

Documentation for LMDZ, Planets version

The $C_p(T)$ dependency

Sébastien Lebonnois

Latest version: May 25, 2020

1 Theoretical aspects

Taken from *Lebonnois et al.* (2010).

The standard version of the LMDZ dynamical core uses a single value for the specific heat C_p , but C_p varies in the atmosphere of Venus from 738 J/kg/K at 100 km altitude to 1181 J/kg/K near the surface (values taken from the Venus International Reference Atmosphere, *Seiff et al.*, 1985). This variation of C_p with temperature needs to be taken into account, in order to get realistic adiabatic lapse rates in the whole atmosphere. We use an analytic approximation for this temperature dependence, that yields values very close to the VIRA profile for C_p (within 4% for temperatures below 200 K, below 1% everywhere else):

$$C_p(T) = C_{p0} \times \left(\frac{T}{T_0} \right)^\nu, \quad (1)$$

with $C_{p0} = 1000$ J/kg/K, $T_0 = 460$ K, and $\nu = 0.35$.

In the LMDZ model, the potential temperature is one of the fundamental prognostic variable in the equations of energy conservation. The potential temperature θ is the temperature an air parcel initially at temperature T and pressure p would reach when brought to a reference pressure p_{ref} (typically, the surface pressure) adiabatically:

$$\int_\theta^T C_p \frac{dT}{T} = \int_{p_{\text{ref}}}^p R \frac{dp}{p} \quad (2)$$

(R is the atmospheric gas constant). Equation 2 is obtained from the first principle of thermodynamics when a parcel of air at pressure p and temperature T is brought adiabatically to the reference pressure p_{ref} .

When C_p is taken as constant with temperature, Eq. 2 yields the classical expression of potential temperature $\theta = T \times (p_{\text{ref}}/p)^\kappa$, with $\kappa = R/C_p$. However, when C_p depends on temperature, this expression is no longer valid.

To keep modifications of the dynamical core to the minimum, we have kept potential temperature as a key variable in the dynamics. Thus, we have calculated the new expression of potential temperature under these conditions. Introducing the expression of $C_p(T)$ (Eq. 1) in Eq. 2 yields

$$\int_\theta^T \frac{T^{\nu-1}}{T_0^\nu} dT = \frac{R}{C_{p0}} \ln \frac{p}{p_{\text{ref}}}, \quad (3)$$

then (for $\nu \neq 0$)

$$\frac{1}{\nu T_0^\nu} (T^\nu - \theta^\nu) = \ln \left(\frac{p}{p_{\text{ref}}} \right)^{\kappa_0}, \quad (4)$$

where $\kappa_0 = \frac{R}{C_{p0}}$. This yields the new expression for the potential temperature:

$$\theta^\nu = T^\nu + \nu T_0^\nu \ln \left(\frac{p_{\text{ref}}}{p} \right)^{\kappa_0}. \quad (5)$$

The adjustments done in the dynamical core enable us to run the GCM with any formulation of C_p and the corresponding potential temperature based on Eq. 2.

2 Pratical aspects in the code

A specific file has been added to the dynamical core, `dyn3d_common/cpdet_mod.F90`, which includes all the needed routines to take the $C_p(T)$ possibility into account. These routines take advantage of the boolean flag `cpofT` (declared in `gcm.def`, default is T for Venus, F otherwise) to implement different expressions of $C_p(T)$. It could also use the keyword `planet_type` if needed to distinguish different planets.

These routines are:

- `ini_cpdet`: initializes the parameters ν (`nu_venus`) and T_0 (`t0_venus`) to either the Venus values, or zero. It is called just once at the beginning of the main routine. These parameters are declared in `comconst_mod.F90`.
- `cpdet`: function, that computes C_p for a given T . For other planets than Venus, it is just `cpdet(T)=cpp`.
- `t2tpot`: converts a temperature vector to a potential temperature vector. Serial `t2tpot` and parallel `t2tpot_p` versions.
- `tpot2t`: converts a potential temperature vector to a temperature vector. Serial `tpot2t` and parallel `tpot2t_p` versions.

In the physics, these routines are also defined `phyvenus/cpdet_phy_mod.F90`. They are initialized with `init_cpdet_phy` that is called from the dynamics/physics interface, passing the `cpp`, `t0_venus` and `nu_venus` variables. The value of `t0_venus` (0 or not) is used to determine the expression of $C_p(T)$.

For Venus, in the dynamics/physics interface, the routine `suphec(cpp)` needed to be called to initialize `RCPD=cpp` in the physics. In the routines, instead of using the constants `cpp` (dynamical core) or `RCPD` (physical module), we need to use the function `cpdet(T)` where T is the temperature at a given point.

Since the potential temperature is the variable used in the dynamical core, the temperature is scarcely used. But in some places, it was computed directly from the potential temperature and the Exner function $pk = cpp \times (\frac{p_{lay}}{p_{pref}})^{\kappa}$. The routine `tpot2t` is now used to compute the temperature when needed, and the variable `tsurpk`, which is the temperature over the Exner function can be used as an argument instead of the former potential temperature (for `sortvarc0.F`, `sortvarc.F`, `dudv2.F` and `geopot.F`). This affects the routines (in `dyn3d`):

- `caldyn0.F`
- `caldyn.F`
- `calfis.F`
- `diagedyn.F`
- `leapfrog.F`
- `vlspltqs.F`

References

Lebonnois, S., F. Hourdin, V. Eymet, A. Crespin, R. Fournier, and F. Forget (2010), Superrotation of Venus' atmosphere analysed with a full General Circulation Model, *J. Geophys. Res.*, 115, E06006, doi:10.1029/2009JE003458.

Seiff, A., J. T. Schofield, A. J. Kliore, et al. (1985), Model of the structure of the atmosphere of Venus from surface to 100 km altitude, *Adv. Space Res.*, 5(11), 3–58.