Parcel model documentation

Anna Zimniak and Anna Jaruga

Institute of Geophysics, Faculty of Physics, University of Warsaw, Poland

January 15, 2016

1 Introduction

The parcel model represents an idealised scenario of a 0-dimensional air parcel rising adiabatically with a constant vertical velocity. Because of its simplicity, the parcel approach provides computationally efficient testbed for cloud microphysics schemes.

Representation of microphysical and chemical processes in the parcel model is done using the particle-based scheme. The particle-based scheme (aka Lagrangian scheme) allows to track the properties of both aerosol particles and cloud droplets throughout the entire simulation. The processes resolved by the particle-based scheme cover: (i) condensational growth of aerosols and cloud droplets, (ii) collisions, (iii) sedimentation[1], (iv) aqueous phase chemical reactions. All the processes can be easily switched on/off by the user.

We are using the particle-based scheme from the *libcloudph++* library. This is a library of algorithms for representing cloud microphysics in numerical models. The introduction to the *libcloudph++* containing description of the particle-based scheme used in the parcel model, complete description of the programming interface of the library and a performance analysis is available online at [1].

2 Solved equations

The particle-based microphysics scheme used within the parcel model expects dry air density ρ_d as one of the input arguments. To evaluate ρ_d, the profile of pressure within the parcel model is predicted by integrating the hydrostatic equation:

$$\frac{dp}{dz} = -\rho g, \tag{1}$$

where p - pressure, z - vertical displacement, ρ - density of air, g - gravitational acceleration. A method used in the parcel model to integrate eq. 1 is to assume a piecewise constant profile of ρ. As a result, at a given time level n the pressure used to predict ρ_d is defined as

$$p^n = p^{n-1} - \rho^{n-1} g z^n.$$

The method uses following function to retrieve dry air density from the pressure:

$$\rho_d(p, \theta, r_v) = \frac{p - p_v(p, r_v)}{R_d \theta (\frac{p}{p_{1000}})^{\frac{R_d}{c_{pd}}}}, \tag{2}$$

where $p_v(p, r_v)$ represents partial pressure of water vapour. The dry air density at a given time level n is calculated as $\rho_d(p^n, \theta^n, r_v^n)$. The pressure outputted by the parcel model is the same as the one used to calculate dry air density for the microphysics scheme.

[1] Should be switched off when used in 0-dimensional setting

The derivation of source terms of heat and moisture due to microphysical processes is available in Appendix A of [1].

3 Initial condition

The initial thermodynamic condition must be set below supersaturation, i.e.

$$r_0 < \epsilon \frac{e_s(T_0)}{p_0 - e_s(T_0)}, \tag{3}$$

where r_0 is initial water vapour mass mixing ratio, $e_s(T_0)$ is water vapour saturation pressure at initial temperature T_0, p_0 is initial pressure and $\epsilon = 0.622$.

The initial lognormal monomodal size distribution of dry aerosol is assumed:

$$n(r) = \frac{n_{tot}}{r\sqrt{2\pi} ln(\sigma_g)} exp(-\frac{(ln(r) - ln(\overline{r}))^2}{2ln^2(\sigma_g)}) \tag{4}$$

where $n(r)$ is spectral density function for particles radius r, n_tot is total aerosol concentration, $\overline{r}$ is mode radius and σ_g is geometric standard deviation. The parametres of the size distribution can be specified by the user.

If chemical reactions are enabled, the model assumes that the initial aerosol is ammonium bisulfate aerosol. The initial mass of chemical compounds is then calculated using the initial dry radius from initial dry aerosol size distribution and assumed dry particle density of 1.8 g/cm^3.

4 Arguments

The arguments of the parcel model are divided into 4 groups:

4.1 thermodynamic variables

- **T_0** : float (default = 300);
  initial temperature [K]
- **p_0** : float (default = 101300);
  initial pressure [Pa]
- **r_0** : float (default = 0.022);
  initial water vapour mass mixing ratio [kg/kg].
- **w** : float (default = 1);
  Updraft velocity [m/s]

4.2 aerosol attributes

- **mean_r** : float (default = 0.04e-6);
  lognormal distribution mode radius [m]
- **gstdev** : float (default = 1.4);
  lognormal distribution geometric standard deviation [1]
- **n_tot** : float (default = 60e6);
  lognormal distribution total concentration in standard conditions (T=20C, p=1013.25 hPa, r=0) [m^{-3}]

- **kappa** : float (default = 0.5);
  κ hygroscopicity parameter (see [2] for more details)

- TODO - mole fractions of trace gases

4.3 simulation parameters

- **dt** : float (default = 0.1);
  timestep [s]

- **z_max** : float (default = 200);
  maximum vertical displacement [m]

- **sd_conc** : int (default = 64);
  number of super-droplets used by the particle-based microphysics scheme

- **pprof** : string (default = "pprof_piecewise_const_rhod");
  method to calculate pressure profile used to calculate dry air density that is used by the super-droplet scheme. The parcel model uses method described in sec. 2. It is possible to swich on another option: "pprof_const_th_rv" (see Appendix A for details).

- TODO - open/closed chem system

4.4 output parametres

- **outfile** : string (default = "test.nc");
  output file name; the output file is in netCDF format

- **outfreq** : int (default = 100);
  output frequency (time gap between outputted points in number of time steps)

- **out_bin** : jason string (default =
  '{"radii": {"rght": 0.0001, "moms": [0], "drwt": "wet", "nbin": 26, "lnli": "log", "left": 1e-09}}');

  It is a dictionary of dictionaries defining spectrum diagnostics. First key defines the name of created variable. The following dictionary defines spectrum characteristics:
  · "rght", "left" - right and left edge of the spectrum (in meters)
  · "moms" - list of numbers, specifying moments of the spectrum
  · "drwt" - choice between ("dry") for dry aerosol spectrum and ("wet") for wet particles
  · "nbin" - number of bins
  · "lnli" - linear ("lin") or logaritmical ("log") spacing between between bins

  For example, user can define two variables - "A" and "B" :

$$\text{"A"} : \begin{cases} \text{"rght"}: 0.0001 \\ \text{"moms"}: [0] \\ \text{"drwt"}: \text{"wet"} \\ \text{"nbin"}: 26 \\ \text{"lnli"}: \text{"log"} \\ \text{"left"}: 1e\text{-}09 \end{cases}$$

$$\text{"B"} : \begin{cases} \text{"rght"}: 2.5\text{e-}05 \\ \text{"moms"}: [0, 1, 2, 3] \\ \text{"drwt"}: \text{"wet"} \\ \text{"nbin"}: 49 \\ \text{"lnli"}: \text{"lin"} \\ \text{"left"}: 5\text{e-}07 \end{cases}$$

It will generate five output spectra: 0-th spectrum moment for 26 bins spaced logaritmically between 10^{-9} and 10^{-4} m for wet radius for variable 'A' and 0, 1, 2 and 3-rd moments for 49 bins spaced linearly between $5 \cdot 10^{-7}$ and $2.5 \cdot 10^{-5}$ m for wet radius for variable 'B'.

TODO - chem output

5 Output

The parcel model uses NetCDF file format[2] for output. The content of the output file can be be viewed in terminal by using the `ncdump` command. Output variables describing time-dependent ambient conditions in the parcel are:

- **t** - time [s],
- **z** - height above the starting point[m],
- **p** - pressure [Pa],
- **r_v** - water vapour mixing ratio [kg/kg],
- **RH** - relative humidity [1],
- **T** - temperature [K],
- **th_d** - dry potential temperature [K],
- TODO - chem trace gases.

  There is also one value for the entire simulation:

- **RH_max** - maximum relative humidity reached during the simulation [1].

Output variables describing size distribution of particles from microphysics scheme depend on the user-defined "out_bin" parameter. For example, for the default setting of "out_bin" parameter there are two output variables describing placing and spacing of size distribution bins:

- **radii_r_wet** - left edges of the bins of the spectrum histogram [m],
- **radii_dr_wet** - bins width [m]

and one output variable containing the time-dependent chosen moment of size distribution:

- **radii_m0** - 0th moment of spectrum.

[2] see unidata.ucar.edu/software/netcdf/ for details

6 Installation

The parcel model requires the *libcloudph++* library to be installed. The *libcloudph++* library can be obtained from the project github repository at https://github.com/igfuw/libcloudphxx. The library dependencies and installation guidelines are available there in a Readme file.

The parcel model is available online at the project repository at https://github.com/igfuw/parcel. The installation guidelines are available there in a Readme file. The parcel model is written in Python 2.7 and was not tested in Python 3.0 or latter.

7 Usage examples

The parcel model is shipped with a set of tests designed for checking the particle-based microphysics scheme from the *libcloudph++* library. The tests may serve as usage examples for the parcel model and *libcloudph++* library. The tests are located in the **unit_test** and **long_test** folders. The tests use the pytest python package for test automation and can be run by typing `py.test unit_test` from a terminal. Example output plots generated by the tests are saved in the plots\outputs folder.

7.1 bash

To run the model using default parameters please type in terminal:

```
python parcel.py
```

Any arguments to parcel model options can be passed via command line. For example:

```
python parcel.py --outfile 'test2.nc' --p_0 100000 --T_0 280 --r_v0 0.025 --out_bin '{"radii":
{"rght": 0.0001, "moms": [0], "drwt": "wet", "nbin": 26, "lnli": "log", "left": 1e-09},
"cloud": {"rght": 2.5e-05, "moms": [0, 1, 2, 3], "drwt": "wet", "nbin": 49, "lnli": "lin",
"left": 5e-07}}'
```

will generate an output file 'test2.nc', which contains results of the simulation with inital pressure 1000 hPa, initial temperature 280 K, initial water vapor mixing ratio 25 g/kg and two spectrum variables - 'radii' and 'cloud' (the same as described in sec. 4.4). Other arguments will be default.

7.2 gdl/idl

The following example shows how to use the parcel model from the IDL (or its free counterpart GDL) programming languages. It takes advantage of the command line interface of the parcel model and executes the simulation via the `spawn` command. It then plots the size distribution of particles at different timelevels.

list.GDL1 (gdl/idl)

```
pro test
  ; specifying parameters
  gstdev = 1.5
  outfile = 'idltest.nc'

  ; executing the simulation
  spawn, [ $
    'python', 'parcel.py', '--outfile', outfile, '--gstdev', string(gstdev) $
  ], /noshell

  ; opening the output file
  nc = ncdf_open(outfile, /nowrite)

  ; plotting initial and final wet spectra
  ncdf_diminq, nc, ncdf_dimid(nc, 'radii'), ignore, nr
  ncdf_diminq, nc, ncdf_dimid(nc, 't'    ), ignore, nt

  !P.MULTI=[0,2,1]
  ncdf_varget, nc, 'radii_dr', radii_dr
  ncdf_varget, nc, 'radii_rl', radii_rl

  foreach it, [0, nt-1] do begin
    ncdf_varget, nc, 'radii_m0', radii_m0, count=[nr, 1], offset=[0, it]
    plot,                                 $
      xtitle='particle wet radius [um]', $
      ytitle='[mg-1 um-1]',              $
      psym=10, /xlog, /ylog,             $
      (radii_rl+radii_dr/2)*1e6, (radii_m0/1e3)/(radii_dr*1e6)
  endforeach
end
```

7.3 python

The following example shows how to use the parcel model from python programming language and plot vertical profiles for pressure, temperature and relative humidity from netCDF file.

list.P1 (Python)

```
from scipy.io import netcdf

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt

from parcel import parcel

# define parcel arguments
outfile = "example_py.nc"
dt      = 0.5
outfreq = 10

#run parcel model
parcel(dt=dt, outfreq = outfreq, outfile=outfile)

# open ncdf file with model results
ncfile = netcdf.netcdf_file(outfile)

# plot the results
plt.figure(1, figsize=(20,10))
plots    = []
legend_l = []
for i in range(3):
    plots.append(plt.subplot(1,3,i+1))

plots[0].set_xlabel('p [hPa]')
plots[1].set_xlabel('T [K]')
plots[2].set_xlabel('RH')

for ax in plots:
    ax.set_ylabel('z [m]')

z = ncfile.variables["z"][:]
plots[0].plot(ncfile.variables["p"][:] / 100. , z)
plots[1].plot(ncfile.variables["T"][:]       , z)
plots[2].plot(ncfile.variables["RH"][:]      , z)

plt.savefig("doc_python.pdf")
```

Appendix A

Another method for integrating equation (1) and retrieving pressure profile is to assume constant potential temperature, θ^0, and constant water vapour mixing ratio, r_v^0 (option "pprof_const_th_rv", see sec. 4.3).

As a result at a given time level n the pressure used to predict ρ_d is defined as

$$p^n = p_{1000} \left((\frac{p^0}{p_{1000}})^{\frac{R_d}{c_{pd}}} - \frac{R_d g (z^n - z^0)}{c_{pd} \theta^0 R(r_v^0)} \right)^{\frac{c_{pd}}{R_d}},$$

where p_{1000} stands for pressure equal 1000 hPa that comes from the definition of potential temperature, $R(r_v^0)$ is the gas constant for moist air and c_{pd} is the specific heat at constant pressure for dry air.

The function (2) is used to retrieve dry air density from pressure. The dry air density at a given time level n is calculated as $\rho_d(p^n, \theta^0, r_v^0)$. The pressure outputted by the parcel model is then calculated as $\rho_d(R_d + r_v R_v)T$, where R_v is the gas constant for water vapor.

This method follows the procedure used in the 2-dimensional kinematic model *icicle*, described in [1] and was added to the parcel model to allow better comparison between the two setups.

Acknowledgements

References

[1] S. Arabas, A. Jaruga, H. Pawlowska, and W.W. Grabowski. libcloudph++ 0.2: single-moment bulk, double-moment bulk, and particle-based warm-rain microphysics library in c++. *Geosci. Model. Dev.*, 8(6):1677–1707, 2015.

[2] M.D. Petters and S.M Kreidenweis. A single parameter representation of hygroscopic growth and cloud condensation nucleus activity. *Atmos. Chem. Phys.*, 7:1961–1971, 2007.