

CORDEX-WRF module: version 2

Lluís Fita

Centro de Investigaciones del Mar y la Atmósfera (CIMA), IRL IFAECI
UBA-CNICET-CNRS-IRD, C. A. Buenos Aires, Argentina

April 11, 2024

Contents

1	Introduction	2
2	Corrected variables	2
2.1	clwvi	2
2.2	Convection indices	2
2.3	rlusgen	2
3	New Variables	6
3.1	zmla: Boundary layer height	6
3.1.1	Nielsen-Gammon08	6
3.1.2	Bulk zmla	6
3.2	0-isotherm	7
3.3	ws50m, u50m, v50m	7
3.3.1	ws50m, u50m, v50m	7
3.3.2	ta50m, hus50m	11
3.4	iutmean, ivtmean	16
3.5	tas_hm, hurs_hm	16
3.6	Implementation in the module	16
3.7	twc	20
3.8	qc_pl, qr_pl, qs_pl, qi_pl, qg_pl, qh_pl	20
3.9	fixing convection diagnostics	20
3.10	x_pl_mc	23
3.11	simultaneous residence time	28
4	WRF time-memory sensitivity to CDXWRF configuration	30
A	Linear interpolation	35
B	2-m iterative method	37
B.1	Hess 95 methodology	37
B.2	LMDZ implementation	38
B.2.1	cdrag_mod.F90	39
B.2.2	screenp_mod	46
B.2.3	screenc_mod.F90	46

Table 1: Locations used to show results

Location number	Name	latitude	longitude	description
0018	Humauaca	-23.485	-65.878	High altitude station within dry and hot climate
0012	Formosa Aero.	-26.120	-58.140	low altitude station within wet and warm climate
0002	Aeroparque Aero.	-34.567	-58.417	low altitude coastal station within urban environment
0005	Bariloche Aero.	-41.090	-71.100	alpine station
0006	Base Marambio	-64.140	-56.400	Antarctic station

1 Introduction

There are some variables in the CORDEX list, that might require an agreement within the community in how to compute / infer / diagnose them. With this in mind, and following the previous work published in (*‘Geoscientific Model Development (GMD)’ Fita et al., 2019, hereafter Fita19*), here is presented a new version of the module which include corrections to certain variables from the previous version and new variables.

In order to present the results of the new implementations, a simulation was performed over a single domain at 25 km horizontal resolution for the entire month of November 2018 over Argentinean territories and temporal evolutions at different locations (see table 1 for description of the stations). All the results (except CAPE analysis) here presented are based in this simulation. See figure 14 for more details about the model set-up.

2 Corrected variables

In this version some corrections have been made into the module in order to fix different errors found in Fita19

2.1 clwvi

The condensed water path has to be fixed. Thanks to Jesús Fernández (IFCA, Spain) in agreement to the CORDEX variables list comment ([CORDEX-CMIP6_atm_variable_list_WRF.csv#L51](#)) the vertical integration has to include also some of the solid species and now becomes as it is shown in equation 1

$$clwvi = \frac{MU + MUB}{g} \sum_{k=1}^{dz} [QVAPOR(k) + QCLOUD(k) + QICE(k) + SNOW(k)] DNW(k) \quad (1)$$

2.2 Convection indices

CAPE, CIN, ZLFC, PLFC and LI convection activity related indices from AFWA module were kept from WRF’s 3.7.1 version. It had some errors which where fixed by Zhixiao Zhang (University of Oxford, UK) and taken now, from WRF version 4.3.1.

Victoria Oruezabal (CIMA, Argentina) used previous and new version as post-processed diagnostic and the impacts are even large when using ERA5 data (figure 1).

2.3 rlusgen

Thanks to Jatin Kala (Murdoch University, Australia), the surface upwelling longwave radiation formulation is not completed. Accordingly to ([notes](#), equation 27) it lacks the the reflection by clouds of the incident long-wave radiation term (for WRF (1-EMISS)GLW).

This is a generic definition of the rlus, so it must be independent of the radiative scheme being used. If I recall properly, not all the radiative schemes in WRF provide the GLW, so we removed this term from the equation, otherwise

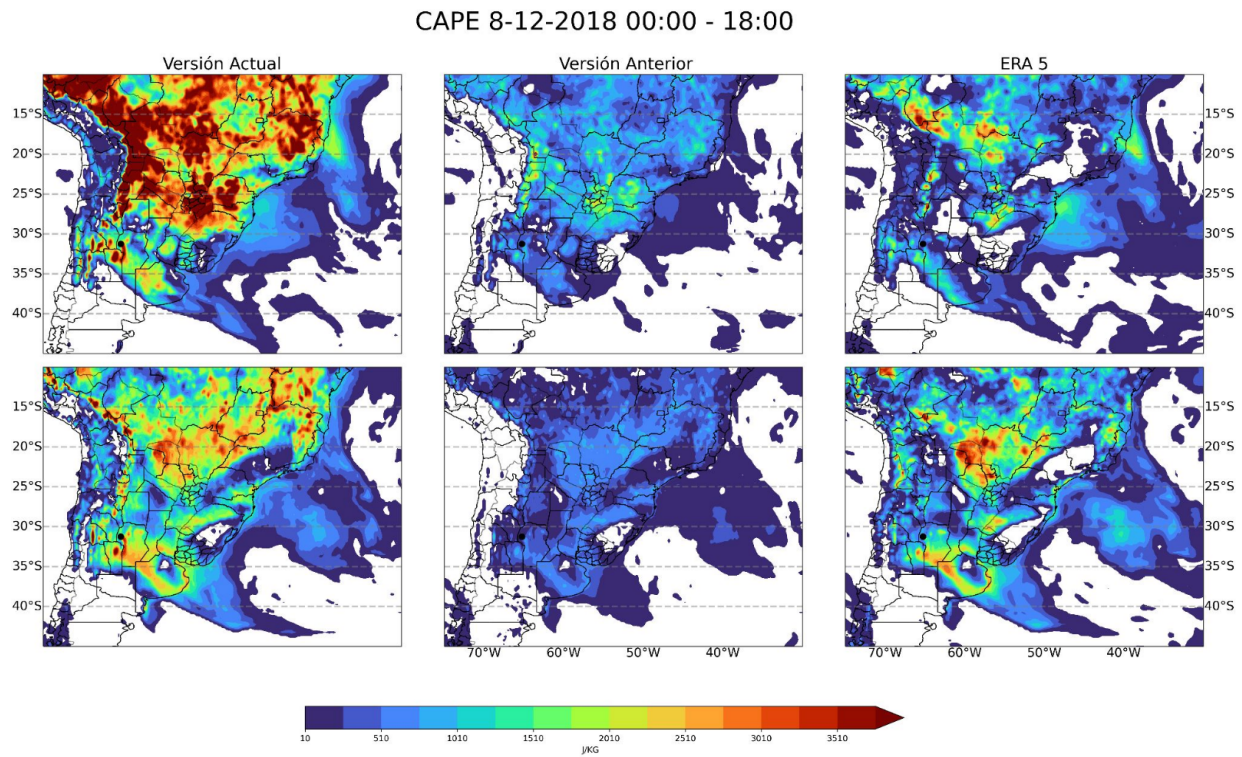


Figure 1: Diagnosed CAPE using ERA5 data and the new version included in CDXWRF v2 (left column), previous version CDXWRF v1.3 (middle col.) and ERA5 diagnostic (right col.) on 2018 December the 8th at 00 (top row) and 18 UTC (bottom row). Figure cutesy of Victoria Oruezabal (CIMA, Argentina)

it would be scheme dependent. For some schemes we already have the upwelling long-wave radiation (e.g. ACLWUPB), so there was no need to re-compute it.

In order to show the impact of this term, the figure 2 shows the values of each term.

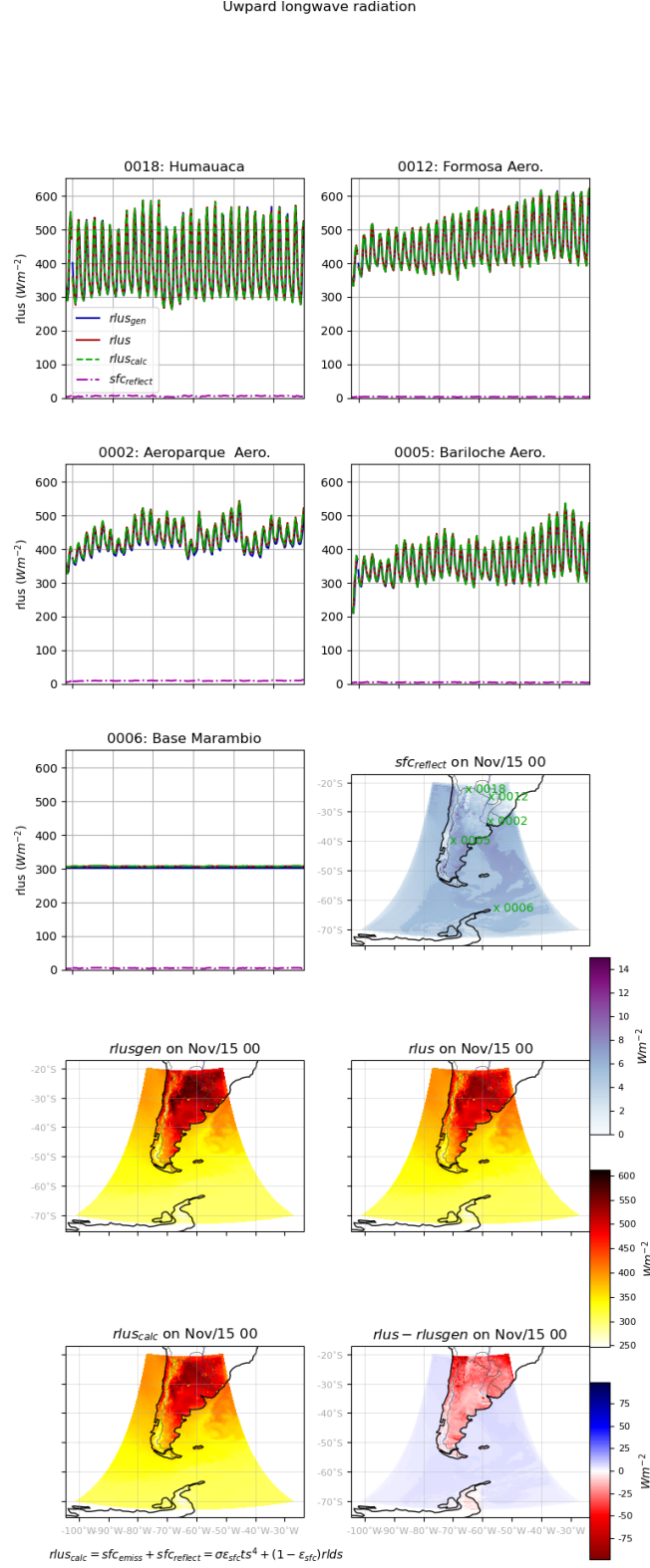


Figure 2: Entire month evolution of the long-wave radiation terms at the surface at 5 different Argentinean locations (see table 1): *Aeroparque airport* (0002 station, top left), *Antartica Marambio base* (0006 station, top right), *Formosa airport* (0012 station, bottom left) and *Humauaca* (0018 station, bottom right) and the different components on 2018 November the 15th at 00 UTC. r_{lus_gen} : the generic definition from CDXWRF, r_{lus} : WRF's variable LWUPB, $r_{lus_calc} = sfc_{emiss} + sfc_{reflect}$: full calculation of r_{lus} using WRF's variables where $sfc_{emiss} = \sigma \epsilon_{sfc} T_s^4 = Ct_{Bolz} * EMISS * TSK^{**4}$: surface black-body emission and $sfc_{reflect} = 1 - \epsilon_{sfc} r_{lds} = (1. - EMISS) * GLW$: surface reflection of long-wave radiation

3 New Variables

Some of the new variables are needed for the new CORDEX-list of variables, some others are included as seen as potentially useful for some communities.

3.1 zmla: Boundary layer height

This variable is provided as a diagnostics by almost all the planetary boundary layer (pbl) schemes implemented in the atmospheric models. However, because these schemes are constructed over different assumptions and set of equations. Therefore it makes not possible to directly compare values among models (or even within the same model, using different pbl schemes).

In the first version (Fita19), a unique methodology was provided. In this update, a new one is introduced. At the same time, due to the existence of 2 different methodologies, a new flag called 'zmlagen_diag' is introduced, with which user select the preferred method of diagnose the generic boundary layer height, being:

- `zmlage_diag = 1`: vertical, theta-e based methodology from (Nielsen-Gammon et al., 2008) as it was in previous version of the module
- `zmlage_diag = 2`: bulk methodology using the Richardson number after (Vogelezang and Holtslag, 1996; Seidel et al., 2004) [default value]

See results in figure 3.

3.1.1 Nielsen-Gammon08

After (Nielsen-Gammon et al., 2008). The method consists in defining the height of the PBL as the first level in the mixed layer (ML) where potential temperature exceeds the minimum potential ML temperature by more than 1.5 K. It has been implemented using the definitions given below:

1. Mixed layer depth (MLD) is defined as the model level (k_{MLD}) starting from the second model level at which the variation of the mixing ratio ($qv(k)$, normalized with its value at the first level) exceeds some predefined threshold value ($qv(1)$): $\frac{|qv(k_{MLD}) - qv(1)|}{qv(1)} > \delta qv$ (here applied a $\delta qv = 0.1$)
2. Within the MLD the value with the minimum potential temperature is taken as: $\theta_{min_{MLD}} = \min[\theta(1), \dots, \theta(k_{MLD})]$
3. The level of the pbl height (k_{zmla}) is the level at which the maximum variation of potential temperature within the MLD exceeds some predefined threshold value: $\theta(k_{zmla}) - \theta_{min_{MLD}} > \delta\theta$, (here $\delta\theta = 1.5 K$)
4. The pbl height ($zmla$) is obtained using the geopotential height zg at the calculated k_{zmla} level above the ground ($zagl$): $zmla = zagl(k_{zmla}) = zg(k_{zmla})/g - HGT$, with HGT being surface elevation height above sea level.

No general rule has been applied to determine the correct value of δqv used to determine MLD. It can be determined by the namelist parameters `zmlagen_dqv` for δqv (default value 0.1) and `zmlagen_dtheta` for $\delta\theta$ (default value 1.5 K). When comparing this implementation with the $zmla$ directly provided by WRF's Mellor-Yamada Nakanishi and Niino Level 2.5 PBL scheme (MYNN2.5 Nakanishi and Niino, 2006) in WRF model (shown in figure 9 in Fita19). In general the generic estimation produces a higher pbl (a panel) with lower values during night (b panel). Spatial distributions between both diagnostics are pretty similar.

3.1.2 Bulk zmla

After (Vogelezang and Holtslag, 1996), $zmla$ is computed using the bulk Richardson number. A threshold ($\mathcal{R}i_{thres} = 0.25$) is defined in order to determine the height of the pbl (see equation 2).

$$zmla = \begin{cases} \mathcal{R}i[z] \leq \mathcal{R}i_{thres} < \mathcal{R}i[z+1] \\ or \mathcal{R}i[z] > \mathcal{R}i_{thres} > \mathcal{R}i[z+1] \end{cases} \quad (2)$$

Using the bulk-Richardson number as defined in the equation 3

$$\mathcal{R}i = \frac{g/T_v \Delta\theta_v \Delta z}{(\Delta u)^2 + (\Delta v)^2 + 100(u^*)^2} \quad (3)$$

where Bulk Richardson number follows AMS Glossary [definition](#) adding the $(u^*)^2$ term as a way to take into account turbulence due to surface friction on neutral boundary layer (eq. 3 in [Vogelezang and Holtslag, 1996](#)). g is gravitational acceleration, T_v , absolute virtual temperature, $\Delta\theta_v$, virtual potential temperature difference across a layer of thickness Δz , Δu and Δv are the changes in horizontal wind components across that same layer.

3.2 0-isotherm

Diagnose of the first height from the ground where air-temperature becomes 0 (see results in [figure 4](#)).

3.3 ws50m, u50m, v50m

Diagnose at 50 m of wind, temperature and humidity has been introduced in this update. User can set-up the desired height with the namelist parameter `z50m`. Its default value is 50. These diagnostics are directly activated when compiling with the flag `-DCORDEXDIAG`.

5 new variables have been introduced:

- WSZ50: 50m wind speed, ms^{-1}
- UZ50: Eastward 50 m wind speed, ms^{-1}
- VZ50: Northward 50 m wind speed, ms^{-1}
- TAZ50: 50 m air temperature, K
- QVZ50: 50 m water vapour, $kgkg^{-1}$

3.3.1 ws50m, u50m, v50m

Following Monin-Obukhov Similarity theory (see [Box 2](#) for remarks) a diagnose to extrapolate winds (for the `ua100m`, `va100m` diagnostic) for winds below 80 m was introduced in the module. The wind at given height is extrapolated following turbulent mechanisms. As it is shown in [equation 4](#), surface wind speed is used as surrogate to estimate 100 m wind direction ($\theta_{10} = \tan^{-1}(uas, vas)$), without considering Eckman pumping, or other effects on wind direction). In this implementation u_* in similarity theory is taken as model estimates of UST, Monin-Obukhov length (\mathcal{L}_O), roughness length (z_0) and thermal time-varying roughness length ZNT.

$$\begin{aligned} wss_{100} &= \frac{UST}{\kappa} \left(\ln \left(\frac{100}{z_0} \right) + \Psi_M \left(\frac{100}{\mathcal{L}_O} \right) \right) \\ \mathcal{L}_O &= \frac{-UST^3 T_v}{\kappa g Q_0} \quad (Obukhov \text{ length}) \\ \Psi_M \left(\frac{z}{\mathcal{L}_O} \right) &= \begin{cases} \frac{4.7z}{\mathcal{L}_O} \left[\ln \left[\left(\frac{1+X^2}{2} \right) \left(\frac{1+X}{2} \right)^2 \right] - 2 \tan^{-1}(X) + \frac{\pi}{2} \right] & \frac{z}{\mathcal{L}_O} > 0 \quad (sTable) \\ \frac{z}{\mathcal{L}_O} < 0 & (unsTable) \end{cases} \\ X &= \left(1 - \frac{15z}{\mathcal{L}_O} \right)^{1/4} \\ \theta_{10} = atan \left(\frac{V10}{U10} \right) \vec{va}_{100} &= \begin{cases} ua_{100} = wss_{100} \cos(\theta_{10}) \\ va_{100} = wss_{100} \sin(\theta_{10}) \end{cases} \end{aligned} \quad (4)$$

wss_{100} : wind speed at 100 m (ms^{-1}), where Ψ_M : stability function after ([Businger et al., 1971](#)), UST : u^* in similarity theory (ms^{-1}), z_0 : roughness length (m), $U10, V10$: 10-m wind speed, θ_{10} : 10-m wind speed direction (rad), ua_{100} : 100 m eastward wind speed, va_{100} : 100 m northward wind speed (note the absence of correction in wind direction to Ekman pumping or other turbulence effects)

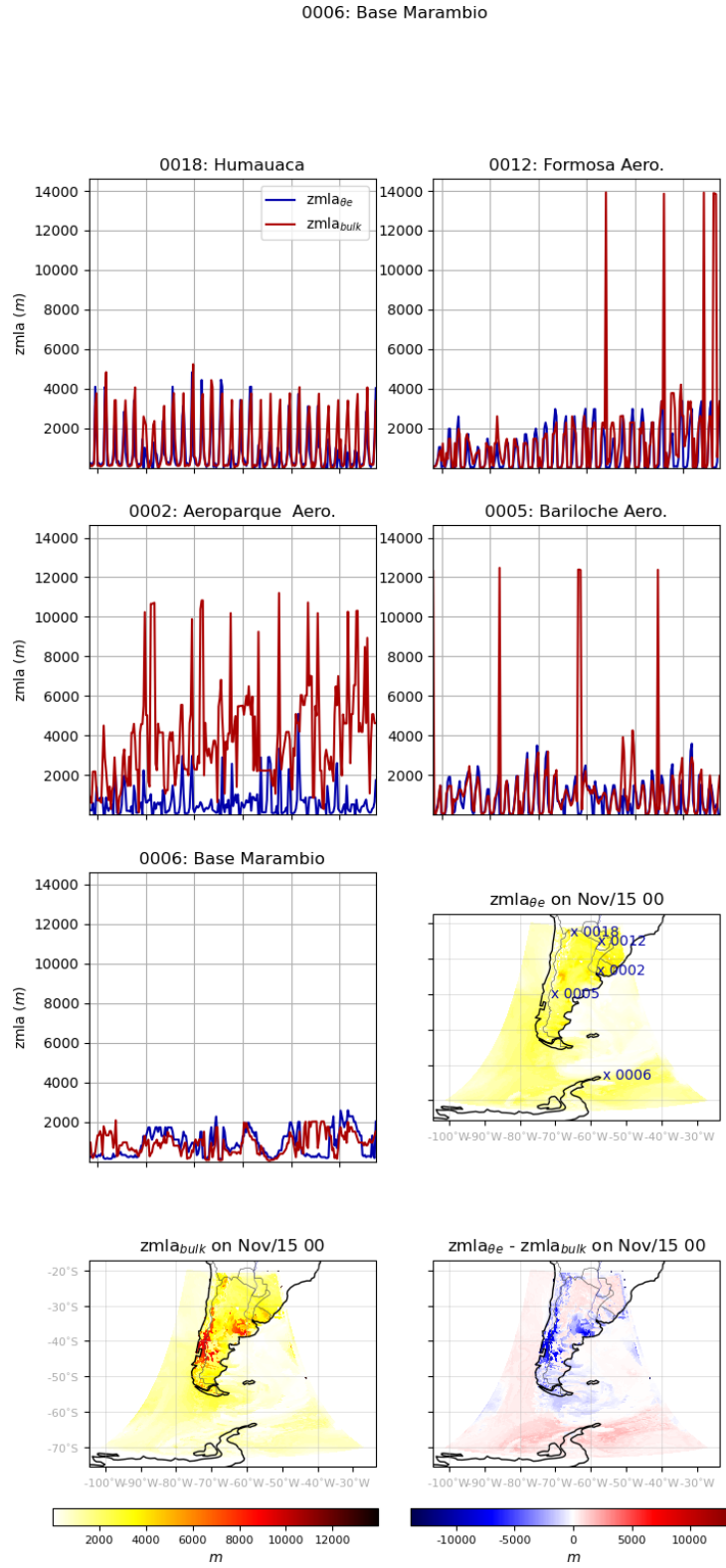


Figure 3: Evolution of generic boundary layer (zmla) by the two methods (theta-e, blue; bulk-Richardson, red) at different stations and on the 15th of November at 00 UTC for all the domain, for the diagnostics and the differences between them

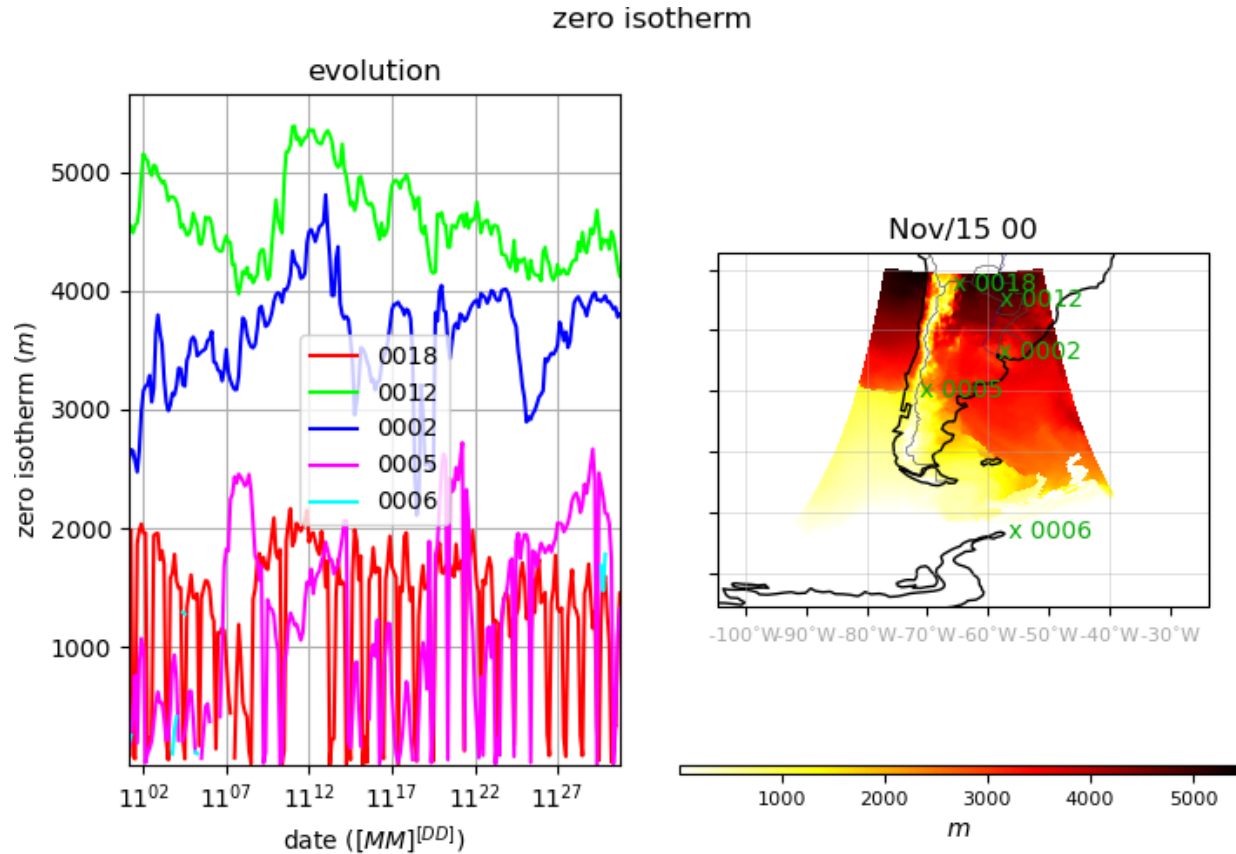


Figure 4: Evolution of zero isotherm ($z|_{ta=0^{\circ}C}$) at different stations and on the 15th of November at 00 UTC for all the domain

Figure 6 from Fita19 shows different preliminary results using the three different approximations for wind gust estimation. It is illustrated (a panel) how wind-gusts are larger than the 10-m diagnostic winds, and also the difference is larger when using Monin-Obukhov method compared to the two others methods. Certain problems (too small Monin-Obukhov length) are recognized when applying Monin-Obukhov for extrapolating wind at 100 m, which is shown in panel b, where wind gusts appear to be strong as 80 ms^{-1} . Therefore user is advised to use this method with care.

Here one should notice that some of the pbl-related variables (UST , z_0 , $U10$, $V10$, $theta_{10}$, ...) will be model / scheme dependent. See results in figure 5.

Box 1: Limitations of Monin-Obukhov Similarity Theory

The implemented equations are the ones from [Businger et al. \(1971\)](#). These equations only apply within certain ranges of the fraction z/\mathcal{L}_0 , being z : height and \mathcal{L}_0 : Monin-Obukhov length. The turbulent assumptions from which, one can vertically extrapolate scalars, must be within the turbulence-driven layer of the atmosphere, which happens to be (accordingly to Bussinger) within the range ($z/\mathcal{L}_0 > -2.$, $z/\mathcal{L}_0 < 1.$).

According to this limitation, diagnose uses models' 3-dimensional zg, ua, va, ta, qv to linearly extrapolate (see appendix A) ua_{50} , va_{50m} , ta_{50m} and hus_{50m} for the grid points that 50 m above ground lays outside the TKE-driven turbulence layer

NOTE: Thanks to Prof. dr hab. inż. Mariusz J. Figurski and Adam Jaczewski from the *Centrum Modelowania Meteorologicznego* (Poland) for their comments, which facilitate the correction of a series of problems with these diagnostics

3.3.2 ta_{50m} , hus_{50m}

To diagnose temperature and humidity at relatively close to the surface ($z < 80$ m), is suggested to use the Monin-Obukhov Similarity Theory (see remarks of box 2), but adapting the equations to these variables.

One should be aware, that in order to proceed, we need to obtain the slope of the vertical profile of the desired variable: u^* for wind, ta^* for temperature and qv^* for humidity and that at the surface whereas $u_0 = 0$, for $ta_0, qv_0 \neq 0$. Models tend to provide u^* , but not ta^*, qv^* , therefore, it would be necessary to perform a two step methodology in order to get the necessary values, needed to estimate ta and qv at a desired height z_d (here we present the example for $ta(z_d)$):

1. Compute the increment (e.g. $\Delta ta' = ta(k+1) - ta(k)$) of the variable respect a known heights (at vertical levels k and $k+1$) from the 3-dimensional atmospheric temperature field
2. Infer ta^*
3. Compute the increment between one of the 3-dimensional temperatures and the desired height using ta^* (e.g. $\Delta ta'' = ta(k) - ta(z_d)$), to obtain ta_0
4. Obtain $ta(z_d)$

The general equation for the vertical profile of temperature and humidity at the surface layer of the PBL follows the equation 5

$$\bar{\chi}(z) - \bar{\chi}_0 = \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z}{z_0 \chi} \right) - \Psi_{\Xi} \left(\frac{z}{\mathcal{L}_O} \right) \right] \quad (5)$$

where $\chi = ta, qv$, for temperature and humidity and in this case will be the instantaneous values of the variable at the grid point during the integration of the model (T (previously converted to actual temperature) and QVAPOR), Ψ_{Ξ} stability function, with dependency on the stability and with different options for the profiles to use, here for completeness, Businger profiles will be used. Here, the formulations for scalars will be used (ta and qv will be treated as scalars fields). $z_0 \chi$ can be replaced by $0.1z_0$ (roughness length) without introducing too much error ($z_0 ta = z_0 qv = 0.1z_0$).

The stability functions for temperature and humidity (being the same) are shown in equation 6

$$\Psi_{ta,qv} \begin{cases} 1 - \phi_{ta,qv} \left(\frac{z}{\mathcal{L}_O} \right) & \frac{z}{\mathcal{L}_O} \geq 0 \\ \ln \left[\left(\frac{1+\xi}{2} \right)^2 \right] & \frac{z}{\mathcal{L}_O} < 0 \end{cases} \quad (6)$$

where $\xi = \phi_{ta,qv}^{-1}$ and $\phi_{ta,qv}$ are the universal functions.

There are different universal functions depending on the works and observations used. Here the ones closer to the same used for the winds will be used, which will correspond to the ones purposed by ([Businger et al., 1971](#), expressed in equation 7).

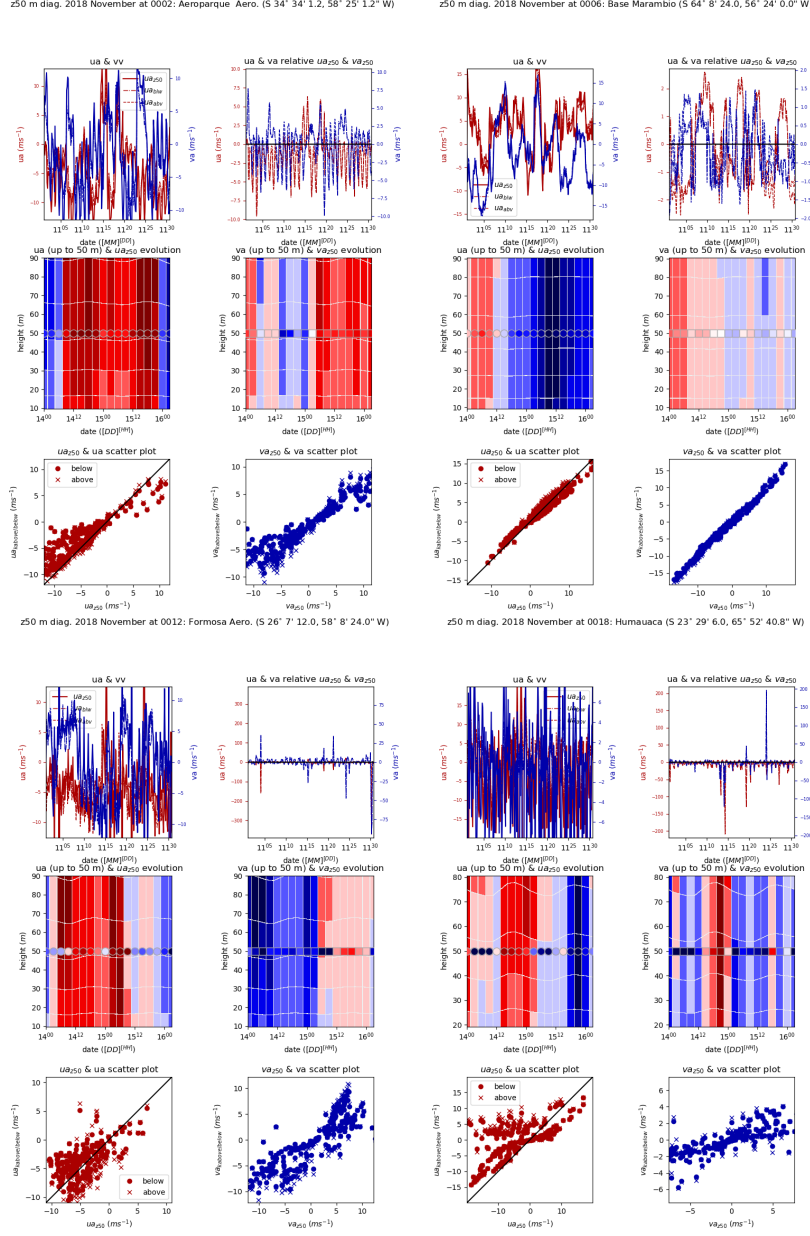


Figure 5: Comparison of air eastward wind speed (ua_{z50}) and northward wind speed (va_{z50}) diagnostics at 50 m following Monin-Obukhov Similarity Theory and vertically closest atmospheric values (below, above 50 m) for a WRF simulation on November 2018 at 5 different Argentinean locations. Top left panel: monthly time-series air eastward wind speed (ua , red, left y-axis), northward wind speed (va , blue, right y-axis). Top right panel: values at the vertical level above/below 50 m respect the diagnostic value. Middle panels: vertical profile evolution around 50 m with diagnostics at 50 m as colored markers, and vertical WRF values as shading. Bottom left panel: Scatter diagram for diagnosed eastward wind speed (x-axis) respect atmospheric values below/above 50 m (y-axis). Bottom right panel: Scatter diagram for diagnosed northward wind speed (x-axis) respect atmospheric values below/above 50 m (y-axis). Different scales and ranges are used in each figure

$$\kappa = 0.35; \phi_{ta,qv} = \begin{cases} 0.74(1 - 9\frac{z}{\mathcal{L}_O})^{-1/2} & -2 < \frac{z}{\mathcal{L}_O} < 0 \text{ (unstable)} \\ 0.74 & \mathcal{L}_O = 0 \text{ (neutral)} \\ 0.74 + 4.7\frac{z}{\mathcal{L}_O} & 0 < \frac{z}{\mathcal{L}_O} < 1 \text{ (stable)} \end{cases} \quad (7)$$

To obtain χ^* , one derives it from the $\Delta\bar{\chi}$ equation 8

$$\begin{aligned} \Delta\bar{\chi} &= (\bar{\chi}(k+1) + \bar{\chi}_0) - (\bar{\chi}(k) + \bar{\chi}_0) = \bar{\chi}(k+1) - \bar{\chi}(k) = \\ &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k+1)}{z_0} \right) - \Psi_{\Xi} \left(\frac{z(k+1)}{\mathcal{L}_O} \right) \right] - \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k)}{z_0} \right) - \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) \right] \\ &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k+1)}{z_0} \right) - \ln \left(\frac{z(k)}{z_0} \right) - \Psi_{\Xi} \left(\frac{z(k+1)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) \right] \\ &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k+1)}{z_0} \frac{z_0}{z(k)} \right) - \Psi_{\Xi} \left(\frac{z(k+1)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) \right] \\ &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k+1)}{z(k)} \right) - \Psi_{\Xi} \left(\frac{z(k+1)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) \right] \end{aligned} \quad (8)$$

Using known heights $z(k+1), z(k)$ and the values of $\bar{\chi}(k), \bar{\chi}(k+1)$ from the model levels, one can reduce χ^* as it is shown in equation 9

$$\chi^* = \kappa \frac{\bar{\chi}(k+1) - \bar{\chi}(k)}{\ln \left(\frac{z(k+1)}{z(k)} \right) - \Psi_{\Xi} \left(\frac{z(k+1)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right)} \quad (9)$$

Using again the $\Delta\bar{\chi}$ for the desired height z_d and with the recently found χ^* , one can obtain the actual value of $\bar{\chi}(z_d)$ (and also retrieve χ_0) as it is shown in equation 10

$$\begin{aligned} \Delta\bar{\chi} &= \bar{\chi}(k) - \bar{\chi}(z_d) \\ &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k)}{z_d} \right) - \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z_d}{\mathcal{L}_O} \right) \right] \\ \bar{\chi}(z_d) &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z(k)}{z_d} \right) - \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z_d}{\mathcal{L}_O} \right) \right] + \bar{\chi}(k) \end{aligned} \quad (10)$$

And for the case where $z_d > z(k)$ in equation 11

$$\begin{aligned} \Delta\bar{\chi} &= \bar{\chi}(z_d) - \bar{\chi}(k) \\ \bar{\chi}(z_d) &= \frac{\chi^*}{\kappa} \left[\ln \left(\frac{z_d}{z(k)} \right) - \Psi_{\Xi} \left(\frac{z_d}{\mathcal{L}_O} \right) + \Psi_{\Xi} \left(\frac{z(k)}{\mathcal{L}_O} \right) \right] + \bar{\chi}(k) \end{aligned} \quad (11)$$

Box 2: IMPORTANT REMARK about the Monin-Obukhov based diagnostics

Monin-Obukhov theory is used as an approximation to the estimation of the values of variables within the ‘surface layer’ of the *Planetary Boundary Layer* (PBL). One must be aware, that Monin-Obukhov theory and its associated vertical profiles are designed to be universal depictions of the dynamics of chaos based on *temporal-means*. Results should be taken with caution when used in atmospheric models to diagnose certain values.

- Used only within the ‘surface layer’ of the planetary boundary layer. This surface layer, is not universal and it will depend on a large number of conditions such as: stability, surface characteristics, existing canopy, among others. There is not an actual way to determine the end of the end of the layer. Bearing this in mind, under ‘normal’ conditions, should not be used above 80 m (maybe far less in stable nights)
- Profiles of the theory are not providing ‘instantaneous’ descriptions of the actual conditions of the atmosphere. These profiles although attempt to be ‘universal’, strongly rely on the observations used to extract them. Also, they are used as temporal mean representations of the chaotic behavior of the atmosphere. These mean representations are averages computed for high frequency measurements (order of Hz) for time-windows of 10 to 30 minutes.
- Due to the lack of proper observations of humidity, the vertical profiles of the field are similar to the ones from the temperature, by using the profiles for scalar variables.
- Therefore, diagnostics based on Monin-Obukhov theory (or similar), should take with caution. Since, from the modeling perspective:
 1. They are used to provide instantaneous values
 2. Distinction among ‘ $PBL_{surface\ layer} - PBL - free\ atmosphere$ ’ is strongly dependent of models
 3. In some cases, there is no information about height of the canopy of the grid cell, and how PBL is interpreted in the areas within / outside the canopy
- There might be vertical profiles for layers above the surface layer and still within the PBL, but these profiles would not follow Monin-Obukhov theory.
- In the cases, where desired heights are outside the PBL, the values of the variables at the desired heights could be directly interpolated from the closest pair of vertical layers of the atmosphere
- There are some vertical profiles for the entire PBL (see for example [Ulke, 2000](#)), but they will not be considered (yet)

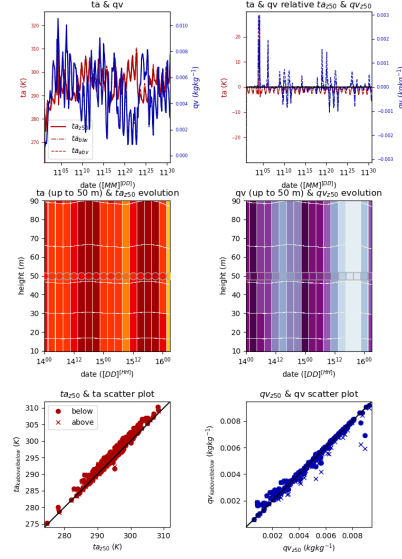
acknowledgments: Thanks to Dr. Marisa Gassman and Natalia Tonti (DCAO, UBA, Argentina) for their suggestions, comments and explanations

In order to avoid non consistent values (see box 1), diagnosed values $\bar{\chi}(z_d)$ are constrained within the values at model’s first layer values $ta|_{k=1}$, $qv|_{k=1}$ and the second one $ta|_{k=2}$, $qv|_{k=2}$. In that cases, the value at 50 m is given as the closest one ($k_{closest}$), following the equations at 12

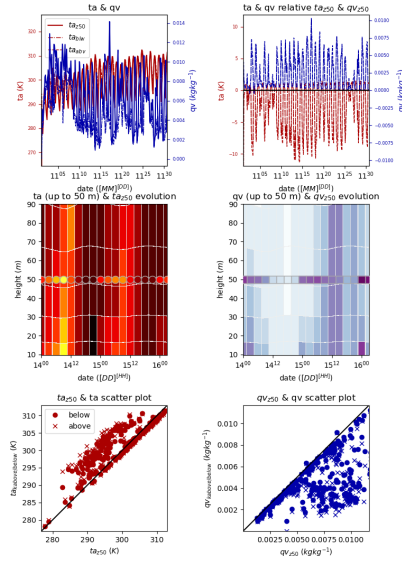
$$\begin{aligned}
 & \left\{ \begin{array}{l} tamin = \min(ta|_{k=1}, ta|_{k=2}) \\ qvmin = \min(qv|_{k=1}, qv|_{k=2}) \\ tamax = \max(ta|_{k=1}, ta|_{k=2}) \\ qvmax = \max(qv|_{k=1}, qv|_{k=2}) \\ k_{closest} \rightarrow \min(abs(z|_{k=1} - 50.), abs(z|_{k=2} - 50.)) \end{array} \right. \\
 ta50m & < tamin \rightarrow ta50m = ta|_{k=k_{closest}} \\
 hus50m & < qvmin \rightarrow hus50m = qv|_{k=k_{closest}} \\
 ta50m & > tamax \rightarrow ta50m = ta|_{k=k_{closest}} \\
 hus50m & > qvmax \rightarrow hus50m = qv|_{k=k_{closest}}
 \end{aligned} \tag{12}$$

See results in figures 6 and 5

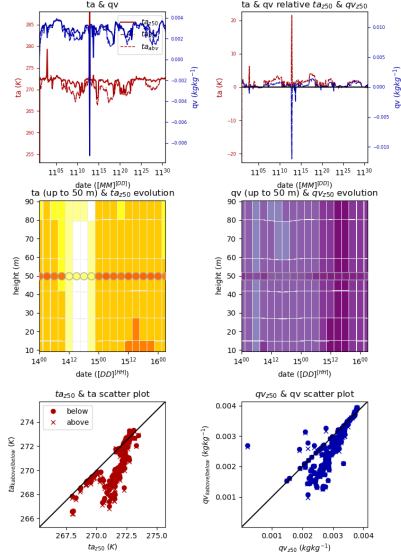
z50 m diag. 2018 November at 0002: Aeroparque Aero. (S 34° 34' 1.2, 58° 25' 12" W)



z50 m diag. 2018 November at 0012: Formosa Aero. (S 26° 7' 12.0, 58° 8' 24.0" W)



z50 m diag. 2018 November at 0006: Base Marambio (S 64° 8' 24.0, 56° 24' 0.0" W)



z50 m diag. 2018 November at 0018: Humauaca (S 23° 29' 6.0, 65° 52' 40.8" W)

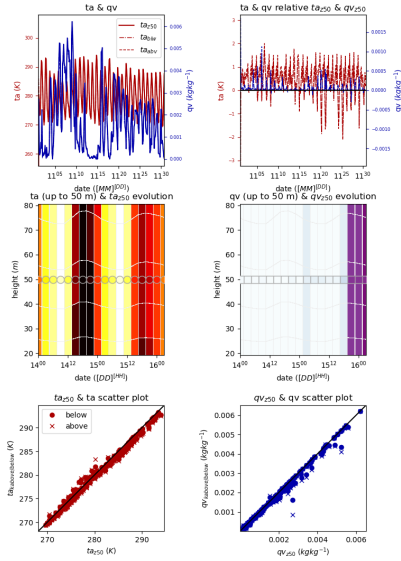


Figure 6: As in 5, but for wind components ta (air temperature) and qv (water vapour mixing ratio)

3.4 iutmean, ivtmean

Vertically integrated horizontal transport of water vapour is a common diagnostic in climate studies. Here is introduced as the mean value between output time-steps (adding the individual amounts during model integration time-steps) as it is shown in equation 13 and figure 7. This variable is activated with compilation flag `-DCDXWRF=1`

$$i\chi t = \frac{MU + MUB}{g} \sum_{iz=1}^{e_{vert}} \chi * QVAPOR[iz](DNW[iz]) \quad (13)$$

where $\chi = u, v$ being each of the wind components.

Two new variables will appear at the `wrfcdx` (`auxhis9`) output file:

- IUTMEAN: mean vertically integrated eastward transport of water vapour, $kgm^{-1}s^{-1}$
- IVTMEAN: mean vertically integrated northward transport of water vapour, $kgm^{-1}s^{-1}$

3.5 tas_hm, hurs_hm

The diagnostics for 2-m temperature and humidity from ‘*Modèle de Circulation Générale du LMD*’ (LMDZ, <https://lmdz.lmd.jussieu.fr> Hourdin et al., 2006) are introduced into WRF as additional diagnostics. These implementations follow an iterative methodology following (Hess et al., 1995). These variables are activated when compiling the module with the pre-comilation flag `-DCORDEXDIAG`

Two new variables are introduced:

- TAS_HM: Corrector-calculator (Hess-McAvaney, 1995) 2m temperature, K
- QVS_HM: Corrector-calculator (Hess-McAvaney, 1995) 2m water vapor mixing ratio, K

It follows a 2-step methodology:

1. First estimate using the Dyer-Businger formulation
2. Corrector of the estimates using the Louis formulation

In more detail (following Hess et al., 1995) see appendix B.1.

3.6 Implementation in the module

Following LMDZ atmospheric model, we need at the same time, to make sure that the diagnostics are computed using as much possible the actual values of the variables simulated by the model. Since WRF-ARW computes its own drag coefficients of momentum (C_m) and latent/sensible heat (C_h), we must use them.

Here, code from an old version of LMDZ is used (see current version in appendix B.2). The main subroutine in `phys/module_diagvar_cordex.F` is `SFCDIAGS_ORCHIDEE1D`

1. The * variables Θ^* (temperature) and Q^* (humidity) are computed by the equations 14 and 15

$$\Theta^* = \frac{C_h \delta_z t wss|_{k=1}}{u^*} \quad (14)$$

$$Q^* = \frac{C_h \delta_z q wss|_{k=1}}{u^*} \quad (15)$$

where $\delta_z t = \theta|_{k=1} - t_{sfc}$: potential temperature difference between first model level ($\theta|_{k=1} = ta|_{k=1}(p_{sfc}/p|_{k=1})^{R_d/C_p}$) and surface (t_{sfc}), $\delta_z q = \max(q|_{k=1}, 0) - \max(q_{sfc}, 0)$, correcting $\delta_z t = \text{sign}(\max(\text{abs}(\delta_z t), 1.e - 10), \delta_z t)$, $wss|_{k=1}$: wind speed at the first model level

WRF/CDX iut, ivt

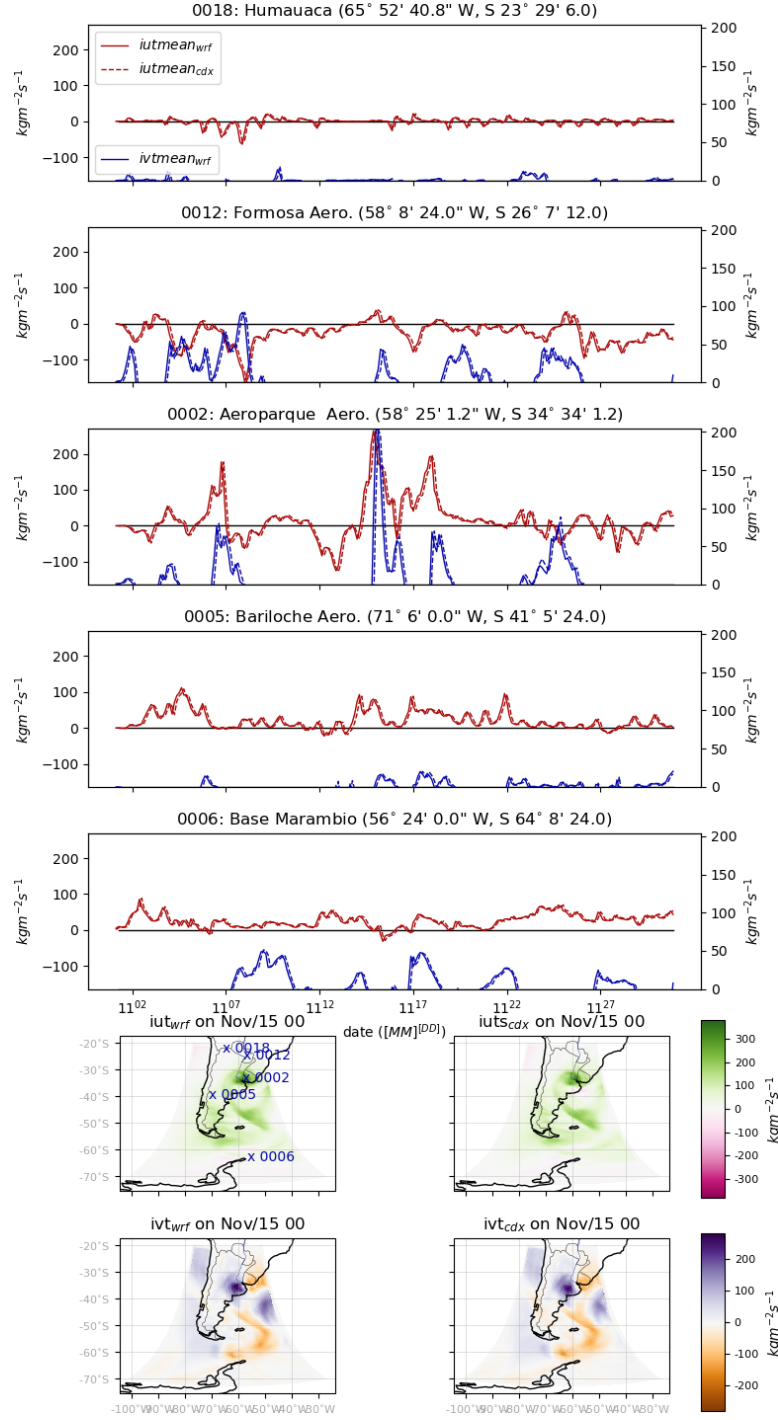


Figure 7: 2018 November 3-hourly evolution of WRF's 3-hourly mean vertically integrated water vapour eastward horizontal transport (iut_{wrf}) and northward horizontal transport (ivt_{wrf}) and CORDEX module based on internal integration values iut_{cdx} , ivt_{cdx} at different locations. Values of iut and ivt on 2011 November 15th at 00 UTC all over the domain

2. The first guess of the Δ are computed at $z_{ref} = 2 \text{ m}$; Δwss , $\Delta\Theta$, ΔQ (from 16 to 18, obtained using subroutine `screenc1D`)

$$\Delta wss = \begin{cases} \text{unstable } (\mathcal{L}_0 \geq 0) & \begin{cases} \frac{u^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) + \min \left(5, 5 \frac{z_{ref}-z_0}{\mathcal{L}_0} \right) \right] & wss > 1.5 \text{ \& } \mathcal{L}_0 < 1 \text{ \& } z_0 \leq 1 \\ 0.1 wss & \text{otherwise} \end{cases} \\ \text{stable} & \begin{cases} \frac{u^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) - 2 \ln (0.5(1 + x_{tmp})) + \right. \\ \quad 2 \ln (0.5(1 + x_{tmp0})) - \\ \quad \ln (0.5(1 + x_{tmp}^2)) + \ln (0.5(1 + x_{tmp0}^2)) + \\ \quad \left. 2 \arctan(x_{tmp}) - 2 \arctan(x_{tmp0}) \right] & wss > 5.0 \text{ \& } abs(\mathcal{L}_0) \leq 50.0 \\ 0.5 wss & \text{otherwise} \end{cases} \end{cases} \quad (16)$$

$$\Delta\Theta = \begin{cases} \text{unstable } (\mathcal{L}_0 \geq 0) & \begin{cases} \frac{\Theta^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) + \min \left(5, 5 \frac{z_{ref}-z_0}{\mathcal{L}_0} \right) \right] & wss > 1.5 \text{ \& } \mathcal{L}_0 < 1 \text{ \& } z_0 \leq 1 \\ 0.1 (\theta_{air} - \theta_s) & \text{otherwise} \end{cases} \\ \text{stable} & \begin{cases} x_{tmp} = (1 - 16(z_{ref}/\mathcal{L}_0))^{1/4} \\ x_{tmp0} = (1 - 16(z_0/\mathcal{L}_0))^{1/4} \\ \frac{\Theta^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) - 2 \ln (0.5(1 + x_{tmp}^2)) + \right. \\ \quad \left. 2 \ln (0.5(1 + x_{tmp0}^2)) \right] & wss > 5.0 \text{ \& } abs(\mathcal{L}_0) \leq 50.0 \\ 0.5 (\Theta_{air} - \Theta_s) & \text{otherwise} \end{cases} \end{cases} \quad (17)$$

$$\Delta Q = \begin{cases} \text{unstable } (\mathcal{L}_0 \geq 0) & \begin{cases} \frac{Q^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) + \min \left(5, 5 \frac{z_{ref}-z_0}{\mathcal{L}_0} \right) \right] & wss > 1.5 \text{ \& } \mathcal{L}_0 < 1 \text{ \& } z_0 \leq 1 \\ 0.1 [\max(Q_{air}, 0) - \max(Q_{surf}, 0)] & \text{otherwise} \end{cases} \\ \text{stable} & \begin{cases} \frac{Q^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_0} + 1 \right) - 2 \ln (0.5(1 + x_{tmp}^2)) + \right. \\ \quad \left. 2 \ln (0.5(1 + x_{tmp0}^2)) \right] & wss > 5.0 \text{ \& } abs(\mathcal{L}_0) \leq 50.0 \\ 0.5 [\max(Q_{air}, 0) - \max(Q_{surf}, 0)] & \text{otherwise} \end{cases} \end{cases} \quad (18)$$

where wss : wind speed at 1st atmospheric model level, Θ_{air} : air temperature at first model level, Q_{air} : specific humidity at first model level, Θ_s : surface air temperature (tsk), Q_{surf} : surface specific humidity, z_0 : rugosity, \mathcal{L}_0 : Monin-Obukov length, u^* : scale factor for the wind, Θ^* : scale factor for the potential temperature, Q^* : scale factor for the humidity, z_{ref} : reference height, Δwss : wind anomaly respect the first model level, $\Delta\Theta$: potential temperature anomaly respect the surface, ΔQ : humidity anomaly respect the surface.

3. Using them, a first guess is computed at the reference height ($z_{ref} = 2$) as by equations from 19 to 22

$$wss_{z_{ref}} = \Delta wss \quad (19)$$

$$q_{z_{ref}} = \max(q_{surf}, 0) + \Delta Q \quad (20)$$

$$\Theta_{z_{ref}} = t_{sfc} + \Delta\Theta \quad (21)$$

$$ta_{z_{ref}} = \Theta_{z_{ref}} \left(\frac{p_{sfc}}{p|_{k=1}} \right)^{-R_d/C_p} \quad (22)$$

4. Iterate to get corrected values by recomputing the coefficients ($\mathcal{C}_m, \mathcal{C}_h$) at the new reference values (χ_{ref} , using subroutine `coefcdrag1D`), applying slight corrections to the computed terms as it is shown in equations 24 to 26. In the CDXWRF implementation this is unnecessary, since, we will not get new drag coefficients, since any of the them changes during the diagnostics in order to keep the same values as the ones provided by WRF and being consistent with what the model simulates.

$$\mathcal{C}'_m, \mathcal{C}'_h, \mathcal{C}'_{neutral}, R'_i, p'_{ref} = \text{coefcdrag1D}() \quad (23)$$

$$\Delta^{corr} wss = \frac{u^*}{\sqrt{\mathcal{C}'_m}} \quad (24)$$

$$\Delta^{corr} \Theta = \Theta^* \frac{\sqrt{\mathcal{C}'_m}}{\mathcal{C}'_h} \quad (25)$$

$$\Delta^{corr} Q = Q^* \frac{\sqrt{C'_m}}{C'_h} \quad (26)$$

where C'_m : corrected drag coefficient for momentum, C'_h : corrected drag coefficient for sensible and latent heat at ($\chi_{z_{ref}}$). Accordingly to the consistency with WRF drag and Richardson number values, subroutine `coefcdrag1D` is modified and only provides updated R'_i and p'_{ref} , therefore $C'_m = C^{wrf}_m$, $C'_h = C^{wrf}_h$ and the Δ are the same as the ones computed at the reference level z_{ref} ; $\Delta^{corr} wss = \Delta wss$, $\Delta^{corr} \Theta = \Delta \Theta$ and $\Delta^{corr} Q = \Delta Q$

In order to obtain the pressure p_{ref} at the reference level (z_{ref}), de equation 27 is used

$$p_{ref} = \exp \left[\ln(p_{sfc}) - \frac{zg|_{k=1}}{R_d ta|_{k=1} (1 + p_{608} max(q|_{k=1}, 0))} \right] \quad (27)$$

where p_{sfc} : surface pressure, $zg|_{k=1}$: geopotential height at the first level of the model, $ta|_{k=1}$: air temperature at the first level of the model, $q|_{k=1}$: air humidity at the first level of the model, $p_{608} = R_v/R_d - 1$

5. The new corrected values become 28 to 31

$$wss'_{z_{ref}} = \Delta^{corr} wss \quad (28)$$

$$Q'_{z_{ref}} = \Delta^{corr} Q + max(q_{sfc}, 0) \quad (29)$$

$$\Theta'_{z_{ref}} = \Delta^{corr} \Theta + ta|_{k=1} \quad (30)$$

$$ta' = \Theta_{z_{ref}} (p_{sfc}/p_{ref})^{-R_d/C_p} \quad (31)$$

6. After 2 iterations (steps 4 and 5), the final diagnosed values become 32 and 33

$$t_{2m} = ta' \quad (32)$$

$$\begin{aligned} q_{2m} &= Q'_{z_{ref}} \\ q_s &= \frac{pq0}{p0} \exp \left(\frac{a2(t_{2m} - a3)}{t_{2m} - a4} \right) \\ q_{2m} &= \min(q_s, q_{2m}) \end{aligned} \quad (33)$$

where $a2 = 17.2693882$, $a3 = 273.16$ and $a4 = 35.86$

So in summary, in CDXWRF module, implementation of diagnostics of tas and hurs at 2 m following (Hess et al., 1995) methodology, does not applies any correction-iterative method and only uses the first guess (see comparison with standard WRF values in figure 8 Code is inherited from (PhD thesis of Stéfannon, 2012) and for legacy, it will be kept in this way.

WRF does not provide the momentum surface exchange coefficient, so, a generic definition of it (shown in equation 34 Garratt, 1992) is used which was already introduced in the v1.3 of the WRF-CORDEX module.

$$Cd_m = \frac{(u^*)^2}{uas^2 + vas^2} \quad (34)$$

Heat momentum exchange coefficient (Cd_h) is provided as it is shown in equation 35

$$Cd_h = \frac{chs}{uas^2 + vas^2} \quad (35)$$

where chs : surface exchange coeff for heat (ms^{-1} , CH in WRF) and uas, vas : 2-m wind speed components (U10 and V10 in WRF)

In order to avoid non consistent values, diagnosed values t_{2m} , q_{sm} are constrained within the surface values t_{sfc} , qv_{sfc} and model's first layer values $ta|_{k=1}$, $qv|_{k=1}$ following the equations at 36

$$\begin{cases} tamin = \min(t_{sfc}, ta|_{k=1}) \\ qvmin = \min(qv_{sfc}, qv|_{k=1}) \\ tamax = \max(t_{sfc}, ta|_{k=1}) \\ qvmax = \max(qv_{sfc}, qv|_{k=1}) \end{cases}$$

$$\begin{aligned}
t_{2m} &< t_{amin} \rightarrow t_{2m} = t_{amin} \\
q_{2m} &< q_{vmin} \rightarrow q_{2m} = q_{vmin} \\
t_{2m} &> t_{amax} \rightarrow t_{2m} = t_{amax} \\
q_{2m} &< q_{vmax} \rightarrow q_{2m} = q_{vmax}
\end{aligned} \tag{36}$$

3.7 tws

Wet-bulb temperature corresponds to the temperature that would have the air if saturated. This is used in surface stations in order to determine the humidity of the surface. It is also of importance for well being, since it provides a good estimate of the physical stress induced for high temperature and humidity (Raymond et al., 2020; Vecellio et al., 2022). This variable is activated with the pre-compilation flag `-DCDXWRF=2`

In this implementation is done from the work of Stull (2011). With relative humidity and temperature, the wet-bulb temperature is derived as it is shown in equation 37.

$$\begin{aligned}
tws = & tas \tan^{-1} \left(0.151977 \sqrt{hurs + 8.313659} \right) + \tan^{-1}(tas + hurs) - \\
& \tan^{-1}(hurs - 1.676331) + 0.00391838 (hurs)^{1.5} \tan^{-1}(0.023101 hurs) - 4.686035
\end{aligned} \tag{37}$$

where *tas*: 2-m temperature, *hurs*: 2-m relative humidity.
3 new variables are addedd to the output:

- TWSMIN: minimum surface wet-bulb temperature, *K*
- TWSMAX: maximum surface wet-bulb temperature, *K*
- TWSMEAN: mean surface wet-bulb temperature, *K*

3.8 qc_pl, qr_pl, qs_pl, qi_pl, qg_pl, qh_pl

WRF directly interpolates to pressure level values during model integration including water vapour mixing ratio. In this version, the interpolation of the rest of main water species have been included (qc: cloud, qr: rain, qs: snow, qi: ice, qg: graupel and qh: hail). This is automatically added when compiling the module.

5 new variables are added to the `wrfpress (auxhist23)` output.

- QC_PL: Pressure level data, cloud mixing ratio, $kgkg^{-1}$
- QR_PL: Pressure level data, rain mixing ratio, $kgkg^{-1}$
- QS_PL: Pressure level data, snow mixing ratio, $kgkg^{-1}$
- QI_PL: Pressure level data, ice mixing ratio, $kgkg^{-1}$
- QG_PL: Pressure level data, graupel mixing ratio, $kgkg^{-1}$
- QH_PL: Pressure level data, hail mixing ratio, $kgkg^{-1}$

3.9 fixing convection diagnostics

Previous version was using AFWA diagnostics from WRF v3.7.1 to diagnose convection related indices: CAPE, CIN, ZLFC, PLFC, LI. In this version, a more updated version of the diagnostics (v4.3.1) has been used, since it corrects some mistakes in the diagnostics (thanks to Zhixiao)

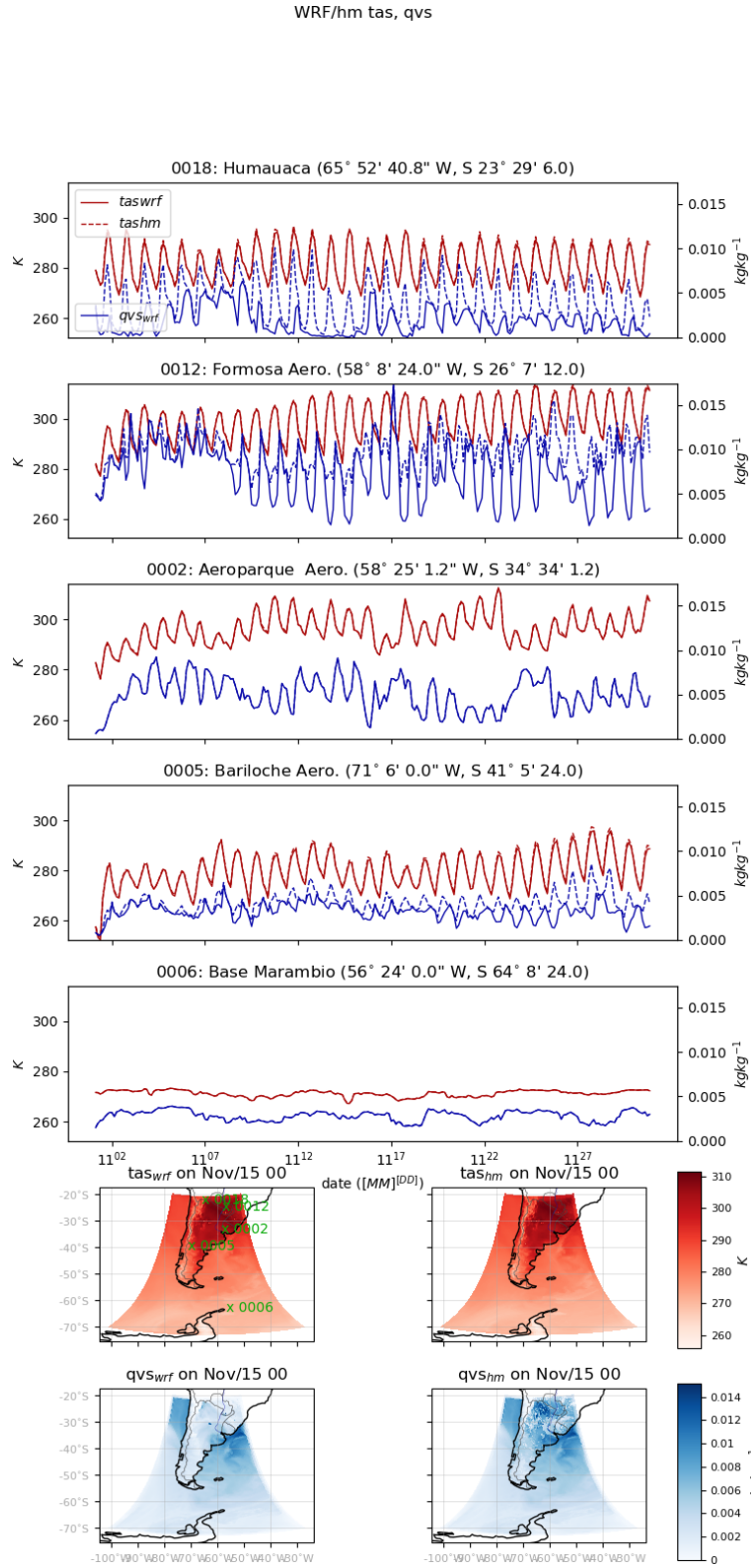


Figure 8: 2018 November 3-hourly evolution of WRF's 2-m air temperature (tas_{wrf}) and vapour mixing ratio (qvs_{wrf}) and LMDZ-based tas_{hm} , qvs_{hm} at different locations. Values of 2-m air temperature and vapour mixing ratio on 2011 November 15th at 00 UTC all over the domain

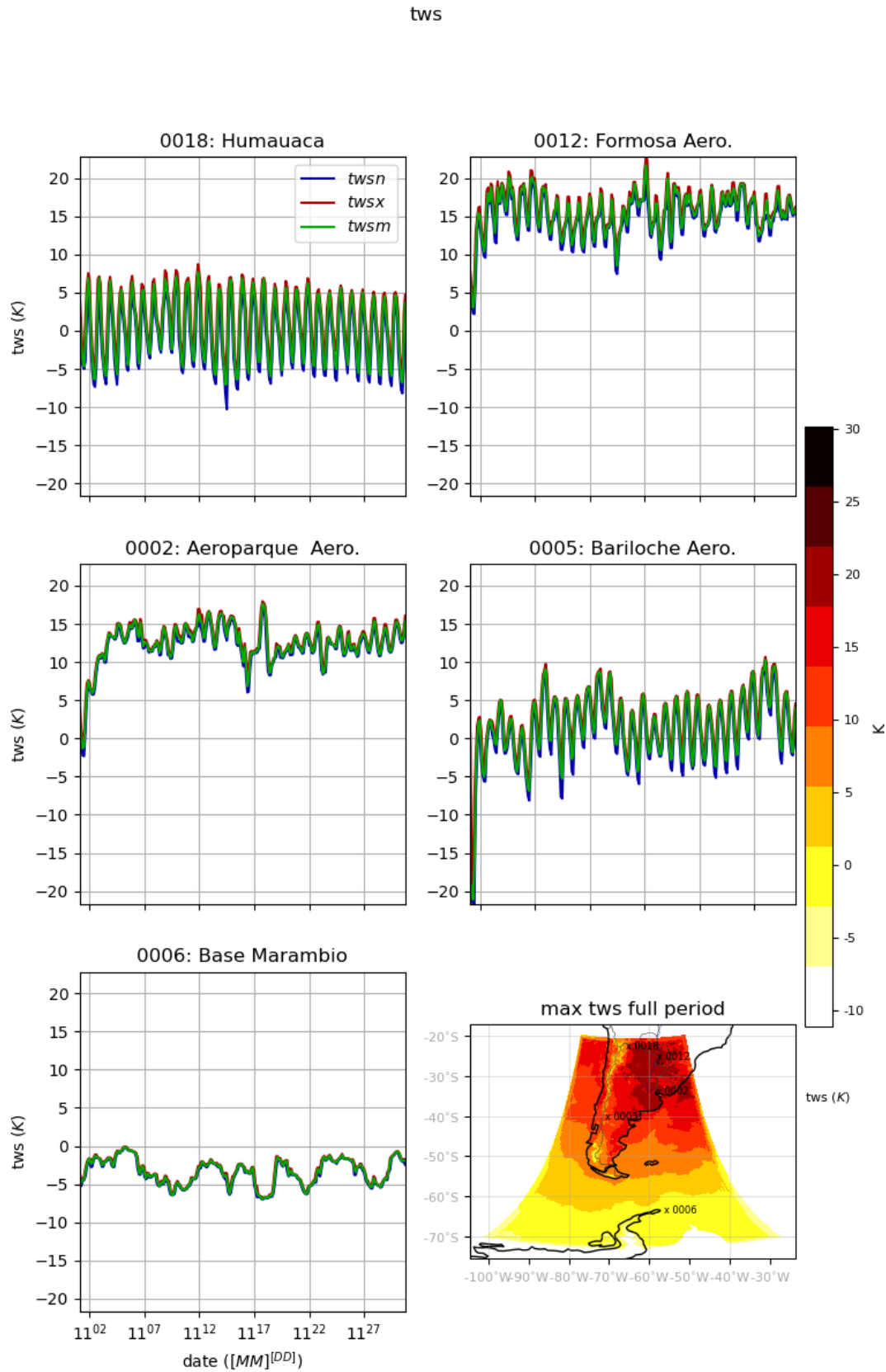


Figure 9: wet-bulb diagnostic evolution at 5 different weather stations and the maximum of tws during the period of simulation

3.10 x_pl_mc

Currently vertical interpolation of water species to various pressure-levels is done by simple linear interpolation using the closest levels of the model. However, this induces a serious problem related to the lack of conservation of mass. Total amount of water in the column is not conserved when compared with the amount of water in the vertical of the model and the values interpolated at pressure-levels.

Here a methodology to preserve water in the column is introduced. The idea is to preserve the amount of water in the column (the mass) in a similar way as it is done when re-project surface variables such as precipitation or surface fluxes from the original projection used by the model to a new one.

This option is activated when compiling the module. It uses the same vertical pressure levels configuration as the one set-up for the standard vertical interpolation.

6 new variables are added to the wrfpress (`auxhist23`) output.

- QV_PL_MC: mass-conservative pressure level data, water vapour mixing ratio, $kgkg^{-1}$
- QC_PL_MC: mass-conservative pressure level data, cloud mixing ratio, $kgkg^{-1}$
- QR_PL_MC: mass-conservative pressure level data, rain mixing ratio, $kgkg^{-1}$
- QS_PL_MC: mass-conservative pressure level data, snow mixing ratio, $kgkg^{-1}$
- QI_PL_MC: mass-conservative pressure level data, ice mixing ratio, $kgkg^{-1}$
- QG_PL_MC: mass-conservative pressure level data, graupel mixing ratio, $kgkg^{-1}$
- QH_PL_MC: mass-conservative pressure level data, hail mixing ratio, $kgkg^{-1}$

In the vertical, due to the vertical discretization imposed in the model, mixing ratios of any water specie are constant within each of the grid cells. Each individual cell has a base and a top in the vertical dimension. When interpolating to a new fixed set of pressure-levels, we can consider them as a new discretization of the vertical axis (usually to a coarse resolution). These new cells, will have also an extension in the vertical with a bottom and a top. It is desired that the total amount of mass ($totmass_\chi$) of the water specie (χ) is preserved when vertically integrated using the model levels (p_{mod}) and the new desired p-levels (p_{des}), as it is shown in equation 38.

$$totmass_\chi = \sum_{ko=1}^{N_{mod}} mass_\chi(ko) = \sum_{kd=1}^{N_{des}} mass_\chi(kd) \quad (38)$$

where χ : water vapour, cloud, rain, snow, ice, graupel and ice, $mass_\chi(k)$: mass of the water specie χ at the k -level, ko : original vertical index in the model, kd : new desired vertical index to interpolate, N_{mod} : amount of vertical levels in the model, N_{des} : amount of desired p-level vertical levels.

Bearing in mind that water content is constant within the column, the idea is to interpolate to the 'new vertical discretization' aggregating all the water content which lays within the new grid cell (as is schematically described in equation 39)

$$mass_{chi}(kd) = \sum_{koin=koin_1}^{koin_N} mass_{chi}(koin)\delta z(koin) \quad (39)$$

where $koin_1$, $koin_N$: model vertical levels within the 'new' vertical level, $\delta z(koin)$, height of the k -model level within the 'new' vertical level.

In order to mass-conservative vertically interpolate, we follow the next steps:

1. In order to make the interpolation, the mixing ratios of the water species q_χ (in $kgkg^{-1}$), are transformed to densities ρ_χ (in kgm^{-3}) using the density of the dry air (ρ_{dry}) as it is shown in equation ??

$$\rho_\chi = \frac{q_\chi}{\rho_{dry}} \quad (40)$$

2. Define vertical extent of the grid cells.

- Original levels of the model:

- (a) WRF's geopotential height (zg in m^2s^{-2}) is vertically staggered, therefore, there is already a value at the bottom and top of each grid cell. Actual extent of the grid cells ($obthgt$, original bottom-top height, in m) which is dynamic (since WRF is no-hydrostatic) will be computed as it is shown in equation 41

$$obthgt(ko) = \begin{cases} hgt_{bot}(ko) = zg(ko)/g \\ hgt_{top}(ko) = zg(ko + 1)/g \end{cases} \quad (41)$$

where g : gravity (9.81 ms^{-2})

- (b) Height at the center of the grid cell ($ohgt$, in m) will be defined as the average of the height at the bottom and top (see eq. 42)

$$ohgt(ko) = 0.5 (obthgt(ko) + obthgt(ko + 1)) \quad (42)$$

- Desired pressure levels:

- (a) There is a list of desired pressure levels at which we want to interpolate ($presdes$). The vertical extent of these grid cells ($btpresdes$, in Pa) will be defined as the mid-point between desired pressure levels. Being the surface pressure ($psfc$) the bottom limit and the pressure at the top of the model ($ptop$) the upper limit (see eq. 43). The desired pressure levels that might lay below ground ($presdes < psf$) will be filled with missing values.

$$btpresdes(kd) = \begin{cases} 0.5 (presdes(kd - 1) + presdes(kd)) \\ 0.5 (presdes(kd) + presdes(kd + 1)) \end{cases} \quad (43)$$

$$; kd = 1, btpresdes(kd) = psfc$$

$$; kd = N_{des}, btpresdes(N_{des} + 1) = ptop \quad (44)$$

- (b) The height of the bottom, center and top of the cells will be retrieved using the pressure ($pres$) and original model heights ($ohgt$, $obthgt$) fields of the model.

- i. Look for the 2 model's pressure nearest to the desired pressure (either for the bottom, center or top) from which one obtains $pres_{bottom}(ko')$, $pres_{top}(ko' + 1)$ (see equation 45)

$$ko'; pres(ko') \leq presdes_{bot,center,top} < pres(ko' + 1) \quad (45)$$

- ii. Using ko' and the various original model heights $ohgt/obthgt(ko')$, $ohgt/obthgt(ko' + 1)$, a linear interpolation at the $presdes_{bot,center,top}$ is used to obtain the value ($preshgtf$ and $btpreshgtf$, see eq. 46)

$$preshgt_{center}(kd) = \mathcal{LinInterp}[pres(ko'), pres(ko' + 1), ohgt(ko'), ohgt(ko'), pres(kd)] \quad (46)$$

$$btpreshgt(kd) = \mathcal{LinInterp}[pres(ko'), pres(ko' + 1), obthgt(ko'), obthgt(ko' + 1), btpres(kd)]$$

3. Once all the heights (center, bottom and top) have been defined for both sets (model, desired) of vertical discretizations, the mass-conservative vertical interpolation ($\rho\chi_{pl_mc}$) of the original densities ($\rho\chi$) can be done as shown in equation 47

$$\begin{aligned} valdes(kd) &= \sum_{koin=1}^{Nlev} \delta_{levorig}(ko) \rho\chi(ko) \\ &+ \delta_{levorig}^{top}(itopkin) \rho\chi(itopkin) \\ &+ \delta_{levorig}^{bot}(ibotkin) \rho\chi(ibotkin) \\ \rho\chi_{pl_mc}(kd) &= valdes(kd) / Dlevdes \end{aligned} \quad (47)$$

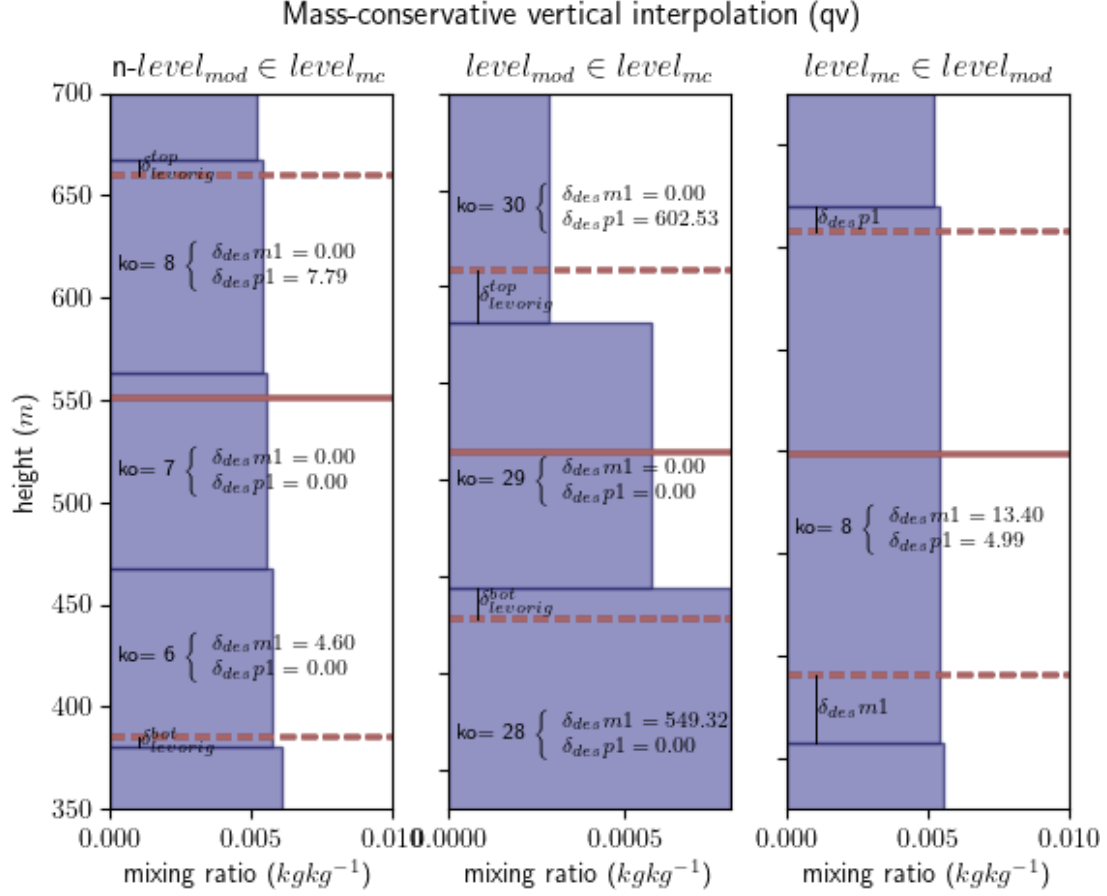


Figure 10: Schematic representation of the definition of the different values used to compute the mass at certain desired level (kd) and the respect k -level values for $\delta_{des}m1$, $\delta_{des}p1$, $\delta_{levorig}^{bot}$ and $\delta_{levorig}^{top}$. When kd encompasses multiple original model levels $ko = [6, 8]$ (left). When kd covers totally a single original model level $ko = 29$ and portions of other ones $ko = 28, 30$ (middle), When kd is within a single model level $ko = 8$ (right)

where $\rho\chi$: original model's density value, $\delta_{levorig}$: height of the original model level within the desired one (see eq. 48, $D_{levdes} = deskp1 - deskm1$: vertical extent of the desired cell, $D_{levorig} = orikp1 - orikm1$: vertical extent of the original model cell.

$$\delta_{levorig}(ko) = D_{levorig} - \delta_{des}m1 - \delta_{des}p1 \quad (48)$$

where $\delta_{des}[m/p]1$: is the vertical extent of the lower(top) section of the model's cell within the desired cell (see eq. 49 and figure 10)

$$\delta_{des}m1 = \begin{cases} deskm1 - orikm1 & deskm1 > orikm1 \\ 0 & deskm1 < orikm1 \end{cases} \quad \delta_{des}p1 = \begin{cases} orikp1 - deskp1 & deskp1 < orikp1 \\ 0 & deskp1 > orikp1 \end{cases} \quad (49)$$

where $\delta_{levorig}^{top}(itopkin)$: is the vertical extent of the top section of the model's cell within the desired cell ($ko' = itopkin$), but with model's cell center outside the desired cell range (see eq. 50)

$$\delta_{levorig}^{top}(itopkin) = orikp1 - deskm1 \quad (50)$$

where $\delta_{levorig}^{bot}(ibotkin)$: is the vertical extent of the bottom section of the model's cell within the desired cell ($ko' = ibotkin$), but with model's cell center outside the desired cell range (see eq. 51)

$$\delta_{levorig}^{bot}(ibotkin) = deskp1 - orikm1 \quad (51)$$

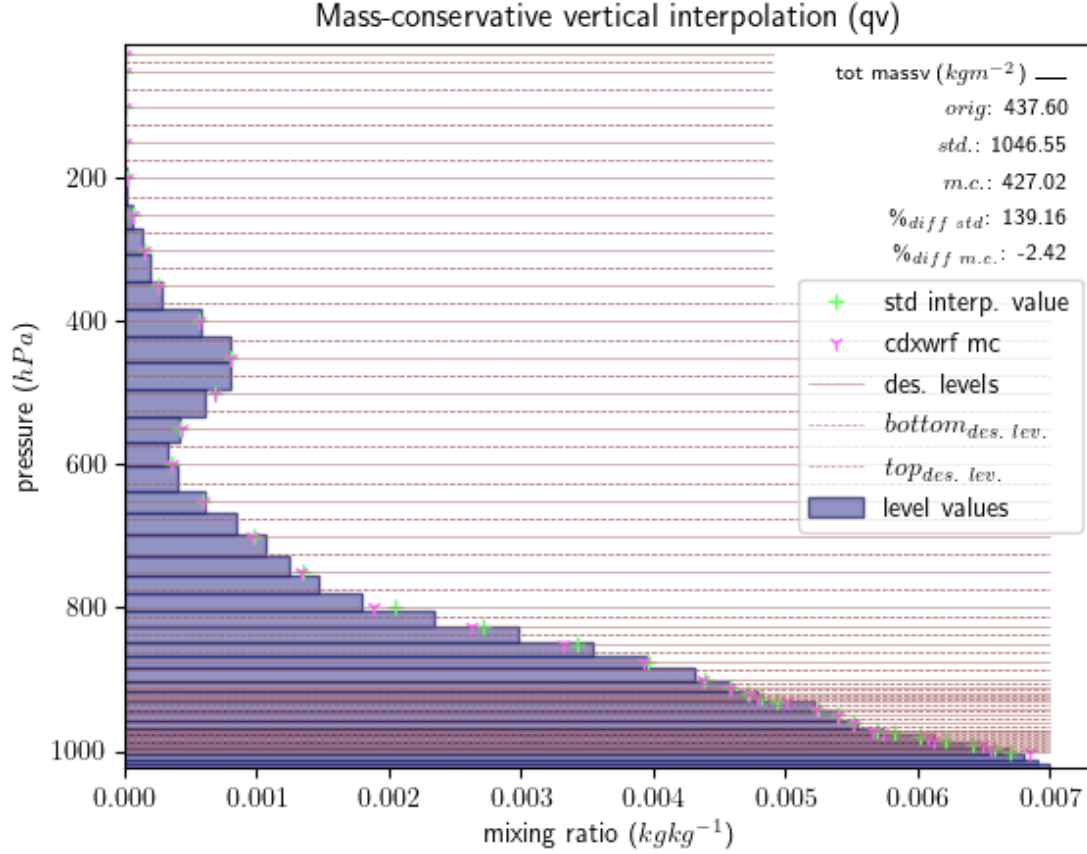


Figure 11: Validation of the mass-conservative interpolation at a given grid point for water vapour mixing ratio (qv in $kgkg^{-1}$, x-axis) for 36 vertical desired pressure levels. Color bars in pale blue are the different qv at the original model η levels (at each respective pressures, y-axis). '+' interpolation using the standard linear interpolation, 'tri-star' interpolation using the mass-conservative interpolation. Straight lines indicate the desired pressure value for the vertical interpolation. Dotted lines indicate the bottom/top of the desired vertical layers. Top right values illustrate the total column mass of water vapour (in kgm^{-2}) for each case. 'orig': original model mass, 'std': mass using standard interpolated values, 'mc': mass using the mass-conservative interpolation method. %diff std.: percentage of difference for the standard total mass with respect the one from the model, %diff m.c.: percentage of difference for the mass-conservative total mass with respect the one from the model

where $orikm1$: value at the base of the original cell ($obthgt(ko')$), $orikp1$: value at the top of the original cell ($obthgt(ko' + 1)$), $deskm1$: value at the base of the desired cell ($obtpreshgt(kd)$), $deskp1$: value at the top of the desired cell ($obtpreshgt(kd + 1)$), $Nlev$: amount of original levels within the height of the kd -desired cell ($Nlev$; $orilev(ko) \leq deskm1$ & $orilev(ko) > deskp1$)

4. Dry air density (ρ_{dry}) of the model is also mass-conservatively interpolated to the desired pressure levels to provide $\rho_{dry_pl_mc}$
5. Finally, obtained interpolated water densities at the desired pressure levels of specie $\rho_{\chi_pl_mc}$, will be transformed to mixing ratios $q_{\chi_pl_mc}$ using the interpolated dry density ρ_{dry_des} as it is shown in eq. 52 (see figure 11)

$$q_{\chi_pl_mc} = \frac{\rho_{\chi_pl_mc}}{\rho_{dry_pl_mc}} \quad (52)$$

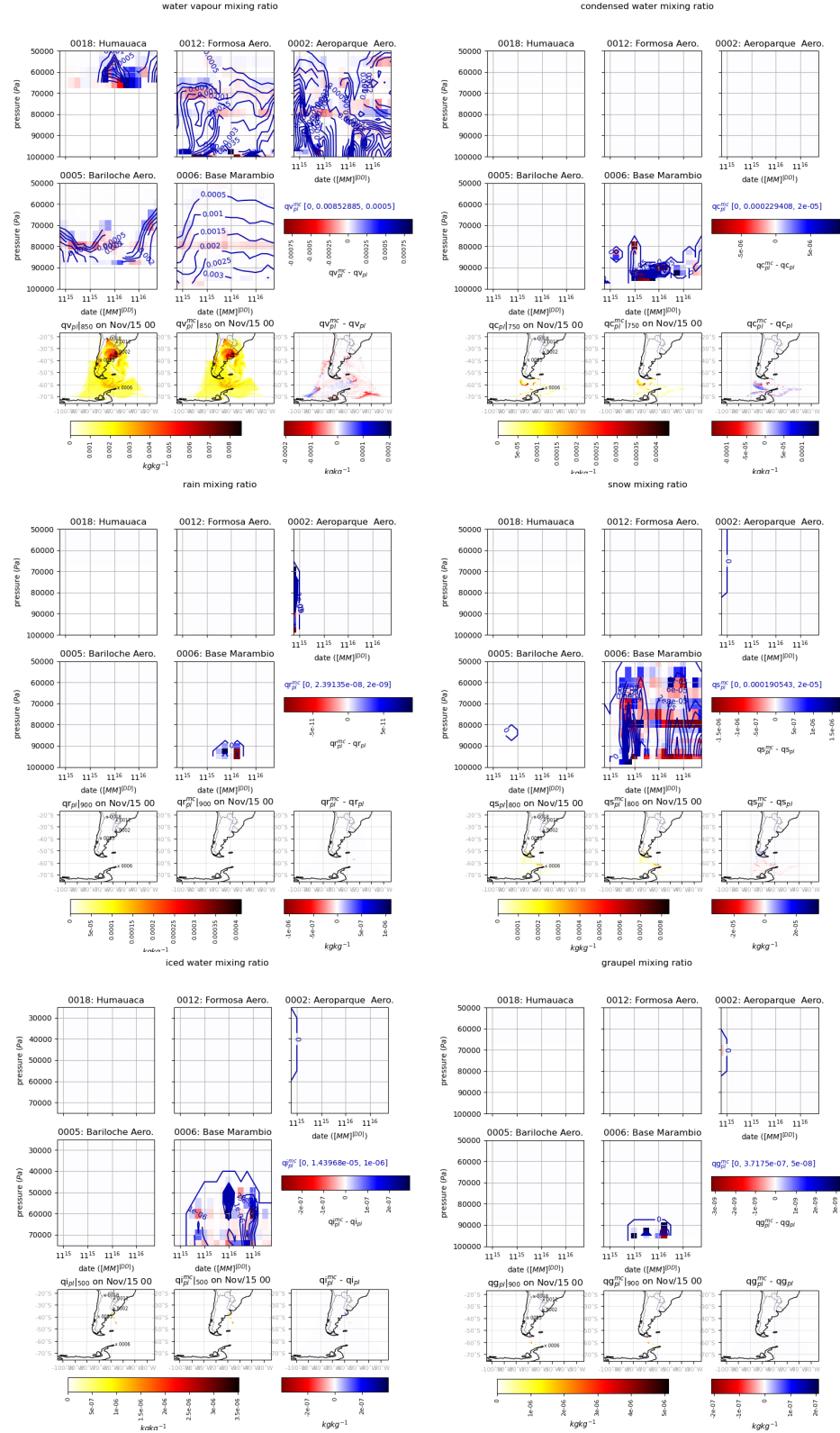


Figure 12: Differences between standard vertical interpolation ($q\chi_{pl}$) and mass-conservative ones ($q\chi_{pl}^{mc}$) over 23 pressure levels. Differences for vapour mixing ratio (qv), cloud (qc), rain (qr), snow (qs), ice (qi) and graupel (qg). Evolution of vertical profiles (contour in blue for mass. consev. interpolation) and differences among interpolations (shaded) at 5 different stations. Maps at a given vertical level and instant (2011 November 15th at 00 UTC) for each interpolation and the differences

3.11 simultaneous residence time

This variable aggregates the time passed (*'time of residence'*) during the integration of the model, at each grid point of the domain, where 2 different variables (in this case: 2-m temperature (tas) and humidity (hurs)) simultaneously passed at within certain range of values (2 bins). The variable will reflect the total time passed at the given location where tas values were within a given range and hurs values within another one. This variable is related to the *growing degree-day* (GDD¹) used in agriculture or pest control. It is more sophisticated, because it incorporates the simultaneity with relative humidity which has also a strong influence in the life cycle of insects (to cite some examples [Rudolfs, 1925](#); [Kingsolver et al., 2011](#); [Tamiru et al., 2012](#)). This diagnose is activated when compiling with `-DCDXWRF=3`.

The diagnose introduces 2 new variables:

- TASHURSTRESHIGHRES: high-resolution of simultaneous temporal residence of 2-m temperature and relative humidity, *second*
- TASHURSTRESLOWRES: low-resolution of simultaneous temporal residence of 2-m temperature and relative humidity, *second*

The option is set-up with 6 new parameters in the namelist:

- `nhtasrng`: amount + 1 of high-resolution bins for temporal residence of tas (default value 42)
- `nhhursrng`: amount + 1 of high-resolution bins for temporal residence of hurs (default value 23)
- `nltasrng`: amount + 1 of low-resolution bins for temporal residence of tas (default value 16)
- `nlhursrng`: amount + 1 of low-resolution bins for temporal residence of hurs (default value 23)
- `htasrng`: high-resolution bins for temporal residence of tas (default value: -5 to 35 by 1. °C)
- `hhursrng`: high-resolution bins for temporal residence of hurs (default value: 0. to 1.05 by 0.05 %)
- `ltasrng`: low-resolution bins for temporal residence of tas (default value: -25 to 45.by 5. °C)
- `lhursrng`: low-resolution bins for temporal residence of hurs (default value: 0. to 1.05 by 0.05 %)

Two different high/low times of residence are defined as a way to provide valuable information to a larger amount of communities (e.g. glaciologists, ecologists, farmers, ...) without overloading the simulation. Is understood that the *high* resolution diagnostic is focused in the 'bio/eco-logical' area, whereas *low* resolution diagnostics might be more suitable for physical analyses (e.g.: glacial melting, heat stress, ...)

The diagnose is computes as follows

1. The tas variable is discretized by a series of bins: $tas(n), tas(n+1), tas(n+2), \dots, tas(N)$
2. The hurs variable is discretized by a series of bins: $hurs(m), hurs(m+1), hurs(m+2), \dots, hurs(M)$
3. The time of residence accumulates the time passed (during the integration of the model at time-steps dt) simultaneously at each possible combination of bins for tas and hurs (see eq. 53).

$$\begin{aligned}
 simultaneous_residence_time(i, j, n+1, m+1) &= \sum_{it} dt \left\{ \begin{array}{l} tas(n) \leq tas(i, j, it) < tas(n+1) \\ hurs(m) \leq hurs(i, j, it) < hurs(m+1) \end{array} \right. \\
 simultaneous_residence_time(i, j, 1, 1) &= \sum_{it} dt \left\{ \begin{array}{l} tas(i, j, it) < tas(1) \\ hurs(i, j, it) < hurs(1) \end{array} \right. \\
 simultaneous_residence_time(i, j, N+1, M+1) &= \sum_{it} dt \left\{ \begin{array}{l} tas(i, j, it) \geq tas(N) \\ hurs(i, j, it) \geq hurs(M) \end{array} \right.
 \end{aligned} \tag{53}$$

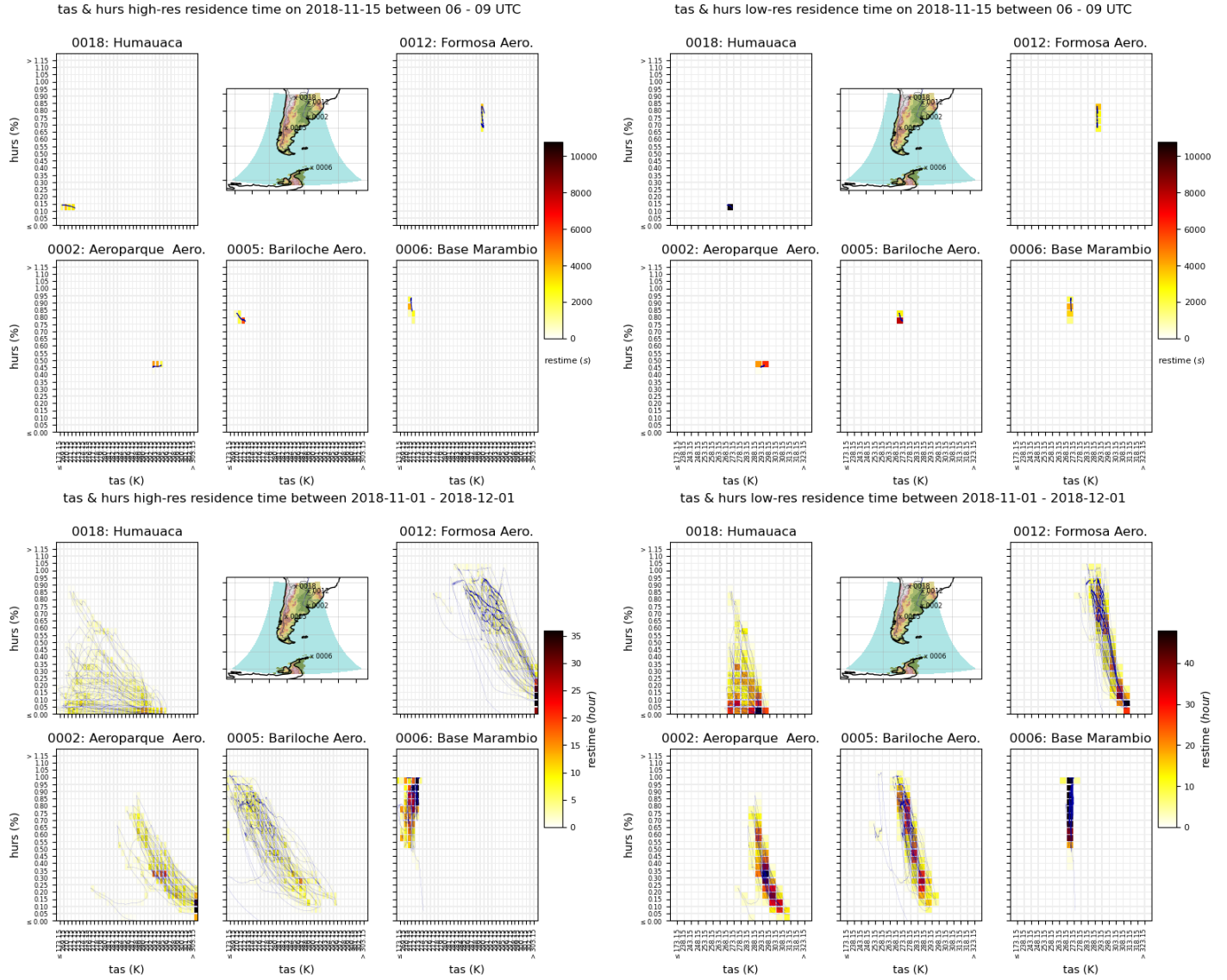


Figure 13: Example of residence time for 25 km simulation over entire Argentina and Chile at 5 different locations (see map and labels on the panels). Residence time in seconds on 2018 November 15th between . (top row). Total monthly residence time in hours for November 2018 (bottom row). For high-resolution bins (left column) for low-resolution bins (right column). Selected bins are given in equation 54

$$\left\{ \begin{array}{l}
tas_{highres} = \left[\begin{array}{l} 173.15, 268.15, 269.15, 270.15, 271.15, 272.15, 273.15, 274.15, 275.15, \\ 276.15, 277.15, 278.15, 279.15, 280.15, 281.15, 282.15, 283.15, 284.15, \\ 285.15, 286.15, 287.15, 288.15, 289.15, 290.15, 291.15, 292.15, 293.15, \\ 294.15, 295.15, 296.15, 297.15, 298.15, 299.15, 300.15, 301.15, 302.15, \\ 303.15, > 303.15 \end{array} \right] \\
hurs_{highres} = \left[\begin{array}{l} 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, \\ 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1, \\ 1.05, 1.1, 1.15, > 1.15 \end{array} \right] \\
tas_{lowres} = \left[\begin{array}{l} 173.15, 233.15, 238.15, 243.15, 248.15, 253.15, 258.15, 263.15, 268.15, \\ 273.15, 278.15, 283.15, 288.15, 293.15, 298.15, 303.15, 308.15, 313.15, \\ 318.15, 323.15, > 323.15 \end{array} \right] \\
hurs_{lowres} = hurs_{highres}
\end{array} \right. \quad (54)$$

4 WRF time-memory sensitivity to CDXWRF configuration

Here are presented the results of the changes in the speed and memory used of the simulation due to the new variables and updates of the module shown in figure 15. The test simulations are performed for the entire month of November of 2018 forced with ERA5 (see figure 14 for details about the domain). Simulation has been ran at CIMA's HPC using 2 nodes of 48 cores, 16 GB RAM each with Linux 5.10.0-25-amd64 #1 SMP Debian 5.10.191-1 (2023-08-16) x86_64 GNU/Linux with the following libraries and compilers: INTEL 2021.4.0: MPICH 3.4.2, NetCDF 4.8.1, HDF5 1.10.5, JASPER 2.0.33

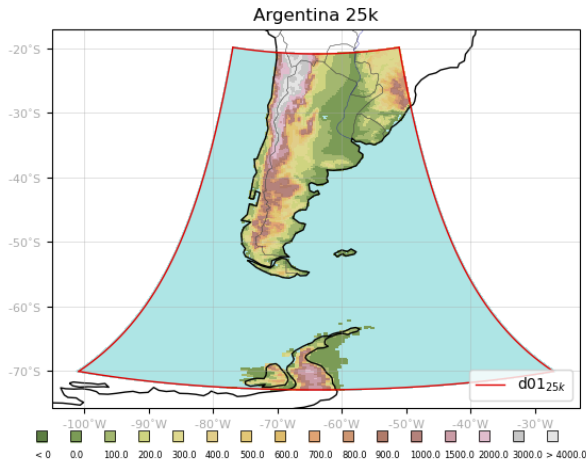
Acknowledgments

Thanks to Dr. Marisa Gassman and Natalia Tonti (DCAO, UBA, Argentina) for their suggestions, comments and explanations. RegIPSL developing team for their comments and patience. Phd candidate C. Mosto for the assistance in vertical integration. Dr. Jesús Fernández (IFCA, Spain), Dr. Zhixiao Zhang (University of Oxford, UK), PhD candidate Victoria Oruezabal (CIMA, Argentina), Prof. Jatin Kala (Murdoch University, Australia) and Prof. dr hab. inż. Mariusz J. Figurski and Adam Jaczewski (CMM, Poland). Simulations were performed at CIMA's HPC resources SNCAD and CIMA's IT support team are acknowledged. Dr A. Sörensson and Elsa Borrell Sörensson for the assistance and suggestions. L. Fita thanks the development of matplotlib above which he developed and make available a suite in python for netCDF management and plotting purposes called 'PyNCplot' (freely available from <http://www.xn--llusfb-5va.cat/python/PyNCplot>).

References

- Belamari, S. (2005). Report on uncertainty estimates of an optimal bulk formulation for surface turbulent fluxes. *MERSEA IP Deliverable 412*, pages 1–29.
- Belamari, S., Decharme, B., Guérémy, J.-F., Beau, I., Younous, A.-L. A., and members of the Task Force CNRM-CM (2016). Vers une nouvelle représentation des flux turbulents océan-atmosphère : la paramétrisation ecume revisitée. *Ateliers de Modélisation de l'Atmosphère, Toulouse 2016*, pages 1–18.
- Beljaars, A. (1992). The parametrization of the planetary boundary layer. *European Centre for Medium-Range Weather Forecasts*, page 40.
- Businger, J. A., Wyngaard, J. C., Izumi, Y., and Bradley, E. F. (1971). Flux-profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, 28(2):181–189.
- Dyer, A. J. (1974). A review of flux-profile relationships. *Boundary-Layer Meteorology*, 7(3):363–372.
- Fairall, C. W., Bradley, E. F., Hare, J. E., Grachev, A. A., and Edson, J. B. (2003). Bulk parameterization of air-sea fluxes: Updates and verification for the coare algorithm. *J. Climate*, 16(4):571 – 591.

¹https://en.wikipedia.org/wiki/Growing_degree-day



- grid points (dx,dy,dz): 120, 240, 50
- resolution (x,y): 25000, 25000 km
- projection: 'lambert'
- reference (lat, lon): S 47°, 64° W
- true lat: S 39°, S 55°
- standard lon: 64° W
- time-step: 90 s
- p_{top} : 150 hPa
- dz_bot: 10

&physics

- mp_physics: 8
- cu_physics: 0
- ra_lw_physics: 4
- ra_sw_physics: 4
- bl_pbl_physics: 5
- sf_sfclay_physics: 5
- sf_surface_physics: 4

- radt: 5

- bldt: 0

- cudt: 0

- icloud: 1

&bdy_control

- spec_bdy_width: 5
- specified: .true.

&diags

- num_press_levels: 23
- press_levels: 100000, 97500, 95000, 92500, 90000, 87500, 85000, 82500, 80000, 75000, 70000, 65000, 60000, 55000, 50000, 45000, 40000, 35000, 30000, 25000, 20000, 15000, 10000
- use_tot_or_hyd_p: 1

Figure 14: Domain of simulation and general WRF (v4.5.1) configuration. No nudging is used for the simulation. The rest of the parameters are mostly the default ones

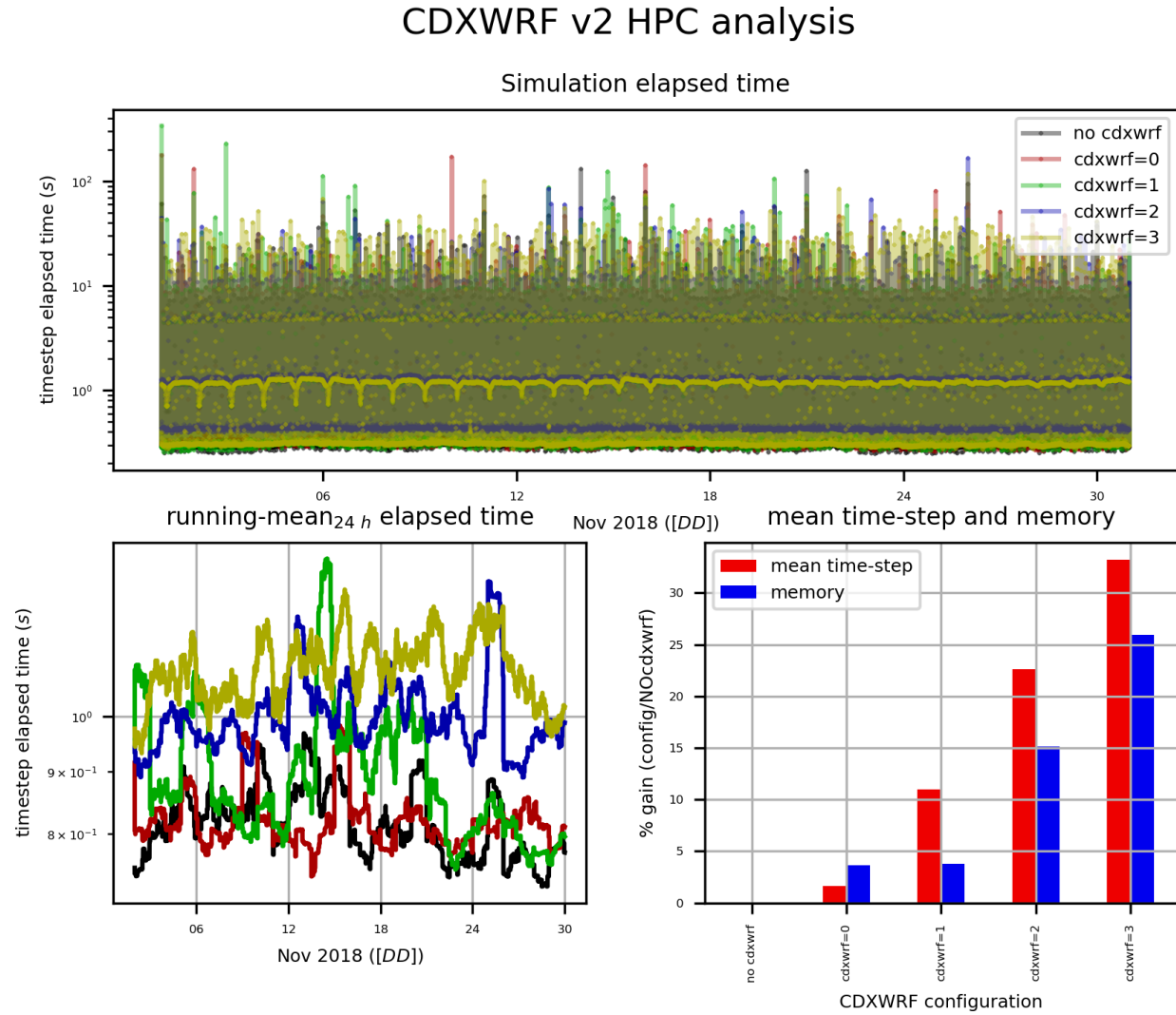


Figure 15: Evolution of the elapsed time for all time steps during 1-month simulation (November 2018) at domain described in figure 14 (top) for the different CDXWRF configurations. 24-h running mean of the the elapsed simulated time (bottom top). Simulation speed and memory gain for each configuration (bottom right)

- Fita, L., Polcher, J., Giannaros, T. M., Lorenz, T., Milovac, J., Sofiadis, G., Katragkou, E., and Bastin, S. (2019). Cordex-wrf v1.3: development of a module for the weather research and forecasting (wrf) model to support the cordex community. *Geosci. Model Dev.*, 12(3):1029–1066.
- Garratt, J. (1992). *The Atmospheric Boundary Layer*. Cambridge Univ. Press, Cambridge, U.K.
- Hess, G. D., Colman, R. A., and McAvaney, B. J. (1995). On computing screen temperatures, humidities and anemometer-height winds in large-scale models. *Aust. Met. Mag.*, 44:139–145.
- Hourdin, F., Musat, I., Bony, S., Braconnot, P., Codron, F., Dufresne, J.-L., Fairhead, L., Filiberti, M.-A., Friedlingstein, P., Grandpeix, J.-Y., Krinner, G., LeVan, P., Li, Z.-X., and Lott, F. (2006). The lmdz4 general circulation model: climate performance and sensitivity to parametrized physics with emphasis on tropical convection. *Clim. Dyn.*, 27(7-8):787–813.
- Kaimal, J. C. and Finnigan, J. J. (1994). *Atmospheric Boundary Layer Flows: Their Structure and Measurement*. Oxford University Press.
- King, J. C., Connolley, W. M., and Derbyshire, S. H. (2001). Sensitivity of modelled antarctic climate to surface and boundary-layer flux parametrizations. *Quarterly Journal of the Royal Meteorological Society*, 127(573):779–794.
- Kingsolver, J. G., Woods, H. A., Buckley, L. B., Potter, K. A., MacLean, H. J., and Higgins, J. K. (2011). Complex Life Cycles and the Responses of Insects to Climate Change. *Integrative and Comparative Biology*, 51(5):719–732.
- Large, W. (2006). *Surface Fluxes for Practitioners of Global Ocean Data Assimilation*, pages 229–270. Springer Netherlands, Dordrecht.
- Large, W. G. and Yeager, S. (2004). Diurnal to decadal global forcing for ocean and sea-ice models: The data sets and flux climatologies. *University Corporation for Atmospheric Research*, No. NCAR/TN-460+STR.
- Louis, J.-F. (1979). A parametric model of vertical eddy fluxes in the atmosphere. *Bound.-Lay. Meteorol.*, 17:187–202.
- Louis, J.-F., Tiedtke, M., and Geleyn, J.-F. (1982). A short history of the pbl parameterization at ecmwf. *Workshop on Planetary Boundary Layer parameterization, 25-27 November 1981*, pages 59–79.
- Mahrt, L., Ek, M., Kim, J., and Holtslag, A. A. M. (1991). Boundary layer parameterization for a global spectral model. page 216.
- Mascart, P., Noilhan, J., and Giordani, H. (1995). A modified parameterization of flux-profile relationships in the surface layer using different roughness length values for heat and momentum. *Boundary-Layer Meteorol.*, 72(4):331–344.
- Miller, M. J., Beljaars, A. C. M., and Palmer, T. N. (1992). The sensitivity of the ecmwf model to the parameterization of evaporation from the tropical oceans. *J. Climate*, 5:418–434.
- Nakanishi, M. and Niino, H. (2006). An improved mellor-yamada level-3 model: Its numerical stability and application to a regional prediction of advection fog. *Bound.-Lay. Meteorol.*, 119(2):397–407.
- Nielsen-Gammon, J. W., Powell, C. L., Mahoney, M. J., Angevine, W. M., Senff, C., White, A., Berkowitz, C., Doran, C., and Knupp, K. (2008). Multisensor estimation of mixing heights over a coastal city. *J. Appl. Meteorol. Clim.*, 47(1):27–43.
- Raymond, C., Matthews, T., and Horton, R. M. (2020). The emergence of heat and humidity too severe for human tolerance. *Science Advances*, 6 (19).
- Rudolfs, W. (1925). Relation between temperature, humidity and activity of house mosquitoes. *Journal of the New York Entomological Society*, 33(3):163–169.
- Seidel, D. J., Angell, J. K., Christy, J., Free, M., Klein, S. A., Lanzante, J. R., Mears, C., Parker, D., Schabel, M., Spencer, R., Sterin, A., Thorne, P., and Wentz, F. (2004). Uncertainty in signals of large-scale climate variations in radiosonde and satellite upper-air temperature datasets. *Journal of Climate*, 17(11):2225–2240.

- Smith, S. R., Legler, D. M., and Verzone, K. V. (2001). Quantifying uncertainties in ncep reanalyses using high-quality research vessel observations. *J. Climate*, 14(20):4062 – 4072.
- Stéfanon, M. (2012). *Heat waves and droughts in Mediterranean: contributions of land-atmosphere coupled processes on mesoscale*. PhD thesis.
- Stull, R. (2011). Wet-bulb temperature from relative humidity and air temperature. *Journal of Applied Meteorology and Climatology*, 50(11):2267 – 2269.
- Tamiru, A., Getu, E., Jembere, B., and Bruce, T. (2012). Effect of temperature and relative humidity on the development and fecundity of *chilo partellus* (swinhoe) (lepidoptera: Crambidae). *Bulletin of Entomological Research*, 102(1):9–15.
- Ulke, A. G. (2000). New turbulent parameterization for a dispersion model in the atmospheric boundary layer. *Atmospheric Environment*, 34(7):1029–1042.
- Vecellio, D. J., Wolf, S. T., Cottle, R. M., and Kenney, W. L. (2022). Evaluating the 35°C wet-bulb temperature adaptability threshold for young, healthy subjects (psu heat project). *Journal of Applied Physiology*, 132(2):340–345. PMID: 34913738.
- Vogelezang, D. H. P. and Holtslag, A. A. M. (1996). Evaluation and model impacts of alternative boundary-layer height formulations. *Boundary-Layer Meteorol.*, 81(3):245–269.

A Linear interpolation

The function `interpolate1D` in module `phys/module_diagvar_cordex.F` linearly interpolates a value at a given position x_{pos} for a series of values $y = f(x)$, for which we have a series of positions $xvals(i)$ and its given values $yvals(i)$ for $i = [1, dx]$ the implemented methodology is explained in equation 55 and shown in figure 16.

$$\forall i = 1, dx \left\{ \begin{array}{l} ip = i + 1 \\ v_i = xvals(i) - xpos \\ v_{ip} = xvals(ip) - xpos \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \text{if } (v_i v_{ip} \leq 0.) \\ \left\{ \begin{array}{l} dx = xvals(ip) - xvals(i) \\ dy = yvals(ip) - yvals(i) \\ a = \frac{[xvals(ip)yvals(i) - xvals(i)yvals(ip)]}{dx} \\ b = \frac{dy}{dx} \\ y(x_{pos}) = a + x_{pos}b \end{array} \right. \\ \text{else} \\ \left\{ \begin{array}{l} av_i = ABS(v_i), av_{ip} = ABS(v_{ip}) \\ i_{closest} \Rightarrow \min(av_i, av_{ip}) \\ \text{if } i_{closest} == dx \left\{ \begin{array}{l} i = dx - 1 \\ ip = dx \end{array} \right. \\ dx = xvals(ip) - xvals(i) \\ dy = yvals(ip) - yvals(i) \\ a = \frac{[xvals(ip)yvals(i) - xvals(i)yvals(ip)]}{dx} \\ b = \frac{dy}{dx} \\ y(x_{pos}) = a + x_{pos}b \end{array} \right. \end{array} \right. \quad (55)$$

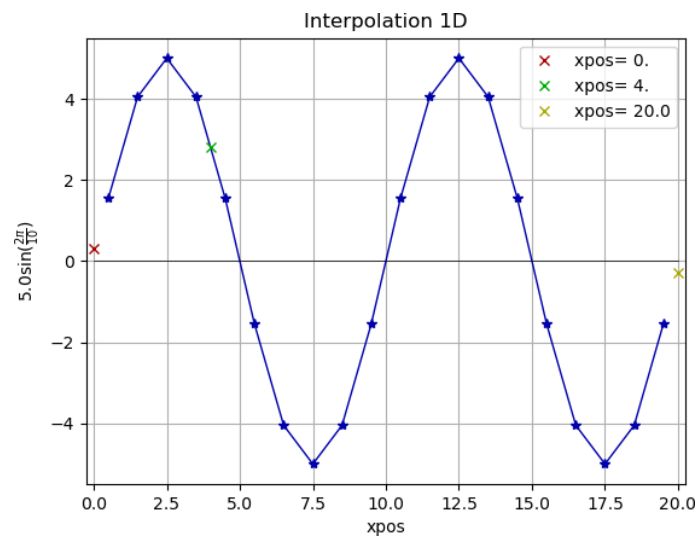


Figure 16: Example of use of the function to linearly interpolate applied to a sinusoid as $y = A \sin(\frac{2\pi}{Nvals/2})$, where $A = 5$ and $Nvals = 20$. With $xpos = [0.5, 19.5]$, interpolation at $xpos = [0, 4., 20.]$

B 2-m iterative method

Here is described in more detail, how the 2-m diagnostics is obtained

B.1 Hess 95 methodology

1. Vertically integrating the Monin-Obukhov equations for temperature, humidity and wind, one obtains the vertical profiles for each variable shown in equations 56 to 58

$$\kappa \frac{wss}{u^*} = \ln\left(\frac{z}{z_0}\right) - \Psi_M\left(\frac{z}{\mathcal{L}_0}\right) + \Psi_M\left(\frac{z_0}{\mathcal{L}_0}\right) = \mathcal{F}_M \quad (56)$$

$$\kappa \frac{\Theta - \Theta_0}{\Theta^*} = \ln\left(\frac{z}{z_H}\right) - \Psi_H\left(\frac{z}{\mathcal{L}_0}\right) + \Psi_H\left(\frac{z_H}{\mathcal{L}_0}\right) = \mathcal{F}_H \quad (57)$$

$$\kappa \frac{Q - Q_0}{Q^*} = \ln\left(\frac{z}{z_Q}\right) - \Psi_Q\left(\frac{z}{\mathcal{L}_0}\right) + \Psi_Q\left(\frac{z_Q}{\mathcal{L}_0}\right) = \mathcal{F}_Q \quad (58)$$

where wss : horizontal wind speed, $(u^*)^2 = \tau_0/\rho$: friction velocity with τ_0 : surface sharing stress, ρ : air density, $\Theta^* = -H_0/\rho C_p u^*$: temperature scale, H_0 : surface heat flux, C_p : specific heat of air at constant pressure, $Q^* = -E_0/\rho u^*$: humidity scale, E_0 : surface evaporative flux, $\mathcal{L}_0 = (u^*)^2/\kappa g/\Theta_{0v}[\Theta^*(1+0.61\Theta Q_0)+0.61\Theta_0\Theta^*]$: Obukhov length, g : gravity, $\Theta_{0v} = T_0(P_r/P_0)^{R/C_p}(1+0.61\Theta_0)$: surface virtual potential temperature, T_0 : surface temperature, P_0 : surface pressure, $P_r = 1000 \text{ hPa}$: reference pressure, R : gas constant for air, Ψ_M, Ψ_H, Ψ_Q : stability functions (empirically determined [Dyer, 1974](#)) as integrated counterparts of the vertical profiles in Monin-Obukhov equations ϕ_M, ϕ_H, ϕ_Q , z_0, z_H, z_Q : surface roughness lengths for momentum, heat and moisture, χ_0 : value of variable χ at the surface. Accordingly to [Kaimal and Finnigan \(1994\)](#) the experimental validity of these equations is found within $-2 \leq z/\mathcal{L}_0 \leq 1$. Model output at the first vertical layer is used to evaluate u^*, Θ^* and Q^* , using the β -formulation for which $Q_0 - Q(k=1) = D_w(Q_{sat0} - Q(k=1))$: differences between surface and first vertical layer, D_w : efficiency factor, Q_{sat0} : saturated value

2. Determine the value of the Monin-Obukhov length \mathcal{L}_0
3. Use equations 56 to 58 to predict a value at $z = 1.5m$ and $\Delta\Theta, \Delta Q$ (where $\Delta\chi = \chi(z = 1.5) - \chi_0$; $\chi = \Theta, Q$), using the stability functions (evaluated at $\zeta = z/\mathcal{L}_0$ and $\zeta_0 = z_0/\mathcal{L}_0$) as in equation 59

$$\Psi_\zeta = \begin{cases} \Psi_M = 2\ln(1+x) + \ln(1+x^2) - 2\tan^{-1}(x) \\ \Psi_H(\zeta) = \Psi_Q(\zeta) = 2\ln(1+x^2) \\ \Psi_M(\zeta) = \Psi_H(\zeta) = \Psi_Q(\zeta) = -5\zeta \end{cases} \begin{matrix} unstable \\ \\ stable \end{matrix} \quad (59)$$

where $x = (1 - 16\zeta)^{1/4}$ and $\zeta = z/\mathcal{L}_0$

4. compute the bulk Richardson number Ri_b (see equation 60) at the screen height ($z = 1.5$) and values from 3

$$Ri_b = \frac{g\Delta z}{\Theta_v} \frac{\Delta\Theta_v}{|\Delta\overline{wss}|^2} \quad (60)$$

where $\Delta\Theta_v$: differences in potential virtual temperature and $\Delta\overline{wss}$: differences in horizontal wind speed over the layer

5. Calculate the corrected values of $wss, \Delta\Theta, \Delta Q$ for the layer between the screen layer and the surface using the Louis equations (see 61 to 63, where it is assumed that $z_0 = z_H = z_Q$)

$$\kappa \frac{wss}{u^*} = \frac{\ln(z/z_0)}{\mathcal{F}_M^{1/2}(Ri_b, z/z_0)} \quad (61)$$

$$\kappa \frac{\Theta - \Theta_0}{\Theta^*} = \frac{\ln(z/z_0)}{\mathcal{F}_H(Ri_b, z/z_0)} \mathcal{F}_M^{1/2}(Ri_b, z/z_0) \quad (62)$$

$$\kappa \frac{Q - Q_0}{Q^*} = \frac{\ln(z/z_0)}{\mathcal{F}_Q(Ri_b, z/z_0)} \mathcal{F}_M^{1/2}(Ri_b, z/z_0) \quad (63)$$

where $\mathcal{F}_M = \mathcal{F}_H = \mathcal{F}_Q = 1/(1 + 10Ri_b(1 + 8Ri_b))$ or the most generic formulation shown in (64, equations retrieved from Mahrt et al., 1991)

$$\mathcal{F}(Ri_b, z/z_0) = \begin{cases} (1 + b'Ri_b)^{-2} & \text{stable} \\ 1 - \frac{bRi_b}{1+c|Ri_b|^{1/2}} & \text{unstable} \end{cases} \quad (64)$$

where $b = 9.4$, $b' = 4.7$, $C^* = 7.4$ and

$$c = C^* \kappa^2 b \left(\frac{z}{z_0} \right)^{(1/2)} / \left(\ln \left(\frac{z}{z_0} \right) \right)^2 \quad (65)$$

6. Steps and are repeated using the recently computed Louis profiles until a convergence criteria is meet.

B.2 LMDZ implementation

This section attempts to describe how ‘*Modèle de Circulation Générale du LMD*’ (LMDZ Hourdin et al., 2006) GCM model diagnose temperature, humidity and wind at 2 and 10 m.

It is based on Hess et al. (1995) from Dyer-Businger relations and Louis equations. All is done at the subroutine `stdlevvar` from `stdlevvar_mod.F90` module.

1. Using `cdrag` to compute \mathcal{C}_m and \mathcal{C}_h momentum and heat drag coefficients and Richardson number \mathcal{R}_i , following multiple different methodologies (see more details at section B.2.1:
 - **unstable:** (Louis, 1979) with (Mascart et al., 1995) modifications ($z_0 \neq z_{0h}$), (Louis et al., 1982) and Laurent Li
 - **stable:** (Louis, 1979) with (Mascart et al., 1995) modifications ($z_0 \neq z_{0h}$), (Louis et al., 1982) and Laurent Li, (SHARP King et al., 2001) and 1st order Monin-Obukhov theory allowing collapse of turbulence
2. Accordingly to `ok_prescr_ust` u^* is used to re-compute $Cd_m = (u^*)^2/(wss)^2$
3. Computing star variables: u^* , ta^* , qv^* and \mathcal{L}_0 (Monin-Obukhov length) as in equations 66

$$\begin{aligned} \theta &= ta|_{k=1} \left(\frac{p_{sfc}}{p|_{k=1}} \right)^{R_d/C_p} \\ u^* &= \sqrt{Cd_m wss^2} \\ \delta_z qv &= \max(qv|_{k=1}, 0) - \max(qv_{sfc}, 0) \\ \delta_z te &= \theta - t_{sfc} \\ \delta_z te &= \max(\delta_z te, \pm 1e^{-10}) \\ te^* &= \frac{Cd_h \delta_z te wss}{u^*} \\ qv^* &= \frac{Cd_h \delta_z qv wss}{u^*} \\ \mathcal{L}_0 &= \frac{(u^*)^2 \theta}{\kappa g te^*} \end{aligned} \quad (66)$$

4. First guess at 2 m (using `screenp` from `screenp_mod.F90`, see section B.2.2.i) to obtain Δwss , $\Delta \Theta$, ΔQ , resulting on wss_{ref} , θ_{ref} , ta_{ref} , qv_{ref} (eq. 67)

$$\begin{aligned} u_{zref} &= \Delta u \\ qv_{zref} &= \max(qv_{sfc}, 0.0) + \Delta qv \\ te_{zref} &= t_{sfc} + \Delta te \\ ta_{zref} &= te_{zref} \left(\frac{p_{sfc}}{p|_{k=1}} \right) - R_d/C_p \\ qv_{zref}^p &= qv_{zref} \\ ta_{zref}^p &= ta_{zref} \end{aligned} \quad (67)$$

5. corrector-iteration (niter=2, configurable from LMDZ model set-up) (using `screenc` from `screenc_mod.F90` to obtain $\Delta^{corr}u$ [at k=1], $\Delta^{corr}te$ [at sfc], $\Delta^{corr}qv$ [at sfc], see section B.2.3.i) from which wss_{ref}^{corr} , θ_{ref}^{corr} , ta_{ref}^{corr} , qv_{ref}^{corr} are recomputed (eq. 68)

$$\begin{aligned} wss_{ref}^{corr} &= \Delta^{corr}u \\ qv_{ref}^{corr} &= \Delta^{corr}qv + \max(qv_{sfc}, 0) \\ te_{ref}^{corr} &= \Delta^{corr}te + t_{sfc} \\ ta_{ref}^{corr} &= te_{ref}^{corr} \left(\frac{p_{sfc}}{p_{ref}} \right)^{-R_d/C_p} \end{aligned} \quad (68)$$

6. At the end of the iterative process, obtaining diagnostics air-temperature at 2m (tas), mixing ratio at 2m (qvs) by equation 69

$$\begin{aligned} tas &= ta_{ref}^{corr} \\ qvs &= qv_{zref}^{corr} \end{aligned} \quad (69)$$

7. repeat 4 and 5, but for diagnostic at 10 m to obtain wind speed at 10 m (wss) as well as air-temperature $taz10$ and mixing ratio $qvz10$

Equations are also corrected after Beljaars (1992) and Miller et al. (1992).

Results from the collective and collaborative workshop, 'Atelier TKE (ATKE)' with the presence of LMDZ developing team with K. Arjda, L. Raillard, C. Dehondt, P. Tiengou, A. Spiga, F. Cheruy, T Dubos, M. Coulon-Decorzens, S. Fromang, G. Riviere, A. Sima, F. Hourdin, E. Vignon are also considered. As a result of the workshop a new routine that computes turbulent Km / Kh coefficients with a 1.5 order closure scheme (TKE) with or without stationarity assumption (for dry atmosphere and horizontal homogeneity) was elaborated and implemented in LMDZ model.

Detailed description of the new implementation of the 2-m and 10-m diagnoses in LMDZ.

B.2.1 cdrag_mod.F90

Subroutine `cdrag` computes differently if it is an ocean (see sub-section B.2.1.i) or a land point (section B.2.1.ii).

First generic values are defined as shown in 70 (some of them are configurable from LMDZ model set-up)

$$\left\{ \begin{array}{ll} C_{KAP} = 0.40; C_{KAPT} = 0.42 & tol[it]_{z0} = 1.e^{-4} \\ iter. rugosity coeff. & CEPDU2 = (0.1)^2 \\ squared minimum wind & \left\{ \begin{array}{l} C_B = 5.0 \\ C_C = 5.0 \\ C_D = 5.0 \end{array} \right. \\ Louis 1982 coeff. & \left\{ \begin{array}{l} C_2 = 0.25 \\ C_3 = 0.0625 \end{array} \right. \\ King 2001 coeff. & \left\{ \begin{array}{l} BPRIME = 4.7 \\ B = 9.4 \end{array} \right. \\ Louis 1979 coeff & \alpha = 5.0 \\ Monin - Obukhov & \left\{ \begin{array}{l} C_\epsilon = 5.87; C_{inf} = 1.5; R_{ic} = 0.25; \\ pr_{asym} = 0.4; pr_{neut} = 0.8; pr_{slope} = 5.0 \\ cn = \left(\frac{1}{\sqrt{C_\epsilon}} \right)^{2/3} \\ ri_0 = \frac{2}{\pi} (C_{inf} - cn) \frac{R_{ic}}{cn} \\ ri_1 = -\frac{2}{\pi} \frac{pr_{asym} - pr_{neut}}{pr_{slope}} \end{array} \right. \\ Consistency with atke scheme & \end{array} \right. \quad (70)$$

$$\begin{aligned} R_{i,Cd}^{min} &= 0.1 \\ C_{drag}^{ter} &= 0.8; C_{drag}^{oce} = 0.8 \\ Cd_m^{max} &= 1.3e^{-3}; Cd_h^{max} = 1.3e^{-3} \end{aligned}$$

as well as performing some consistency tests 71 to remove 0 values

$$initial\ tests \begin{cases} q|_{k=1} < 0 \longrightarrow q|_{k=1} = 0 \\ q_{sfc} < 0 \longrightarrow q_{sfc} = 0 \end{cases} \quad (71)$$

and preparing the iterative methodology 72 where at the first call, the stability functions to be used will be `iflag_corr_sta=2, iflag_corr_insta=2, ok_cdrag_iter = .FALSE.` (or passed as namelist values in LMDZ configuration)

$$\begin{aligned} z0_m^{iter}(1,2) & \begin{cases} (1) = z0_m & \text{initial value} \\ (2) = 3tol[it]z0z0_m & \text{iterative momentum value} \end{cases} \\ z0_h^{iter}(1,2) & \begin{cases} (1) = z0_h & \text{initial value} \\ (2) = 3tol[it]z0z0_h & \text{iterative heat value} \end{cases} \end{aligned} \quad (72)$$

Adittional preliminary values as by 73

$$\begin{aligned} Re_{tv} &= R_v/R_d - 1.; \quad RC_{pd} = 1/C_{pd}; \quad RC_{pv} = 4R_v \\ Rv_{tmp2} &= RC_{pv}/RC_{pd} - 1 \\ z_{du}^2 &= max(CEPDU2, wss^2) \\ Pref &= \exp \left[\ln(p_{sfc}) - \frac{zg|_{k=1}}{R_d t|_{k=1} (1 + Re_{tv} max(qv|_{k=1}, 0))} \right] \\ zt_{sfcv} &= t_{sfc} (1 + Re_{tv} max(q_{sfc}, 0)) \\ zt_{vd} &= \left(t|_{k=1} + \frac{zg|_{k=1}/RC_{pd}}{1 + Rv_{tmp2} qv|_{k=1}} \right) (1 + Re_{tv} max(q|_{k=1}, 0)) \end{aligned} \quad (73)$$

B.2.1.i ocean points

First the multiple coefficients are computed using `clc_core_cp` (from `libf/phyldm/clc_core_cp.F90`) as 74 with the parameters from (Smith et al., 2001; Large and Yeager, 2004; Large, 2006), mixt calculation will be accordingly to different configuration parameters

```
choix_bulk > 1 && choix_bulk < 4
choix_bulk == 2;
  mixte = .false.
else
  mixte = .true.
```

$$\begin{aligned} Z_\chi &= zg|_{k=1}/g, \quad \chi = wss, t, q \\ logz\chi_{10} &= \ln(z_\chi/10) \\ logz\chi_{zu} &= logz\chi_{10} - logzu_{10} \\ C_{pa} &= 1004.67 \\ t_v &= ta|_{k=1} (1 + 0.608q|_{k=1}), \quad \text{virtual temperature} \\ \rho &= \frac{p_{sfc}}{287.1 ta|_{k=1} (1 + 0.61q|_{k=1})}, \quad \text{sfc. density} \\ vert.gradient & \begin{cases} \delta ta = t|_{k=1} - t_{sfc} \\ \delta qv = q|_{k=1} - q_{sfc} \end{cases} \\ L_e &= (2.501 - 0.00237(ta|_{k=1} - 273.15 - \delta ta)) 1e^6, \quad \text{latent heat} \\ wss_{first} &= max \left(\sqrt{z_{du}^2}, 5 \right) \\ Large\ and\ Yaeger\ 2006,\ 2004 & \begin{cases} \alpha = 2.7e^{-3} \ (a_1\ Smith) \\ \beta = 1.42e^{-4} \ (a_2\ Smith) \\ \gamma = 7.64e^{-5} \ (a_3\ Smith) \\ q_0 = 1.64474 \end{cases} \\ g &= 9.81, \quad \text{gravity} \end{aligned} \quad (74)$$

First approximation to z_0 (rugosity), C_d (momentum coefficient, C_d^{10} at 10 m), C_h (heat coeff.) and C_e (moisture) as by 75 to 78

$$First\ guess \quad \left\{ \begin{array}{l} mixte\ (Smith) \\ Large\ and\ Yaeger\ 2004,\ 2006 \end{array} \right. \quad \left\{ \begin{array}{l} z0_m = 1e^{-4} \\ C_d^{10} = \frac{\kappa^2}{(\ln(10/z0_m))^2} \\ C_d^{10} = \frac{\kappa^2}{wss_{first} + \beta + \gamma wss_{first}} \\ z0_m = 10 \exp\left(-\kappa/\sqrt{C_d^{10}}\right) \end{array} \right. \quad (75)$$

$$Stability \quad \left\{ \begin{array}{l} C_h^{10} = 0.018\sqrt{C_d^{10}} \\ C_h^{10} = 0.0327\sqrt{C_d^{10}} \end{array} \right. \quad \begin{array}{l} stable\ (\delta ta < 0) \\ (unstable) \end{array} \quad (76)$$

$$C_e^{10} = 3.46e^{-2}\sqrt{C_d^{10}} \quad (77)$$

$$coeffs. \quad \left\{ \begin{array}{l} C_d = C_d^{10} \\ C_h = C_h^{10} \\ C_e = C_e^{10} \end{array} \right. \quad (78)$$

Computing star coefficients χ^* (for wss , ta , qv) 79 by iterations for n_it (1, default value, but configurable in LMDZ)

$$\chi^* \quad \left\{ \begin{array}{l} u^* = wss\sqrt{C_d} \\ ta^* = \delta ta \frac{C_h}{\sqrt{C_d}} \\ qv^* = \delta qv \frac{C_e}{\sqrt{C_d}} \end{array} \right. \quad (79)$$

$$stability \quad \left\{ \begin{array}{l} \mathcal{Z}(wss, ta, qv) = (z_u, z_{ta}, z_{qv}) \frac{\kappa g}{(u^*)^2} \frac{ta^*}{ta|_{k=1}} \\ \mathcal{Z}(wss, ta, qv) = \min(\mathcal{Z}(wss, ta, qv), \pm 10) \end{array} \right. \quad (80)$$

$$wind\ speed \quad \left\{ \begin{array}{l} stable\ (\mathcal{Z}(1) > 0) \left\{ \begin{array}{l} \chi = 0.018 \\ \psi_a(1) = -5\mathcal{Z}(1) \\ \psi_b(1) = \psi_a(1) \end{array} \right. \\ unstable \left\{ \begin{array}{l} \chi = 0.0327 \\ \mathcal{X} = \sqrt{\sqrt{1 - 16\mathcal{Z}(1)}} \\ \psi_a(1) = \frac{\pi}{2} + 2 \ln\left(\frac{1+\mathcal{X}}{2}\right) + \ln\left(\frac{1+\mathcal{X}^2}{2}\right) - 2\arctan(\mathcal{X}) \\ \psi_b(1) = 2 \ln\left(\frac{1+\mathcal{X}^2}{2}\right) \end{array} \right. \end{array} \right. \quad (81)$$

$$ta\ (2),\ qv\ (3) \quad \left\{ \begin{array}{l} stable\ (\mathcal{Z}(j) > 0) \left\{ \begin{array}{l} \psi_a(j) = -5\mathcal{Z}(2) \\ \psi_b(j) = \psi_a(1) \end{array} \right. \\ unstable \left\{ \begin{array}{l} \mathcal{X} = \sqrt{\sqrt{1 - 16\mathcal{Z}(j)}} \\ \psi_a(j) = \frac{\pi}{2} + 2 \ln\left(\frac{1+\mathcal{X}}{2}\right) + \ln\left(\frac{1+\mathcal{X}^2}{2}\right) - 2\arctan(\mathcal{X}) \\ \psi_b(j) = 2 \ln\left(\frac{1+\mathcal{X}^2}{2}\right) \end{array} \right. \end{array} \right. \quad (82)$$

$$\delta ta_u = \delta ta - \frac{ta^*}{\kappa} (\log z ta_{zu} + \psi_b(1) - \psi_b(2)) \quad (83)$$

$$\delta qv_u = \delta qv - \frac{qv^*}{\kappa} (\log z qv_{zu} + \psi_b(1) - \psi_b(3)) \quad (84)$$

$$wss_N = \frac{wss_{first}}{1 + \sqrt{C_d^{10}}/\kappa (\log z u_{10} - \psi_a(1))} \quad (85)$$

$$\text{coef}s \left\{ \begin{array}{l}
\text{mixte (Smith)} \left\{ \begin{array}{l} z0_m = 0.018 \frac{(u^*)^2}{\kappa^2} + \frac{0.11 \times 14e^{-6}}{u^*} \\ C_d^{10} = \frac{g}{(\ln(10/z0_m))^2} \\ C_h^{10} = 1e^{-3} \end{array} \right. \\
\text{Large and Yaeger 2004, 2006} \left\{ \begin{array}{l} C_d^{10} = \frac{\alpha}{wss_N} + \beta + \gamma wss_N \\ z0_m = 10 \exp\left(-\kappa/\sqrt{C_d^{10}}\right) \\ C_h^{10} = \chi \sqrt{C_d^{10}} \end{array} \right. \\
C_e^{10} = 3.46e^{-2} \sqrt{C_d^{10}} \\
\phi_d = 1 + \frac{\sqrt{C_d^{10}}}{\kappa} (\log zu_{10} - \psi_a(1)) \\
\phi_h = 1 + \frac{\chi}{\kappa} (\log zu_{10} - \psi_b(1)) \\
\phi_e = 1 + \frac{3.46e^{-2}}{\kappa} (\log zu_{10} - \psi_b(1)) \\
C_d = \left(\frac{\sqrt{C_d^{10}}}{abs(\phi_d)} \right)^2 \\
C_h = \frac{C_h^{10}}{\phi_h \phi_d} \\
C_e = \frac{C_e^{10}}{\phi_e \phi_d}
\end{array} \right. \quad (86)$$

Coefficients are finally computed as 87 to 89

$$u^* = wss_{first} \sqrt{C_d} \quad (87)$$

$$ta^* = \delta ta_u \frac{C_h}{\sqrt{C_d}} \quad (88)$$

$$qv^* = \delta qv_u \frac{C_e}{\sqrt{C_d}} \quad (89)$$

To obtain the coefficients over sea, different methodologies are used:

- `choix_bulk = 4`: ECUME parametrization (Belamari, 2005; Belamari et al., 2016) and other modifications
- `choix_bulk = 5`: bulk algorithm COARE3.0 (Fairall et al., 2003) and subsequent modifications
- `choix_bulk = 1`: simplified version of the bulk algorithm COARE performed by LMDZ developers
- `default LMDZ`: $R_i = zg|_{k=1} \frac{zt_{vd} - zt_{sfv}}{z_{du}^2 zt_{vd}}$

Then the neutral coefficients are obtained:

$$\begin{aligned}
C_d^{neut} &= \frac{C_{KAP}}{\ln(1 + zg|_{k=1}/(gz0_m^{iter}(2)))} \\
C_{dm}^{neut} &= (C_d^{neut})^2 \\
C_{dh}^{neut} &= C_d^{neut} \left(\frac{C_{KAP}}{\ln(1 + zg|_{k=1}/(gz0_h^{iter}(2)))} \right)
\end{aligned} \quad (90)$$

the vertical profiles can be computed as in 91 ($R_i < 0$, unstable, selecting by `iflag_corr_insta`) and 93 ($R_i > 0$, stable, selecting via `iflag_corr_sta`) following an iterative method by `nit_bulk` (1, by default but adjustable in

LMDZ)

$$\begin{aligned}
& \left\{ \begin{array}{l} \mu = \ln(\max(z0_m/z0_h, 0.01)) \\ CM^* = 6.8741 + 2.6933\mu - 0.3601\mu^2 + 0.0154\mu^3 \\ \mathcal{P}_m = 0.5233 - 0.0815\mu + 0.0135\mu^2 - 0.001\mu^3 \\ CH^* = 3.2165 + 4.3431\mu + 0.536\mu^2 - 0.0781\mu^3 \\ \mathcal{P}_h = 0.5802 - 0.1571\mu + 0.0327\mu^2 - 0.0026\mu^3 \\ C_h = \frac{CH^* BC_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \frac{C_{KAPT}}{\ln(z0_h + zg|_{k=1}/(gz0_h))} \left(\frac{zg|_{k=1}}{gz0_h} \right)^{\mathcal{P}_h} \\ C_m = \frac{CM^* BC_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \frac{C_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \left(\frac{zg|_{k=1}}{gz0_m} \right)^{\mathcal{P}_m} \\ \mathcal{F}_m = 1 - \frac{BR_i}{1 + C_m \sqrt{ABS(R_i)}} \\ \mathcal{F}_h = 1 - \frac{BR_i}{1 + C_h \sqrt{ABS(R_i)}} \end{array} \right. \quad (91) \\
& \text{Louis 1982 (2)} \quad \left\{ \begin{array}{l} zu_{cf} = \frac{1}{1 + 3C_B C_C C_{dm}^{neut} \sqrt{abs(R_i)(1 + zg|_{k=1}/(gz0_m))}} \\ \mathcal{F}_m = \max \left((1 - 2C_B R_i zu_{cf}), R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left((1 - 3C_B R_i zu_{cf}), R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{Laurent Li (3)} \quad \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\sqrt{1 - 18R_i}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\sqrt{1 - 18R_i}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{After ATKE (6)} \quad \left\{ \begin{array}{l} s_m = \frac{2}{\pi} (C_{inf} - cn) \arctan(-R_i/r_{i0}) + cn \\ prandtl = -\frac{2}{\pi} (pr_{asym} - pr_{neut}) \arctan(R_i/r_{i1}) + pr_{neut} \\ \mathcal{F}_m = \max \left(s_m^{3/2} \sqrt{C_\epsilon} (1 - R_i/prandtl)^{1/2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\mathcal{F}_m/prandtl, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{default (Louis 1982)} \quad \left\{ \begin{array}{l} zu_{cf} = \frac{1}{1 + 3C_B C_C C_{dm}^{neut} \sqrt{abs(R_i)(1 + zg|_{k=1}/(gz0_m))}} \\ \mathcal{F}_m = \max \left(1 - 2C_B R_i zu_{cf}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(1 - 3C_B R_i zu_{cf}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{drag coeff.} \quad \left\{ \begin{array}{l} Cd_m = C_d^{neut} \mathcal{F}_m \\ Cd_h = C_{drag}^{ter} C_d^{neut} \mathcal{F}_h \\ \text{without gustiness} \left\{ \begin{array}{l} z_{cr} = \frac{0.0016}{C_{dm}^{neut} \sqrt{z_{du}^2}} abs(zt_{vd} - zt_{sfcv})^{1/3} \\ Cd_h = C_{drag}^{oce} C_{dh}^{neut} (1 + z_{cr}^{1.25})^{1/1.25} \end{array} \right. \\ Cd_m = \min(Cd_m, Cd_m^{max}) \\ Cd_h = \min(Cd_h, Cd_h^{max}) \end{array} \right. \quad (92)
\end{aligned}$$

For the stable cases previously one applies $R_i = \min(20, R_i)$

$$\begin{aligned}
& \text{Louis 1979 \& Mascart 1995 (1)} \quad \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\frac{1}{(1 + BPRIMER_i)^2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \quad (93) \\
& \text{Louis 1982 (2)} \quad \left\{ \begin{array}{l} zs_{cf} = \sqrt{1 + C_D abs(R_i)} \\ \mathcal{F}_m = \max \left(\frac{1}{1 + 2C_B R_i / zs_{cf}}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\frac{1}{1 + 3C_B R_i / zs_{cf}}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{Laurent Li (3)} \quad \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\frac{1}{1 + 10R_i(1 + 8R_i)}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \\
& \text{King 2001 (4)} \quad \left\{ \begin{array}{l} R_i < C_2/2 \left\{ \begin{array}{l} \mathcal{F}_m = \max \left((1 - R_i/C_2)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \\ \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(C_3(C_2/R_i)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \end{array} \right.
\end{aligned}$$

$$\begin{aligned}
\text{Monin - Obukhov (5)} & \quad \begin{cases} R_i < 1/\alpha \begin{cases} \mathcal{F}_m = \max \left((1 - \alpha R_i)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{cases} \\ \begin{cases} \mathcal{F}_m = \max \left(1e-7, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{cases} \end{cases} \\
\text{After ATKE (6)} & \quad \begin{cases} s_m = \max(0, cn(1 - R_i/R_{ic})) \\ prandtl = pr_{neut} + R_i pr_{slope} \\ \mathcal{F}_m = \max \left(s_m^{3/2} \sqrt{C_\epsilon} (1 - R_i/prandtl)^{1/2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\mathcal{F}_m/prandtl, R_{i,Cd}^{min} \right) \end{cases} \\
\text{Louis 1982 (default)} & \quad \begin{cases} z_{scf} = \sqrt{1 + C_{Dabs}(R_i)} \\ \mathcal{F}_m = \max \left(\frac{1}{1+2C_B R_i/z_{scf}}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\frac{1}{1+3C_B R_i z_{scf}}, R_{i,Cd}^{min} \right) \end{cases} \\
\text{drag coeff.} & \quad \begin{cases} Cd_m = Cd_m^{neut} \mathcal{F}_m \\ Cd_h = C_{drag}^{ter} Cd_h^{neut} \mathcal{F}_h \end{cases} \quad (94) \\
\text{choix_bulk} = 0 & \quad \begin{cases} Cd_m = Cd_m^{neut} \mathcal{F}_m \\ Cd_h = C_{drag}^{ter} Cd_h^{neut} \mathcal{F}_h \end{cases} \quad (95) \\
\text{ocean points} & \quad \begin{cases} Cd_h = C_{drag}^{oce} Cd_h^{neut} \mathcal{F}_h \\ Cd_m = \min(Cd_m, Cd_m^{max}) \\ Cd_h = \min(Cd_h, Cd_h^{max}) \end{cases} \quad (96) \\
\text{whilst iterating} & \quad \begin{cases} z0_m^{iter}(1) = z0_m^{iter}(2) \\ z0_h^{iter}(1) = z0_h^{iter}(2) \\ z0_m^{iter}(2) = 0.018 * Cd_m^{wss} + 0.11 \frac{14e^{-6}}{\sqrt{Cd_m z_{du}^2}} \\ \text{differentiate } z0 \begin{cases} z0_h^{iter}(2) = z0_m^{iter}(2) \\ z0_h^{iter}(2) = 0.40 \frac{14e^{-6}}{\sqrt{Cd_m z_{du}^2}} \\ z0_m^{iter}(2) = \max(1.5e^{-05}, z0_m^{iter}(2)) \\ z0_h^{iter}(2) = \max(1.5e^{-05}, z0_h^{iter}(2)) \end{cases} \end{cases} \quad \begin{aligned} & iflag_z0_oce = 0 \\ & iflag_z0_oce = 1 \end{aligned} \quad (97)
\end{aligned}$$

For which the drag coefficients over ocean become [98](#) to [101](#)

$$Cd_m = \min(Cd_m, Cd_m^{max}) \quad (98)$$

$$Cd_h = \min(Cd_h, Cd_h^{max}) \quad (99)$$

$$z0_m = z0_m^{iter}(2) \quad (100)$$

$$z0_h = z0_h^{iter}(2) \quad (101)$$

B.2.1.ii land points

For the land points the same equations as for the ocean are used, except for the final calculation of the drag and rugosity coefficients.

Then the neutral coefficients are obtained:

$$\begin{aligned}
R_i &= zg|_{k=1} \frac{zt_{vd} - zt_{sfcv}}{z_{du}^2 zt_{vd}} \quad (102) \\
C_d^{neut} &= \frac{C_{KAP}}{\ln(1 + zg|_{k=1}/(gz0_m))} \\
C_{dm}^{neut} &= (C_d^{neut})^2 \\
C_{dh}^{neut} &= C_d^{neut} \left(\frac{C_{KAP}}{\ln(1 + zg|_{k=1}/(gz0_h))} \right)
\end{aligned}$$

the vertical profiles can be computed as in [103](#) ($R_i < 0$, unstable, selecting by `iflag_corr_insta`) and [105](#) ($R_i > 0$, stable, selecting via `iflag_corr_sta`)

$$\begin{aligned}
& \begin{array}{l} \text{Louis 1979} \\ \& \text{Mascart 1995} \\ (1) \end{array} \left\{ \begin{array}{l} \mu = \ln(\max(z0_m/z0_h, 0.01)) \\ CM^* = 6.8741 + 2.6933\mu - 0.3601\mu^2 + 0.0154\mu^3 \\ \mathcal{P}_m = 0.5233 - 0.0815\mu + 0.0135\mu^2 - 0.001\mu^3 \\ CH^* = 3.2165 + 4.3431\mu + 0.536\mu^2 - 0.0781\mu^3 \\ \mathcal{P}_h = 0.5802 - 0.1571\mu + 0.0327\mu^2 - 0.0026\mu^3 \\ C_h = \frac{CH^* BC_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \frac{C_{KAPT}}{\ln(z0_h + zg|_{k=1}/(gz0_h))} \left(\frac{zg|_{k=1}}{gz0_h} \right)^{\mathcal{P}_h} \\ C_m = \frac{CM^* BC_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \frac{C_{KAP}}{\ln(z0_m + zg|_{k=1}/(gz0_m))} \left(\frac{zg|_{k=1}}{gz0_m} \right)^{\mathcal{P}_m} \\ \mathcal{F}_m = 1 - \frac{BR_i}{1 + C_m \sqrt{ABS(R_i)}} \\ \mathcal{F}_h = 1 - \frac{BR_i}{1 + C_h \sqrt{ABS(R_i)}} \end{array} \right. \quad (103) \\
& \text{Louis 1982 (2)} \left\{ \begin{array}{l} zu_{cf} = \frac{1}{1 + 3C_B C_C C_{dm}^{neut} \sqrt{abs(R_i)(1 + zg|_{k=1}/(gz0_m))}} \\ \mathcal{F}_m = \max \left((1 - 2C_B R_i zu_{cf}), R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left((1 - 3C_B R_i zu_{cf}), R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{Laurent Li (3)} \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\sqrt{1 - 18R_i}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\sqrt{1 - 18R_i}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{After ATKE (6)} \left\{ \begin{array}{l} s_m = \frac{2}{\pi} (C_{inf} - cn) \arctan(-R_i/r_{i0}) + cn \\ prandtl = -\frac{2}{\pi} (pr_{asym} - pr_{neut}) \arctan(R_i/r_{i1}) + pr_{neut} \\ \mathcal{F}_m = \max \left(s_m^{3/2} \sqrt{C_\epsilon} (1 - R_i/prandtl)^{1/2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\mathcal{F}_m/prandtl, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{default (Louis 1982)} \left\{ \begin{array}{l} zu_{cf} = \frac{1}{1 + 3C_B C_C C_{dm}^{neut} \sqrt{abs(R_i)(1 + zg|_{k=1}/(gz0_m))}} \\ \mathcal{F}_m = \max \left(1 - 2C_B R_i zu_{cf}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(1 - 3C_B R_i zu_{cf}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{drag coeff.} \left\{ \begin{array}{l} Cd_m = C_d^{neut} \mathcal{F}_m \\ Cd_h = C_{drag}^{ter} C_d^{neut} \mathcal{F}_h \end{array} \right. \quad (104)
\end{aligned}$$

For the stable cases previously one applies $R_i = \min(20, R_i)$

$$\begin{aligned}
& \text{Louis 1979 \& Mascart 1995 (1)} \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\frac{1}{(1 + BPRIMER_i)^2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \quad (105) \\
& \text{Louis 1982 (2)} \left\{ \begin{array}{l} zs_{cf} = \sqrt{1 + C_D abs(R_i)} \\ \mathcal{F}_m = \max \left(\frac{1}{1 + 2C_B R_i / zs_{cf}}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\frac{1}{1 + 3C_B R_i / zs_{cf}}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{Laurent Li (3)} \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(\frac{1}{1 + 10R_i(1 + 8R_i)}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \\
& \text{King 2001 (4)} \left\{ \begin{array}{l} R_i < C_2/2 \left\{ \begin{array}{l} \mathcal{F}_m = \max \left((1 - R_i/C_2)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \\ \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(C_3(C_2/R_i)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \end{array} \right.
\end{aligned}$$

$$\begin{aligned}
& \text{Monin - Obukhov (5)} \quad \left\{ \begin{array}{l} R_i < 1/\alpha \left\{ \begin{array}{l} \mathcal{F}_m = \max \left((1 - \alpha R_i)^2, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \\ \left\{ \begin{array}{l} \mathcal{F}_m = \max \left(1e-7, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \mathcal{F}_m \end{array} \right. \end{array} \right. \\
& \text{After ATKE (6)} \quad \left\{ \begin{array}{l} s_m = \max(0, cn(1 - R_i/R_{ic})) \\ prandtl = pr_{neut} + R_i pr_{slope} \\ \mathcal{F}_m = \max \left(s_m^{3/2} \sqrt{C_\epsilon} (1 - R_i/prandtl)^{1/2}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\mathcal{F}_m/prandtl, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{Louis 1982 (default)} \quad \left\{ \begin{array}{l} z_{scf} = \sqrt{1 + C_D abs(R_i)} \\ \mathcal{F}_m = \max \left(\frac{1}{1+2C_B R_i/z_{scf}}, R_{i,Cd}^{min} \right) \\ \mathcal{F}_h = \max \left(\frac{1}{1+3C_B R_i/z_{scf}}, R_{i,Cd}^{min} \right) \end{array} \right. \\
& \text{drag coeff.} \quad \left\{ \begin{array}{l} Cd_m = Cd_m^{neut} \mathcal{F}_m \\ Cd_h = C_{drag}^{ter} Cd_h^{neut} \mathcal{F}_h \end{array} \right. \quad (106)
\end{aligned}$$

B.2.2 screenp_mod

B.2.2.i screenp

To get the different $\Delta\chi$ depends on the stability of the boundary layer as it is shown in equations 107, 108

$$\begin{aligned}
& \text{Stable } (\mathcal{L}_0 \geq 0) \quad \left\{ \begin{array}{l} wss > 1.5 \\ \mathcal{L}_0 \leq 1.0 \\ z_{0m} \leq 1.0 \end{array} \right\} \left\{ \begin{array}{l} \Delta u = \frac{u^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) + \min(5, 5(z_{ref} - z_{0m})/\mathcal{L}_0) \right] \\ \Delta te = \frac{te^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) + \min(5, 5(z_{ref} - z_{0m})/\mathcal{L}_0) \right] \\ \Delta qv = \frac{qv^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) + \min(5, 5(z_{ref} - z_{0m})/\mathcal{L}_0) \right] \end{array} \right. \quad (107) \\
& \left\{ \begin{array}{l} \Delta u = 0.1 wss|_{k=1} \\ \Delta te = 0.1 (ta|_{k=1} - t_{sfc}) \\ \Delta qv = 0.1 (\max(qv|_{k=1}, 0) - \max(qv_{sfc}, 0)) \end{array} \right. \\
& \text{Unstable} \quad \left\{ \begin{array}{l} wss > 5 \\ abs(\mathcal{L}_0) \leq 50 \end{array} \right\} \left\{ \begin{array}{l} x_{tmp} = \left(1 - 16 \frac{z_{ref}}{\mathcal{L}_0} \right)^{1/4} \\ x_{tmp0} = \left(1 - 16 \frac{z_{0m}}{\mathcal{L}_0} \right)^{1/4} \\ \Delta u \frac{u^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) - 2 \ln(0.5(1 + x_{tmp})) + \right. \\ \quad \left. 2 \ln(0.5(1 + x_{tmp0})) - \ln(0.5(1 + x_{tmp}^2)) + \right. \\ \quad \left. \ln(0.5(1 + x_{tmp0}^2)) + 2 \arctan(x_{tmp}) - 2 \arctan(x_{tmp0}) \right] \\ \Delta te \frac{te^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) - 2 \ln(0.5(1 + x_{tmp}^2)) + \right. \\ \quad \left. 2 \ln(0.5(1 + x_{tmp0}^2)) \right] \\ \Delta qv \frac{qv^*}{\kappa} \left[\ln \left(\frac{z_{ref}}{z_{0m}} + 1 \right) - 2 \ln(0.5(1 + x_{tmp}^2)) + \right. \\ \quad \left. 2 \ln(0.5(1 + x_{tmp0}^2)) \right] \end{array} \right. \quad (108) \\
& \left\{ \begin{array}{l} \Delta y = 0.5 wss \\ \Delta te = 0.5 (ta|_{k=1} - t_{sfc}) \\ \Delta qv = 0.5 (\max(qv|_{k=1}, 0) - \max(qv_{sfc}, 0)) \end{array} \right.
\end{aligned}$$

B.2.3 screenc_mod.F90

B.2.3.i screenc

First the drag coefficients Cd_m^{ref} and Cd_h^{ref} are recomputed using te reference values $wss_{ref}, ta_{ref}, qv_{ref}, z_{ref}$ using `cdrag` subroutine (see B.2.1). With the new coefficients compute de new Δ^{corr} (see eq. 109) `ok_prescr_ust` flag to

force the use of u^*

$$\begin{aligned}
 \Delta^{corr}u & \quad \left\{ \begin{array}{l} ok_prescr_ust \\ \Delta u = \frac{u^*}{\sqrt{Cd_m^{ref}}} \end{array} \right\} \quad \left\{ \begin{array}{l} Cd_m = \frac{(ust^*)^2}{1+ws^2} \\ \Delta u = \frac{u^*}{\sqrt{Cd_m}} \end{array} \right. \\
 \Delta^{corr}te & = \frac{te^* \sqrt{Cd_m^{ref}}}{Cd_h^{ref}} \\
 \Delta^{corr}qv & = \frac{qv^* \sqrt{Cd_m^{ref}}}{Cd_h}
 \end{aligned} \tag{109}$$