Exercise 4:
Moist flow and microphysics - Part I

So far, the simulated atmosphere has been assumed to be dry. In this exercise, we will adapt our model for simulations of moist flows over the mountain ridge. To do so, additional budget equations have to be solved for the mixing ratios of water vapor, cloud liquid water, and rain droplets in Exercise 4.1.

Since lifting over the ridge will lead to condensation and subsequent formation of precipitation, we also have to consider and parameterize the formation of condensated water and rain droplets. This is achieved by a warm-rain bulk-microphysics scheme, which is formulated following Kessler (1969) and applied in Exercise 4.2.

Note that up to now we benefitted from the fact that we simulated an adiabatic flow ($\dot{\theta}$ = 0). Including condensational heating and evaporative cooling adds diabatic sources, such that the vertical velocity $\dot{\theta}$ is unequal zero. We will take care of this issue in Exercise 5.

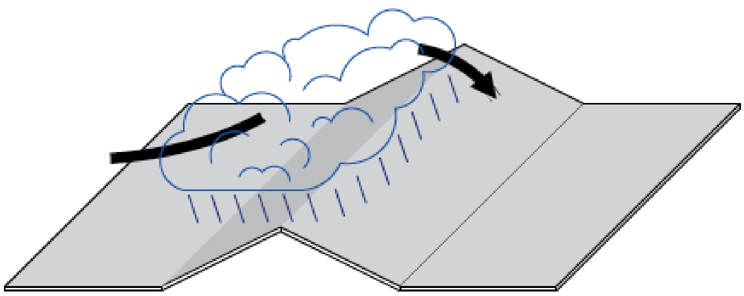

Figure 1: Orographic rain caused by a flow over a mountain ridge.

4.1 Passive advection of moisture scalars

We start by advecting the three new moisture scalars without any microphysical interactions among them. The moisture variables are co-located on the numerical grid at the same gridpoints as isentropic density σ. Then the budget equations for water vapor, cloud water, and rain droplets are written as

$$\frac{\partial q_v}{\partial t} + u\frac{\partial q_v}{\partial x} = 0$$
$$\frac{\partial q_c}{\partial t} + u\frac{\partial q_c}{\partial x} = 0$$
$$\frac{\partial q_r}{\partial t} + u\frac{\partial q_r}{\partial x} = 0$$

In this exercise we implement the horizontal advection:

- First of all, discretize these equations using centered finite differences both in space and time (as in Exercise 1) and solve for $q_{vi,k}^{n+1}, q_{ci,k}^{n+1}, and\ q_{ri,k}^{n+1}$. Note that the

horizontal wind speed *u* has to be interpolated to the non-staggered grid. Again, formulate the discretized equations in Python notation using integer indices only.
The new variables are already defined in the code, e.g., as *qvold*, *qvnow*, *qvnew* for water vapor.

- The effects of moisture should only be active for *imoist=1*, which can be switched on and off in *namelist.py*. Use if-statements for all following additions to ensure that all computations regarding moisture are only considered for *imoist=1*.
- The initial vertical profile is zero everywhere for q_c and q_r, but a non-zero profile has to be specified for water vapor q_v within the sub function *makesetup.py* at *Initial moisture profile*. Initialize the moisture profile over the vertical layers *k* in terms of relative humidity *rh0*, which is already defined. Use a humidity perturbation given as

$$rh0(k) = \begin{cases} rh_{max} cos^2\left(\frac{|k-k_c|}{k_w}\frac{\pi}{2}\right) & for\ k_c - k_w < k < k_c + k_w \\ 0 & elsewhere \end{cases}$$

  to set a smooth transition from *rh0*(k = 2) = 0 to *rh0*($k = k_c$ = 12) = rh_{max} = 0.98 and again decreasing to zero above. k_w = 10 is the half-width of the moist layer. Call the sub function *rrmixv1* to convert relative humidity *rh0(k)* to water vapor *qv0(k)*. The call should read as follows:

  ```
  qv0[k] = rrmixv1(0.5*(prs0[k]+prs0[k+1])/100,...
  0.5*(th0[k]/cp*exn0[k]+th0[k+1]/cp*exn0[k+1]),rh0[k],2)
  ```
- Add the call of the prognostic step for the new variables in *solver.py* at timestep for moisture scalars. The corresponding sub function is *prog_moisture* in *prognostics.py*, which needs to be completed with your discretized equations for $q_{vi,k}^{n+1}, q_{ci,k}^{n+1}, and\ q_{ri,k}^{n+1}$.
- Lateral boundary conditions (relaxation, periodicity) are already implemented.
- Exchange the variables for the next time step below the section where you already exchange the variables for velocity and isentropic density in *solver.py*.
- The moisture scalars are positive definite by nature. However, the dispersivity of the numerical scheme will result in unphysical negative values. Here, we simply set all negative values $q_{vi,k}^{n+1}, q_{ci,k}^{n+1}, and\ q_{ri,k}^{n+1}$ to zero, that is for water vapor
  qvnew[qvnew < 0] = 0
  and similar for cloud and rain water content. Add this at *Clipping of negative values* in *solver.py*.
- High values for specific humidity can lead to instabilities or require a low timestep. For all experiment involving moisture it is recommended to lower the temperature in the model. Set *th00* to 280 in the namelist.

Now you should test the advection. To this end set *imoist=1*, perform a first simulation and plot q_v.
Since, we did not couple the moisture scalars with the aid of a microphysics scheme yet, we can see q_v as a passive tracer, which is advected by the flow. Make the advection visible by

adding an x-dependent initial perturbation in q_v (e.g. a sin^2 function) in *makesetup.py* for *qvold* and *qvnow* and by advecting by one cycle using periodic lateral boundary conditions.

Hint: Use the functions *np.sin()*, *np.linspace()* and *np.pi* for the initial perturbation. In order to multiply a 2-D array *A* with the shape (*nx, nz*) with a 1-D array *b* with the shape (*nx*) for each level of *nz* you can use A * b[:,None].
Use flat topography for this test. What result do you expect? What is the effect of applying/not applying horizontal diffusion to q_v (*imoist_diff*=1/0)?

4.2 Kessler microphysics

As soon as saturation is achieved in a moist atmosphere, water vapor will start to condensate and a complex chain of exchange processes between vapor, cloud droplets, and precipitating droplets will set in. Usually, this process is framed by ice-, mixed-phase and warm-rain processes. The Kessler scheme only captures the latter.
First, the scheme comprises the condensation rate of vapor and the evaporation rate of cloud water (G). Second, the conversion rate of cloud water q_c to precipitation particles q_r by collision and coalescence is described by the so called autoconversion (CC). In addition, the accretion rate (AC) of small droplets by rain drops is incorporated. Third, the sedimentation of rain (Beard 1976) and the evaporation rate of rain (EP) to water vapor q_v is included. Thus, the budget equations from Exercise 4.1 now include sink/source terms:

$$\frac{\partial q_v}{\partial t} + u\frac{\partial q_v}{\partial x} = -G + EP$$
$$\frac{\partial q_c}{\partial t} + u\frac{\partial q_c}{\partial x} = G - CC - AC$$
$$\frac{\partial q_r}{\partial t} + u\frac{\partial q_r}{\partial x} = CC + AC - EP$$

These equations are already implemented in the sub function *kessler* within *microphysics.py*. This sub function should only be called for *imicrophys=1*.

- Call the Kessler parameterization scheme at *add call of kessler subfunction* in *solver.py*:
  [lheat,qvnew,qcnew,qrnew,prec,tot prec] = kessler(snew, qvnew, qcnew, qrnew, prs, exn, zhtnow, th0, prec, tot prec)
  Try to understand the arguments of this sub function. What are the input variables? What are the output ones? Try to identify the individual processes in *kessler*.
- Perform several sensitivity simulations (use relaxed boundary conditions) using the unperturbed moisture profile from Exercise 4.1:
  1. Decrease/increase the terminal fall velocity *vt* of rain drops by changing its multiplication factor *vt_mult* in namelist. How does this affect the spatial distribution of rainfall over the ridge? You may plot the accumulated

precipitation q_r with the argument -v *specific_rain_water_content* for the plotting function.

2. Switch the evaporative sink (ern) of rain drops on/off (the corresponding switch is named *iern*). How does this affect the accumulated amount of precipitation over the ridge?
3. Autoconversion does only set in, if the cloud water mixing ratio exceeds a certain threshold. Increase / decrease this threshold (*autoconv_th*). How is the precipitation formation affected, if you double the autoconversion rate (*autoconv_mult*)?

References

Kessler, E., 1969: On the distribution and continuity of water substance in atmospheric circulation, Volume 10, Meteor. Monogr., Amer. Meteor. Soc., 84 pp.

Beard K., 1976: Terminal Velocity and Shape of Cloud and Precipitation Drops Aloft. J. Atmos. Sci., 33, 851-864