.

In this week's assignment we will learn how to include radiative transfer calculations. To be able to do quick runs download the modified file `program.f90` from Brightspace

(see Content → Les input files → Radiation). In this model version all physics subroutines have been switched off, except for radiative transfer. This is achieved by putting `!scm` at the beginning of a line of code that needs to be skipped. In Fortran all text after '`!`' will be assumed to be a comment and will ignored when compiling the code. The comment `scm` has been included for easy readability and to indicate that this version actually operates in a single-column model (SCM) mode.

6.3 Compiling new code

- Store the original file `program.f90` in a separate directory, i.e. ∼`/Les_versions/Original/program.f90`.
- Put the modified file `program.f90` in the directory ∼`/src`.
- Go to the directory ∼`/dales.build`.
- Type `make clean`. This will remove all previously compiled code.
- Type `make`. A new executable version of Dales will be build.

6.4 Switches in namoptions

In practice it is not possible to run the model with only one column. This can be circumvented by using the minimum number of requested grid points 2×2 in the horizontal directions and switching off the randomization of the thermodynamic fields in `namoptions`. Table 10 shows the switches to be used for an experiment including radiation. In particular for solar radiation one needs to prescribe the initial time, number of the day in a year (1-365), and the longitude and latitude.

Because radiative transfer is expensive, the parameter `timerad` can be used to let the radiative transfer code to be called at larger time steps than used for the dynamics.

6.5 Setting up a radiation experiment (without dynamics)

- Compile Dales with the modified `program.f90` according to the description above.
- Make a directory ∼`/Cases/RRTMG`
- Make and go to the directory containing the new experiment ∼`/Experiments/Radiation/H000`.
- Copy the new Dales executable version to H000.
- Download the files `rrtmg_lw.nc` and `rrtmg_lw.nc` and put them in ∼`/Cases/RRTMG`. They contain data needed for RRTMG. For use of RRTMG you will need to put these input files in the experiment directory. You can copy the files, but because they are rather large it is more convenient to link them. Go to the directory `H000` and type
  `ln -s ∼/Cases/RRTMG/rrtmg_lw.nc .`
  `ln -s ∼/Cases/RRTMG/rrtmg_sw.nc .`
- Copy the `prof.inp` and `lscale.inp` files you made for the Cabauw experiment to `H000`.
- Download `backrad_cabauw_20120329.nc` from Brightspace and copy it to the experiment directory with the name `backrad.inp.000.nc`.

&RUN		
....		
randthl	= 0.0	switch off inertial perturbation
randqt	= 0.0	switch off inertial perturbation
/		
&DOMAIN		
itot	= 2	minimum nr of grid points
jtot	= 2	minimum nr of grid points
xlat	=52.	latitude
xlon	=5.	longitude
xday	=89	julian day
xtime	=0.0	time (UTC)
....		
/		
&PHYSICS		
thls	= 264.9	surface potential temperature
rad_ls	= .false.	do not use radiative tendencies from lscale
iradiation	= 4	use RRTMG
timerad	= 1.0	time step for radiation calculations
sw0	= 1361.	value of the solar constant
/		
&NAMSURFACE		
albedoav	= 0.24	prescribed albedo value
/		
&NAMRADIATION		
lCnstAlbedo	= .true.	uses prescribed value above .false. value will compute albedo over the ocean
usero3	= .false.	uses climatological mean value .true. value will use ozone profile from backrad.nc
co2factor	= 1.	allows to modify climatological mean value
/		
&NAMRADSTAT		
lstat	= .true.	
dtav	= 1	
timeav	= 1.	
/		

Table 10: Some key namoptions settings for radiative transfer calculations. An example of a set to be used for the exercise is posted on Brightspace. The Dales code provides a description of many more switches that can be used for idealized studies with a constant solar zenith angle as applied in the CGILS experiment (Blossey et al., 2013).

- Download `namoptions.005` from Brightspace and adapt it (experiment number, number of vertical levels, correct filename extension).
- Execute Dales using only 1 processor.

References

Beare, R. J., and Coauthors, 2006: An intercomparison of large-eddy simulations of the stable boundary layer. *Boundary-Layer Meteorol.*, **118**, 247–272.

Blossey, P. N., and Coauthors, 2013: Marine low cloud sensitivity to an idealized climate change: The CGILS LES intercomparison. *J. Adv. Model. Earth Syst.*, **5**, 1–25.

Boers, R., T. Brandsma, and A. P. Siebesma, 2017: Impact of aerosols and clouds on decadal trends in all-sky solar radiation over the netherlands (19662015). *Atmos. Chem. Phys.*, **17**, 8081–8100.

Böing, S. J., 2014: The interaction between deep convective clouds and their environment. Ph.D. thesis, Technical University Delft, Delft, 133 pp (Available from Technical University Delft, Delft, The Netherlands, http://repository.tudelft.nl).

Brown, A. R., S. Derbyshire, and P. J. Mason, 1994: Large-eddy simulation of stable atmospheric boundary layers with a revised stochastic subgrid model. *Quarterly Journal of the Royal Meteorological Society*, **120 (520)**, 1485–1512.

Deardorff, J. W., 1973: Three-dimensional numerical modeling of the planetary boundary layer. *Workshop on Micrometeorology*, D. A. Haugen, Ed., Amer. Met. Soc. Boston, 271–311.

Deardorff, J. W., 1980: Cloud-top entrainment instability. *J. Atmos. Sci.*, **37**, 131–147.

Duynkerke, P. G., and Coauthors, 1999: Intercomparison of three- and one-dimensional model simulations and aircraft observations of stratocumulus. *Boundary-Layer Meteorol.*, **92**, 453–487.

Duynkerke, P. G., and Coauthors, 2004: Observations and numerical simulations of the diurnal cycle of the EUROCS stratocumulus case. *Quart. J. Roy. Meteorol. Soc.*, **130**, 3269–3296.

Heus, T., and Coauthors, 2010: Formulation of the dutch atmospheric large-eddy simulation (DALES) and overview of its applications. *Geosci. Model Development*, **3**, 415–444, doi:10.5194/gmd-3-415-2010.

Kleissl, J., C. Meneveau, and M. B. Parlange, 2003: On the magnitude and variability of subgrid-scale eddy-diffusion coefficients in the atmospheric surface layer. *J. Atmos. Sci.*, **60**, 2372–2388.

Larson, V. E., K. E. Kotenberg, and N. B. Wood, 2007: An analytic longwave radiation formula for liquid layer clouds. *Mon. Weather Rev.*, **135**, 689–699.

Lesieur, M., O. Métais, and P. Comte, 2005: *Large-eddy simulations of turbulence*. Cambridge University Press.

Mason, P. J., 1989: Large-eddy simulation of the convective atmospheric boundary layer. *J. Atmos. Sci.*, **46**, 1492–1516.

Mason, P. J., and D. J. Thomson, 1992: Stochastic backscatter in large-eddy simulations of boundary layers. *J. Fluid Mech.*, **242**, 51–78.

Meneveau, C., and J. Katz, 2000: Scale-invariance and turbulence models for large-eddy simulation. *Annu. Rev. Fluid Mech.*, **32**, 1–32.

Nieuwstadt, F. T., P. J. Mason, C.-H. Moeng, and U. Schumann, 1993: Large-eddy simulation of the convective boundary layer: A comparison of four computer codes. *Turbulent Shear Flows 8*, Springer, 343–367.

Schumann, U., 1975: Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *J. Comp. Phys.*, **18**, 376–404.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations. Part I: The basic experiment. *Mon. Weather Rev.*, **91**, 99–164.

Stevens, B., and Coauthors, 2005: Evaluation of large-eddy simulations via observations of nocturnal marine stratocumulus. *Mon. Weather Rev.*, **133**, 1443–1462.

A Linux tips

It is possible to open multiple terminals, for example one for the directory ∼/Cases and one for ∼/Experiments. If you want to quickly change from one directory to another, you can define a shortcut, e.g. `alias exp='cd ∼/Experiments'`. Put this line in the file ∼/.bashrc and type `source ∼/.bashrc` to activate the alias. Note that .bashrc will be executed automatically when you open a new terminal. The .bashrc file can be filled with as many aliases as you want.
Ubuntu includes an open source version of Matlab which is called Octave.

B Installing Linux on Ubuntu 16.04

First install the needed packages:

```
sudo apt install git
sudo apt install cmake
sudo apt install netcdf-dev
sudo apt install netcdf
sudo apt install ccmake
sudo apt install cmake-curses-gui
sudo apt install libnetcdf-dev
sudo apt install libopenmpi-dev
sudo apt install gfortran
sudo apt install libnetcdff-dev
sudo apt install doxygen
```

Next, clone the Dales git repository

```
cd ∼
mkdir dales
cd dales
git clone git@github.com:dalesteam/dales.git
cd dales
git checkout v4.1
```

Now build Dales from source code

```
cd ∼/dales
mkdir dales.build
cd dales.build
cmake ../dales
make
```

The dales4 executable is now in `src/dales`.