CONSORTIUM FOR SMALL SCALE MODELING

COSMO

PT AEVUS2

Progresses and problems

PROTEZIONE CIVILE
Presidenza del Consiglio dei Ministri
Dipartimento della Protezione Civile

Valeria Garbero
Massimo Milelli

PIEMONTE
Arpa
Agenzia Regionale per la Protezione Ambientale

Sistema Nazionale per la Protezione dell'Ambiente

28.05.2020

State of the art

Matthias files are fine (no more miss) but:

- the names of the new variables have to be changed according to DWD definitions (general remark)
- files are in netcdf4 and not recognised by int2lm (our problem only, it could be a question of libraries in compilation)

Once we run int2lm, in order to use lurb_urbfr=T, lurb_h2w=T and lurb_urbh=T we must use 1 cpu otherwise the .nc external file is not read correctly (known bug).

State of the art

The output of int2lm (laf file) in netcdf contains many miss that were not present originally, therefore COSMO crashes when trying to read it.

The output of int2lm (laf file) in grib contains NaN in U, V, PP and T fields, in the lowest level(s) close to the soil, therefore COSMO crashes when trying to read it.

The same thing happens when lurb_*=F in int2lm (and still using 1 cpu).

BUT, having lurb_*=F and using more cpus, the laf is good, both in netcdf or in grib.

State of the art

The modifications to the grib_definition_file are fine, now the variables have their correct codification in grib1/grib2.

Urban double-counting

Rscript calculates external parameters based on Globcover land use classes, as ExtPar routines do, but excluding the urban contribution in order to avoid the urban double-counting.

Rscript then modifies those grid characterized by URBAN >0 & FR_LAND >0.5

Bug in Rscript -> FIXED: I will provide the correct Rscript to Matthias!

An accurate verification has been conducted by modifying the Rscript so that it calculates the external parameters as ExtPar, namely without excluding the land use class.

The re-calculated external parameters are: PLCOV_MX, PLCOV_MN, ROOTDP, LAI_MX, LAI_MN, FOR_E, FOR_D, EMIS_RAD, z0, SKC.

The external parameter fields derived from ExtPar and from my Rscript are compared.

Urban double-counting: verification

field_$_{extPar}$-field_$_{Rscript}$ has been computed for each parameter and the maximum absolute value is calculated:

0.01241099 for PLCOV_MX -> to investigate

0.009590692 for PLCOV_MN -> to investigate

5.048157e-08 for ROOTDP -> negligible

2.785492e-07 for LAI_MX -> negligible

1.048714e-07 for LAI_MN -> negligible

5.364418e-08 for FOR_D -> negligible

0.0052257754376448 for FOR_E -> negligible

0.06576184 for z0 -> to investigate

7.82311e-08 for EMIS_RAD -> negligible

3.921565 for SKC -> to investigate

Urban double-counting: verification

The differences match the grid characterized by FR_LAND ~ 0.5, and they are due to inconsistencies between FR_LAND and the land use classes.

[Ex. lu[21]=FR_LAKE=0.499982 AND FR_LAND=0.51 -> **TOTAL ≠ 1**

It could be that FR_LAKE is assumed 0.49 and then FR_LAND=0.51!]

-> see QGIS

Solution could be to modify those grid characterized by **URBAN >0 & FR_LAND >**0.51

Rscript that calculates the external parameters as ExtPar are verified

Urban double-counting ??

The script to avoid double-counting is built in such a way as to perform the weighted sums excluding land use class 19 (urban).

So it is applied to grid characterized by URBAN >0 & FR_LAND>0.5

What about URBAN=1 and ISA <1?

What about albedo?