Exercise 5:
Moist flow and microphysics - Part II

The Kessler scheme used in Exercise 4.2 is one of many parameterizations of the warm-phase microphysics. The parameterization is solely based on the mixing ratios q_v, q_c and q_r, but it does not include any information on the droplet size distribution. For instance, collision of cloud droplets and accretion of cloud droplets by falling rain drops highly depend on the size of the droplets. This deficiency is (partly) overcome in the two-moment scheme (Seifert and Beheng, 2005) by introducing two new prognostic variables (the so-called second moments): the number densities of cloud droplets n_c and of rain droplets n_r. The goal of Exercise 5.1 is to apply the two-moment scheme and to compare the simulation results to the Kessler scheme.

The processes parameterized by the microphysics scheme comprise diabatic processes (e.g. condensation of water vapor to cloud droplets) for which $\dot{\theta} \neq 0$. However, when implementing the dynamical equations in Exercise 2, we assumed adiabatic flow. Thus, the dynamics was artificially decoupled from the microphysics. This is remedied in Exercise 5.2 by including the vertical motion arising due to $\dot{\theta}$ into the dynamical equations.

5.1 The warm-phase two moment scheme

The two-moment scheme parameterizes, although in a slightly different way, the same processes as the Kessler scheme does for the first moments (i.e. for q_v, q_c and q_r). The number densities n_c and n_r change due to autoconversion from cloud droplets to rain droplets (CC), the accretion of cloud droplets by rain droplets (AC) and the condensation of vapor and evaporation of rain (G). An additional process, which only modifies n_r, is the self-collection and breakup of rain droplets (SC). Thus, the equations for the five prognostic microphysical fields read:

$$\frac{\partial q_v}{\partial t} + u\frac{\partial q_v}{\partial x} = -G_1 + EP_1$$
$$\frac{\partial q_c}{\partial t} + u\frac{\partial q_c}{\partial x} = G_1 - CC_1 - AC_1$$
$$\frac{\partial q_r}{\partial t} + u\frac{\partial q_r}{\partial x} = CC_1 + AC_1 - EP_1$$
$$\frac{\partial n_c}{\partial t} + u\frac{\partial n_c}{\partial x} = -CC_2 - AC_2 + G_2$$
$$\frac{\partial n_r}{\partial t} + u\frac{\partial n_r}{\partial x} = CC_2 + SC_2 - EP_2$$

where the indices 1, 2 denote the tendencies either for the first or the second moments.

In order for the two-moment scheme to work properly, we have to add the prognostic tendencies for the two new variables n_c and n_r. In *solver.py* these variables are already defined as *ncold*, *ncnow*, *ncnew* and *nrold*, *nrnow*, *nrnew*.

- Use the namelist switch *imicrophys* to choose between the Kessler (1) and the two-moment (2) scheme.
- Add the call of the prognostic step for the n_c and n_r variables in *solver.py* at *timestep for moisture scalars*. It should only be called for the namelist setting *imicrophys=2*.
  The corresponding subfunction is *prog_numdens* within *prognostics.py*, which needs to be completed with the advective tendencies $n_{ci,k}^{n+1}, n_{ri,k}^{n+1}$. Do not forget to exchange the variables for the next timestep, and to clip the negative values (only when *imicrophys=2*)!
- Call the two-moment scheme at *Exercise 5.1 Two Moment Scheme* in *solver.py* in the same fashion as you called the Kessler scheme:
  [lheat,qvnew,qcnew,qrnew,tot prec,prec,ncnew,nrnew]=seifert(unew,th0,prs, snew,qvnew,qcnew,qrnew,exn,zhtold,zhtnow,tot_prec,prec,ncnew,nrnew)
- The evaporation of rain droplets can again be switched on/off by changing *iern* in *namelist.py*.
- Compare several simulations for different configurations of the model (mountain height and width, initial wind velocity, initial moisture profile ...) with the results you obtained using the Kessler scheme. Do you see any differences and why?

5.2 Coupling of physics and dynamics

Up to now we neglected the vertical advection in the prognostic equations of u, σ, q_v, q_c, q_r, n_c and n_r, although condensational heating and evaporative cooling necessarily lead to a non-zero vertical velocity $\dot{\theta} \neq 0$. In this exercise we complete the model by coupling the diabatic forcing to the dynamics and account for vertical advection. The advection operator then reads

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\left(\frac{\partial}{\partial x}\right)_\theta + \dot{\theta}\frac{\partial}{\partial \theta}.$$

Thus, the tendency of any prognostic variable χ receives an additional term $\frac{\partial \chi}{\partial t} = \cdots - \dot{\theta}\frac{\partial \chi}{\partial \theta}$.

- Discretize the additional tendency term from vertical advection of any prognostic variable using centered finite differences and solve for χ^{n+1}. For the discretization you need to consider that $\dot{\theta}$ is staggered vertically and unstaggered in the horizontal. Thus, you need to interpolate $\dot{\theta}$ adequately to the grid points on which χ is defined. As a result, the interpolation of $\dot{\theta}$ must be different depending on which variable represents χ.
- The vertical velocity $\dot{\theta}$ has to be passed as an input argument to all the prognostic sub functions and to the two-moment microphysical scheme. Pass the additional input argument *dthetadt* as *dthetadt=dthetadt*.
  Within the sub functions, the tendency due to vertical advection should then only be evaluated if the additional argument *dthetadt* has been passed. Otherwise, only

horizontal advection should be calculated. To this end, add the vertical advection tendency in all the four prognostic sub functions to the previously calculated tendency from horizontal advection by the following principle:

```
unew[i,:] = uold[i,:]-dtdx*unow[i,:]...       # horizontal advection of u
if idthdt == 1:
    ii,kk = np.ix_(i,k)                       # np.ix_ makes accessing an array
                                              # with multiple 1D arrays (i,k) possible.
    unew[ii,kk] = unew[ii,kk]-dt/dth...       # Update of u^(n+1) by vertical advection
```

- Since our lower boundary condition is $\theta_s = const.$ we might run into trouble if $\dot{\theta}_s \neq 0$. Therefore, we force $\dot{\theta}_s$ explicitly to zero (already implemented) and similarly for the upper boundary θ_t. When adding the vertical advection tendency to the prognostic equations, remember not to allow for latent heat release at the surface and at the upper boundary, to stay consistent with the above assumption! In order to prevent instability near the ground, we should also prevent raindrops from evaporating near the ground. To do so, set *iern=0* in your *namelist.py*.
- Perform dynamically coupled simulations using the two-moment microphysical scheme. Do not forget to switch *idthdt* to 1. What is the consequence of the diabatic heat release on, e.g., surface precipitation?

Model Check 2

Model Check 2 should give you the possibility to check whether your model yields meaningful results after you finished implementing the microphysics. Compare your results with the plot for the cloud number density after 1 hour of simulation shown in Figure 1. The namelist settings to obtain the right plot can be found as *namelist_ex5.py* in OLAT. Produce a pdf with the plot and upload it to Olat at task ModelCheck2. **The deadline for handing in the Model check 2 pdf plot is January 9, 2026**.

If you get the same plot, you can proceed with the preparation of the final report/presentation. If not let me know and we can try to find differences in the code together. I will try to check your submission as soon as possible.

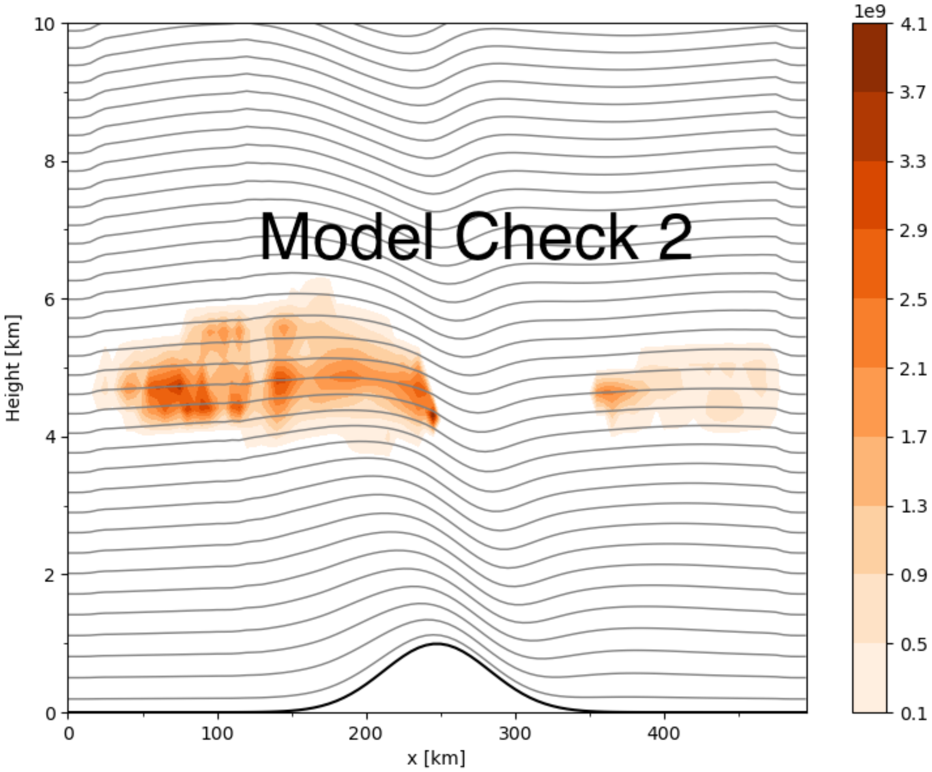

Figure 1: Model verification 2. Cloud number density created with namelist_ex5.py.

References

Seifert, A. and Beheng K. D., 2005: A microphysics two moment cloud parameterization for mixed-phase clouds. Part 1: Model description, Meteorol. Atmos. Phys. 92, 4566