Parcel model documentation

Anna Zimniak and Anna Jaruga

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

March 4, 2020

1 Introduction

The parcel model represents an idealized 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 Arabas et al. [2015]. The aqueous phase chemistry module of *libcloudph++* is focused on the oxidation of SO2 dissolved inside water drops to sulfate. Two reaction paths are included - oxidation by O3 and H2O2. A description of the representation of aqueous phase chemistry in *libcloudph++* is available in the PhD thesis of Jaruga.

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 represents pressure, z is the vertical displacement, ρ is the density of air and g is the gravitational acceleration. To integrate Eq. (1) it is assumed that ρ is constant at each height level (i.e. piecewise constant profile). 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 following function is used to retrieve dry air density from the pressure:

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

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

where $p_v(p, r_v)$ represents partial pressure of water vapor, p_{1000} stands for pressure equal 1000 hPa that comes from the definition of potential temperature, R_d is the gas constant for dry air and c_{pd} is the specific heat at constant pressure for dry air. 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.

The parcel model uses water vapor mixing ratio and dry air potential temperature as model variables, see Arabas et al. [2015] for definition. Both variables can only be changed due to the microphysical processes (the model is adiabatic and includes only microphysics and aqueous phase chemistry). The derivation of source terms of heat and moisture due to microphysical processes is available in Appendix A of [Arabas et al., 2015]. For simulations with aqueous phase chemistry six additional model variables are needed that specify the mixing ratio of SO2, O3, H2O2, CO2, NH3 and HNO3 trace gases in the model. The aqueous phase chemistry module of *libcloudph++* changes the trace gas mixing ratios as described in Chapter 3.3 of the PhD thesis of Jaruga.

3 Initial condition

The user needs to specify the initial temperature T_0, pressure p_0 and water vapor mixing ratio r_0. Instead of explicitly specifying r_0, the user can specify the initial relative humidity RH_0. 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 $e_s(T_0)$ is water vapor saturation pressure at initial temperature and $\epsilon = 0.622$. The initial size distribution of dry aerosol can be a combination of any number of lognormal distributions:

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

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 parameters of the size distribution can be specified by the user. Each of the lognormal modes can have a different value of the hygroscopicity parameter κ.

For simulations with aqueous phase chemistry the initial trace gas mixing ratios need to be additionally specified. If chemical reactions are enabled, the model assumes that the initial aerosol is ammonium bisulfate aerosol (NH4HSO4). The initial mass of chemical compounds is then calculated using the initial dry aerosol size distribution and assumed dry particle density of 1.8 g/cm^3. The dry particle density can be changed by the user.

4 Arguments

The arguments of the parcel model are divided into 5 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 vapor mass mixing ratio [kg/kg].
- **RH_0** : float (no default value);
  initial relative humidity, alternative to the **r_0** option.
- **w** : float (default = 1);
  Updraft velocity [m/s]

4.2 chemistry module variables

- **SO2_g** : float (default = 0);
  initial SO2 trace gas mixing ratio [kg/kg]
- **O3_g** : float (default = 0);
  initial O3 trace gas mixing ratio [kg/kg]
- **H2O2_g** : float (default = 0);
  initial H2O2 trace gas mixing ratio [kg/kg]
- **CO2_g** : float (default = 0);
  initial CO2 trace gas mixing ratio [kg/kg]
- **NH3_g** : float (default = 0);
  initial NH3 trace gas mixing ratio [kg/kg]
- **HNO3_g** : float (default = 0);
  initial HNO3 trace gas mixing ratio [kg/kg]
- **chem_rho** : float (default = 1800);
  assumed dry particulate density [kg/m3]
- **chem_dsl** : bool (default = false);
  switch for dissolving trace gases into droplets
- **chem_dsc** : bool (default = false);
  switch for dissociation of chemical compounds in droplets
- **chem_rct** : bool (default = false);
  switch for aqueous phase reactions in droplets

4.3 aerosol attributes

- **aerosol** : jason string (default =
  '{"ammonium_sulfate": {"kappa": 0.61, "mean_r": [0.02e-6], "gstdev": [1.4], "n_tot": [60.0e6]}}');

  It is a Python dictionary of dictionaries defining initial dry aerosol spectrum. First key defines the name of the aerosol type. The following dictionary defines the spectrum of aerosol of this type:
  · "kappa" - hygroscopicity parameter of the aerosol (see [Petters and Kreidenweis, 2007] for more details)
  · "mean_r" - list of mean radii of lognormal modes of aerosol of this type [m]
  · "gstdev" - list of geometric standard deviations of lognormal modes of aerosol of this type
  · "n_tot" - list of concentrations (at the STP dry air density $\rho_d = 1.225$ kg/m^3) of lognormal modes of aerosol of this type [m^{-3}]

For example, user can define two types of aerosol - bimodal ammonium bisulfate and monomodal sea salt :

$$\text{”ammonium\_bisulfate” :} \begin{cases} \text{”kappa”: 0.61} \\ \text{”mean\_r”: [0.02e-6, 0.07e-7]} \\ \text{”gstdev”: [1.4, 1.2]} \\ \text{”n\_tot”: [120.0e6, 80.0e6]} \end{cases}$$

$$\text{”sea\_salt” :} \begin{cases} \text{”kappa”: 1.28} \\ \text{”mean\_r”: [2e-6]} \\ \text{”gstdev”: [1.6]} \\ \text{”n\_tot”: [1e3]} \end{cases}$$

4.4 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 switch on another option: ”pprof_const_th_rv” (see Appendix A for details).

4.5 output parameters

- **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 Python 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” :

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

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

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'.

For simulations with aqueous phase chemistry the "moms" key can also be used for outputting the total mass of chemical compounds dissolved in droplets. The allowed values for moms are "SO2_a", "CO2_a", "NH3_a", "HNO3_a", "O3_a", "H2O2_a", "S_VI" and "H". The first six output the total dissolved mass of the corresponding trace gases. The "S_VI" outputs the mass of created H2SO4 and the "H" outputs the mass of H^{+}*ions.Thenumberofbinsandthesizerangeofdropletsselectedforoutputisdefinedinthesamewayasforthe*

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. All the initial parameters of the parcel model are saved as global attributes of the output file. Additionally the maximum relative humidity reached during simulation **RH_max** is also saved as a global attribute. Output variables describing time-dependent ambient conditions in the parcel are saved as:

- **t** - time [s],
- **z** - height above the starting point[m],
- **p** - pressure [Pa],
- **r_v** - water vapor mixing ratio [kg/kg],
- **RH** - relative humidity [1],
- **T** - temperature [K],
- **th_d** - dry potential temperature [K],
- **SO2_g, O3_g, H2O2_g, CO2_g, NH3_g, HNO3_g** - trace gas mixing ratios [kg/kg of dry air]

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

- **SO2_a, O3_a, H2O2_a, CO2_a, NH3_a, HNO3_a** - total number of moles dissolved in droplets [moles/kg of dry air]

Output variables describing size distribution of particles or mass of the dissolved chemical compounds from the particle tracking 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.

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.

7 Usage examples

7.1 bash

To run the model using default parameters 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 initial 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. 4.5). Other arguments will be default.

7.2 py.test

The parcel model is shipped with a set of tests designed for checking the particle-based microphysics and aqueous phase chemistry schemes 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 py.test Python package for test automation and can be run by typing from terminal:

```
py.test unit_test
```

To enable more output during testing type:

```
py.test -s -v long_test
```

Example output plots generated by the tests are saved in the `plots/outputs` folder.

7.3 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)

```
kpro test
c+cSingleline  ; specifying parameters
  gstdev o= l+m1.l+m5
  outfile o= idltest.nc

c+cSingleline  ; executing the simulation
  n+nbspawn, [
    python, parcel.py, ooutfile, outfile, ogstdev, n+nbstring(gstdev)
  ], o/noshell

c+cSingleline  ; opening the output file
  nc o= n+nbncdfopen(outfile, o/nowrite)

c+cSingleline  ; plotting initial and final wet spectra
  n+nbncdfdiminq, nc, n+nbncdfdimid(nc, radii), ignore, nr
  n+nbncdfdiminq, nc, n+nbncdfdimid(nc, t    ), ignore, nt

  !P.MULTIo=[l+m0,l+m2,l+m1]
  n+nbncdfvarget, nc, radiidr, radiidr
  n+nbncdfvarget, nc, radiirl, radiirl

  kforeach it, [l+m0, ntol+m1] kdo kbegin
    n+nbncdfvarget, nc, radiim0, radiim0, counto=[nr, l+m1], offseto=[l+m0, it]
    n+nbplot,
      xtitleo=particle wet radius [um],
      ytitleo=[mgol+m1 umol+m1],
      psymo=10, o/xlog, o/ylog,
      (radiirlo+radiidro/l+m2)o*1e6, (radiim0o/1e3)o/(radiidro*1e6)
  kendforeach
kend
```

7.4 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)

```
k+knfrom n+nnscipy.io k+knimport nnetcdf

k+knimport n+nnmatplotlib
nmatplotlibo.nusep(l+sl+sAggl+sp)
k+knimport n+nnmatplotlib.pyplot k+knas n+nnplt

k+knfrom n+nnparcel k+knimport nparcel

c define parcel arguments
noutfile o= l+sl+sexamplepy.ncl+s
ndt      o= l+m+mf0.5
noutfreq o= l+m+mi10

crun parcel model
nparcelp(ndto=ndtp, noutfreq o= noutfreqp, noutfileo=noutfilep)

c open ncdf file with model results
nncfile o= nnetcdfo.nnetcdffilep(noutfilep)

c plot the results
nplto.nfigurep(l+m+mi1p, nfigsizeo=p(l+m+mi20p,l+m+mi10p)p)
nplots    o= p[p]
nlegendl o= p[p]
kfor ni o+owin n+nbrangep(l+m+mi3p)p:
    nplotso.nappendp(nplto.nsubplotp(l+m+mi1p,l+m+mi3p,nio+l+m+mi1p)p)

nplotsp[l+m+mi0p]o.nsetxlabelp(l+sl+sp [hPa]l+sp)
nplotsp[l+m+mi1p]o.nsetxlabelp(l+sl+sT [K]l+sp)
nplotsp[l+m+mi2p]o.nsetxlabelp(l+sl+sRHl+sp)

kfor nax o+owin nplotsp:
    naxo.nsetylabelp(l+sl+sz [m]l+sp)

nz o= nncfileo.nvariablesp[l+sl+szl+sp]p[p:p]
nplotsp[l+m+mi0p]o.nplotp(nncfileo.nvariablesp[l+sl+spl+sp]p[p:p] o/ l+m+mf100. p, nzp)
nplotsp[l+m+mi1p]o.nplotp(nncfileo.nvariablesp[l+sl+sTl+sp]p[p:p]        p, nzp)
nplotsp[l+m+mi2p]o.nplotp(nncfileo.nvariablesp[l+sl+sRHl+sp]p[p:p]       p, nzp)

nplto.nsavefigp(l+sl+sdocpython.pdfl+sp)
```

Appendix A

Another method for integrating equation (1) and retrieving pressure profile is to assume constant potential temperature, $\theta = \theta^0$, and constant water vapor mixing ratio, $r_v = r_v^0$ (option "pprof_const_th_rv", see sec. 4.4).

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 [Arabas et al., 2015] and was added to the parcel model to allow better comparison between the two setups.

References

S. Arabas, A. Jaruga, H. Pawlowska, and W.W. Grabowski. libcloudph++ 1.0: a single-moment bulk, double-moment bulk, and particle-based warm-rain microphysics library in c++. *Geosci. Model. Dev.*, 8(6):1677–1707, 2015. doi: 10.5194/gmd-8-1677-2015. URL http://www.geosci-model-dev.net/8/1677/2015/.

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. doi: 10.5194/acp-8-6273-2008. URL http://www.atmos-chem-phys.net/8/6273/2008/.