Dutch Atmospheric Large Eddy Simulation version 3.1, 'Kronenbourg'

Chiel van Heerwaarden (WUR)
Thijs Heus (KNMI)
Stephan de Roode (TU Delft)
www.ablresearch.org/dales
dales@ablresearch.org

January 30, 2009

Contents

1 Introduction

The incentive for this new version of DALES came from the need for a faster code, plus a less diffusive advection scheme than the Leap-frog/kappa scheme could provide. This is now achieved due to implementation of a third order Runge Kutta time integration scheme and (a.o.) a 5th order advection scheme. The time scheme is rather rigidly set throughout DALES, but another advection scheme can be easily added (should be clear by looking in advection.f90). The same goes for the surface handling scheme, the radiation scheme or the moisture handling. RK3 requires 3 intermediate time steps, but the increased stability allows increased timestepping. The first tests showed a decrease in runtime of up to 50%.
Another aspect of this version is the modular setup. The increasing amount of

specialized code (e.g. chemistry, microphysics, statistics, katabatic flow) makes it more difficult to share the core of the code. In the way the code is set up now, it should be possible to keep the core cleaner and the use and the sharing of additional forcings easier.

Known Issues

- When using OpenMPI, the MPI_CART_CREATE statement in modmpi.f90 at line 74/75 needs to be commented out, and line 78-79 needs to be activated - Issue found by Pedro Soares
- Microphysics may become unstable, in combination with a non-equidistant grid that increases the allowed time step. Workaround: Limit the peclet and/or courant numbers in the namelist.

Major Changes - Version 3.1

- Non-equidistant grid for 5th order, particles
- Mixed buoyancy mixing over the wet/dry interface
- Bulk microphysics implemented
- Revised radiation, subgrid and surface routines
- Adaptive timestepping
- More reliable statistical routines

Major Changes - Version 3.0

- 3rd order Runge Kutta integration
- Several advection schemes: 2nd, 4th and 6th order central as well as 3rd and 5th order schemes that are slightly diffusive and therefore nearly positive. The kappa scheme is also implemented.
- Allocatable arrays, meaning that new compilation is only necessary for real code changes, and not anymore for changes in resolution or number of processors. The number of processors is now not an input parameter anymore; it is read from MPI.
- New setup of the namelists; only one namelist (namoptions) file is now necessary. See section 4
- Major cleanup of the main program; most subroutines are now sorted into modules.
- The modular setup also allows for statistical routines and add-ons. See section 5 for a description of those.
- To keep bookkeeping of open files easy, files are always directly closed after use; as an added bonus, results are directly flushed into the files upon closing.
- guard constructions are replaced; thermodynamical 'constants' 0.622, 0.61, 1.61 and 0.378 are replaced with combinations of 1, rv and rd.

2 Roadmap / Wish List

- Chemistry addon
- Land surface scheme
- Full radiation scheme
- Non-periodic boundary conditions
- Optimized microphysics
- Deep convection / anelastic Navier Stokes solver
- Block instead of stripe parallelization

DALES has been tested (although not always thoroughly) on the following platforms:

- x86 with gcc 4.1.2, ifort 9.1 and ifort 10.0
- x86_64 with ifort 10.0
- itanium2 with ifort 9.1

3 Installation

To set the dependencies correctly, make sure you have `makedepf90` somewhere in your path. The source code as well as a binary for IBM p5 and one for x86 machines is available in the subdir makedepf90.

The makefile switches between various systems using the `\$SYST` variable that can be set outside the Makefile, for instance in your .bashrc or .cshrc script.

The makefile switches into debug mode with the debug flag set to true in the header of the Makefile.

4 Use

Please check the directories 'example' and 'exampleplugin' for working examples that illustrate the capabilities of DALES and give a sample set of input files.

namoptions

The general options file is called namoptions, but can be replaced by any other file using a command line option at the invocation of DALES:

```
dales3 namoptions.001
```

Namoptions consists by default of the namelists run, domain, physics and dynamics.

Namelist **run** contains:

- *iexpnr* number of the experiment

- *lwarmstart* flag for "cold" or "warm" start
- *startfile* name of the restart file
- *runtime* simulation time in secs
- *dtmax* time integration interval
- *ladaptive* flag for adaptive time stepping; supercedes dtmax
- *courant* maximum courant number in adaptive timestepping; optional
- *peclet* maximum peclet number in adaptive timestepping; optional.
- *trestart* each trestart sec. a restart file is written to disk
- *dtav_glob* default value for sampling of statistiscs
- *timeav_glob* default value for write interval of statistics
- *irandom* number to seed the randomnizer with
- *randqt* amplitude of the qt randomnization
- *randthl* amplitude of the thl randomnization
- *krand* top-height of randomnization
- *nsv* number of additional scalar fields. Maximum 999.

namelist **domain** contains:

- *imax jtot kmax* domain size in grid points
- *xsize ysize* domain size in meters
- *xlat xlon* latitude/longitude in degrees
- *xday* day of the year
- *xtime* GMT time
- *ksp* lower height of the sponge layer
- *chi_half* coefficient to determine wet (0) dry (1) or mixed (0.5) mixing over the cloud interface. Default 0.5
- *lneutraldrag*

namelist **physics** contains:

- *z0* surface roughness
- *ustin* prescribed friction velocity
- *wtsurf* prescribed surface thl-flux
- *wqsurf* prescribed surface qt-flux
- *wsvsurf* prescribed surface scalar-flux

- *ps* surface pressure
- *thls* surface liq. water pot. temperature
- *lmoist* switch to calculate moisture fields
- *isurf* flag for surface parametrization
- *iradiation* flag for radiation scheme, with 0 for no radiation, 1 for full radiation, 2 for parameterized radiation, 10 for user defined in the file rad_user.f90. If iradiation=2:
  - *rad_ls* switches directly imposed large scale radiative forcing from lscale.inp
  - *rad_longw* switches parameterized longwave radiation.
  - *rad_shortw* switches parameterized shortwave radiation.
  - *rad_smoke* switches parameterized for smoke-like radiation.
  - *isvssmoke* chooses the scalar used for the smoke calculations
  - *rka*
  - *gca*
  - *dlwtop* top of the domain radiative flux
  - *dlwbot* surface radiatve flux
  - *sw0*
  - *sfc_albedo* planetary albedo
  - *reff*
- *timerad* timescale of the radiation scheme. Defaults to every time step
- *lcoriol* switch for coriolis force
- *ltimedep* switch for timedependent fluxes and large scale forcings

namelist **dynamics** contains:

- *llsadv* switch for large scale forcings
- *lqlnr* Switch for Newton Raphson approximation of the liquid water content.
- *cu, cv* Transformation velocity of the Galilei transform.
- *iadv_mom iadv_tke iadv_thl iadv_qt iadv_sv(:).* Chooses the advection scheme for momentum, tke, thl, qt, and the scalars. Can be set to:
  - 1 = 1st order upwind (ONLY FOR SCALARS)
  - 2 = 2nd order central
  - 4 = 4th order central
  - 5 = 5th order
  - 6 = 6th order central
  - 7 = kappa (ONLY FOR SCALARS)

namelist **subgrid**

- *ldelta* switch for diminshed sfs in stable flow
- *lmason* reduced length scale near the surface
- *cf,cn,Rigc,Prandtl* sfs-parameters

namelist **microphysics**

- *imicro* flag for microphysical scheme. 0 for none, 1 for cloud water sedimentation only, 2 for bulk microphysics
- *l_sb* switch between SeifertBeheng parameterization (default) and KhairoutdinovKogan
- *l_rain,l_sedc* switches rain and cloud water sedimentation
- *l_mur_cst, mur_cst, Nc0, sig_g, sig_gr*

prof.inp.$iexp

Used to generate the initial values of the variables in case of a cold start. The first two lines are unimportant header lines. From there onwards, *kmax* rows give in 6 columns the initial profile of height (used to check whether the grid is equidistant), θ_l, q_t u, v, and subfilter scale TKE.

lscale.inp.$iexp

Gives the large scale forcings. The first two lines are unimportant header lines. The third and onwards show 8 columns: height, geostrophic wind in the x-direction, geostrophic wind in the y-direction, large scale subsidence, large scale inflow of moisture in the x-direction and in the y-direction, additional large scale moisture change in time and the radiation tenency of θ_l.

ls_flux.inp.$iexp

Gives timedependent forcings and fluxes, if *ltimedep =.true.* The first block starts with three header lines, followed by lines reading the time, surface heat flux and surface moisture flux at those times, this goes on until at least runtime.

The second block and onwards always starts with a header line that starts with a #, followed by a time. The next row and onwards of the block contains the large scale forcings in a lineup identical to lscale.inp. These blocks are followed upto at least runtime has passed.

The file **ls_fluxsv.inp.$iexp** has a similar set up for scalar treatment.

scalar.inp.$iexp

Gives the initial scalar profiles. First two rows are unimportant header. Then onwards in *nsv+1* columns is the height versus the respective scalar profiles.

5 Add-Ons and Statistical Routines

To keep DALES fast and tidy, add-ons and statistical routines are now moved to seperate modules. The difference between an add-on and a statistical routine is that the former may alter the prognostics of the code; this is also expressed in their respective places in program.f90

The idea is to keep the impact of these routines minimal: if possible (and this should be nearly always the case) only a use statement (sec. 0.1 of program.f90) an init statement (sec 2), execution (sec 3.4 resp. 3.9) and if necessary finalization (sec. 4). To ensure that variable or subroutine names are not interfering, a 'private' statement should be included.

Each add-on may have it's own switches and options; for example the sampling and output timestep of the statistical routines (used to be the globally set dtav and timeav). These are set by adding an additional namelist to namoptions; it should at least contain a switch on when to enter the module; that should be set to .false. by default. This way, one could omit namelists that one doesn't use.

Short description of available add-ons

modnudge θ_l and q_t are kept at their initial profiles by exertion of a source term $-\frac{<\phi>-<\phi_{init}>}{\tau}$. Namelist namnudge contains:

- *tnudge*, the timescale of the source term. tnudge = 0 (default) disables nudging.

modparticles Calculates Lagrangian tracks of massless tracer particles. Namelist namparticles contains:

- *lpartic* switches the routine on/off
- *lpartsgs* switches the subgrid diffusion on/off
- *intmeth* time integeration scheme; 0 = particles stand still; 2 = runge kutta
- *dtpart* integration time step
- *startfile* name of the particle start-up file this text file has the number of particles on the first line and then for every particle a line with initial t x y z in physical coordinates
- *lstat* switches the particle statistics on/off
- *dtav* time interval for sampling of statistics
- *timeav* time interval for writing of statistics
- *ldump* switches the dump of the particle field on/off
- *timedump* time interval for dump of particle field
- *npartdump* number of variables written at timedump in order:x, y, z, u, v, w, thl, thv, qt, ql

modnetcdfstats Statistical output in NetCDF format stored in x, z, t format

- *lnetcdfstats* switches the routine on/off

- *dtav* time interval for writing of statistics
- *ncklimit* number of vertical levels to save

modnetcdfmovie xy- or yz-slices in NetCDF format

- *lnetcdfmovie* switches the routine on/off
- *dtmovie* time interval for writing slices
- *ncklimit* number of vertical levels to save
- *lmoviez* $.true.$ creates yz-slice, $.false.$ creates xy slice
- *slicex* x-coordinate for yz slice
- *slicez* z-coordinate for xy slice

modtilt manages a tilted boundaryy layer for katabatic flow

Short description of available statistics

modchecksim Calculates divergence and Courant number. Namelist namchecksim contains:

- *tcheck* interval between checks. Default is every timestep.

modgenstat calculates generic slabaveraged statistics. Namelist namgenfield contains:

- *dtav* sampling interval
- *timeav* interval of writing
- *lstat* switch to enable timeseries

modsampling calculates conditional sampled fields. Namelist namsampling contains:

- *dtav* sampling interval
- *timeav* interval of writing
- *lsampco* switch to sample cloud ($q_l > 0$)
- *lsampcl* switch to sample cloud core ($q_l > 0$,$\theta_v > 0$)
- *lsampupd* switch to sample updrafts ($w > 0$)

modtimestat calculates timeseries of several variables. Namelist namtimestat contains:

- *dtav* sampling interval
- *ltimestat* switch to enable timeseries

modcrosssection dumps an instantenous crosssection of the field. Namelist namcrosssection contains:

- *dtav* sampling interval
- *lcross* switch to enable crosssection
- *crossheight* height of the xy-crosssection

- *crossplane* location of the yz-plane on every processor

modcloudfield dumps all the wet points in the field. Namelist namcloudfield contains:

- *dtav* sampling interval
- *timeav* interval of writing
- *lcloudfield* switch to enable cloudfield dumps
- *laddinfo* switch to write ql and w values

modfielddump dumps complete 3d-fields of u,v,w, thl, qt and ql in 2-byte integers. Namelist namfielddump contains:

- *dtav* sampling interval
- *lfielddump* switch to enable fielddump
- *ldiracc* switch to dump in direct access files instead of Fortran unformatted files.

6 Coding Principles in DALES3

- Instead of tab, use a double space as indention
- Try to avoid 0.61, 1.61 and friends, but use combinations of Rd and Rv. In general, use constants instead of numbers.
- Don't add anything case specific to the core modules. If you need to change a hard coded constant, consider adding it to a namelist.
- Make an attempt to include some documentiation of new and improved features. A lot of features risk of never being used, and even being reimplemented once in a while, just because nobody knows they're there.

7 Code description

For a comprehensive description of the code, Heus et al. [2009] is in preparation.

8 Contact

If you have a bug report, please submit it at the wiki of www.ablresearch.org/dales , preferably including a bug fix. If you want to contact the developers, please mail to dales@ablresearch.org . A mailinglist for users is available as dales-users@ablresearch.org .

9 Terms of use

DALES is being released under the GPLv3 license; see the file COPYING in the package. DALES is free to use, modify or redistribute. It is regarded common courtesy to report bugs, bug fixes, and appreciated if code that could be of use for other users is contributed to the main code. In case of a (first) publication based on DALES results, please contact the development group to discuss co-authorship. In any case, always refer to this code using the name DALES (so no IMAU-LES, KNMI-LES or WUR-LES defining the same code), and cite Heus et al. [2009] as the reference.

References

Thijs Heus, Chiel C. van Heerwaarden, Harm J. J. Jonker, A. Pier Siebesma, Simon Axelsen, Olivier Geoffroy, Arnold Moene, David Pino, Stephan R. de Roode, and Jordi Vilà-Guerau de Arellano. An overview of the Dutch Atmospheric LES. *Geoscientific Model Development*, 2009. In preparation.