Urbanized WRF modeling using WUDAPT

Web version: March 2016

by

Alberto Martilli, Oscar Brousse, Jason Ching

1. Introduction:

An important objective of WUDAPT is to provide model parameter inputs to environmental models so as to be able to study and perform surface energy budget, weather, climate and air quality model applications for urban areas and anywhere in the world in consistent manner. This document focus on documentation of the initial and successful deployment of WUDAPT into an urbanized version of a modeling system called "Weather Research and Forecasting" or WRF. WRF (Skamarock et al., 2008) is a community and science-based atmospheric modeling system in widespread use (currently ~ 20,000 code downloads) throughout the world. This document, in combination with the WRF Users Guide and Tutorials (www2.mmm.ucar.edu), provides prospective users, the information and detailed instructions to enable the running of both urban canopy multi-layer versions in WRF (BEP by Martilli et al., 2002, and BEP-BEM Salamanca et al., (2010), Salamanca and Martilli (2010) and Salamanca et al., 2011) by using the accompanying form and function data tables of WUDAPT level "0" data based on the Local Climate Zone (LCZ) classification scheme by Stewart and Oke, (2012) and & Stewart, Oke and Krayenhoff (2014). Figure 1 is a summary of the methodology; an initial implementation for Madrid is in Brousse et al. (2016).

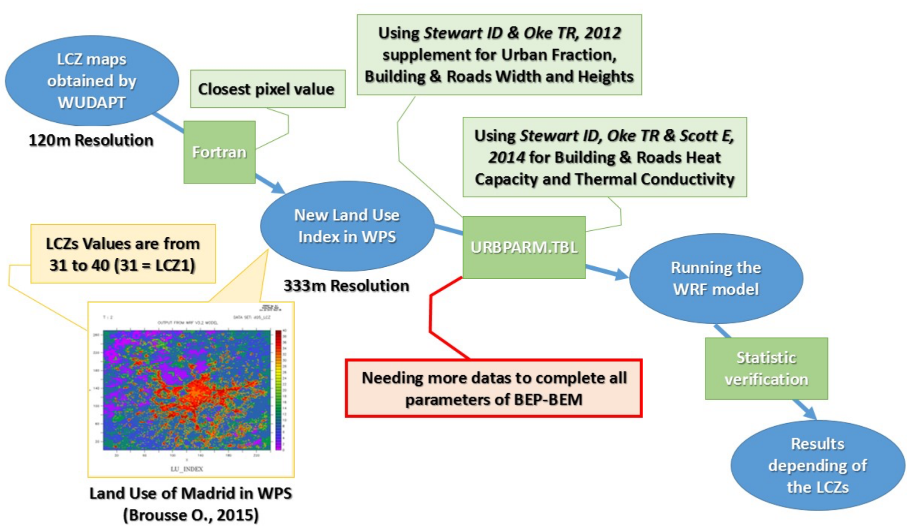


Figure 1. Scheme for using WUDAPT in multi-layer urban canopy versions of WRF.

The philosophy adopted in order to use level 0 WUDAPT as input for BEP and BEP-BEM in WRF, is based on the following three steps:

1) Extension of the number of urban classes from 3 (the actual default value) to 10 (the 10 Local (Urban Climate Zones of Stewart and Oke 2012).
2) Preparation, based on the LCZ maps created by WUDAPT level 0, of a modified landuse field (LU_INDEX) that merges the WUDAPT information (in urban areas), with the default information (or information coming from other sources) in non-urban areas. For the initial implementation, the median value of each urban canopy parameter for each attending LCZ class was utilized and introduced into the URBPARM table before running WRF "Real" applications. The user is encouraged to introduce alternative values other than the median, if justified available, or alternatively, unique WUDAPT level "1" and "2" values when they become available.
3) Modification of the URBPARM.TBL table to define, for each urban class, several supplementary parameters not currently in the WUDAPT "lookup tables for each LCZ. These include: urban fraction, building heights and road width and are from Stewart and Oke (2012), the heat capacity and thermal conductivity of building and roads are from Stewart, Oke and Krayenhoff (2014).

2. Required modifications to WRF –Urban (BEP and BEP_BEM Options using WUDAPT Level"0"

Below is a description of the modifications that have to be done in the code WRF in order to extend the number of urban classes from 3 to 10 (step 1 above).
In the standard version of WRF there is a series of "if statements" that assign landuse classes 31,32, and 33 to urban classes (internal in the code) 3, 2 and 1. These are the statements that have to be modified. The "new" 10 urban classes correspond to landuse classes 31-40. Below is the list of modules to be modified, all in the physics directory:
module_sf_urban.F
Replace the lines:

```
IF( IVGTYP(I,J) == 31) THEN
   UTYPE_URB2D(I,J) = 3  ! low-intensity residential
    UTYPE_URB = UTYPE_URB2D(I,J)  ! low-intensity residential
    IF (FRC_URB2D(I,J) == 0.) THEN
      FRC_URB2D(I,J) = FRC_URB_TBL(UTYPE_URB)
    ELSE
      CONTINUE
    ENDIF
  ENDIF
  IF( IVGTYP(I,J) == 32) THEN
    UTYPE_URB2D(I,J) = 2  ! high-intensity
```

```
            UTYPE_URB = UTYPE_URB2D(I,J)  ! high-intensity
            IF (FRC_URB2D(I,J) == 0.) THEN
              FRC_URB2D(I,J) = FRC_URB_TBL(UTYPE_URB)
            ELSE
              CONTINUE
            ENDIF
          ENDIF
          IF( IVGTYP(I,J) == 33) THEN
            UTYPE_URB2D(I,J) = 1  !  Commercial/Industrial/Transportation
            UTYPE_URB = UTYPE_URB2D(I,J)  !  Commercial/Industrial/Transportation
```

With the lines.

```
          IF( IVGTYP(I,J) .ge. 31) THEN
            UTYPE_URB2D(I,J) = ivgtyp(i,j)-30  !
            UTYPE_URB = UTYPE_URB2D(I,J)  !
```

Moreover, in the same module, adapt the value assigned to the points where IVGTYP(I,J) == ISURBAN (the points classified as urban in the default WRF landuse not overwritten by the WUDAPT database – usually these points are few, but it can happen that the default WRF landuse classifies some points as urban that are not detected as urban by WUDAPT):

```
         IF( IVGTYP(I,J) == ISURBAN)  THEN
           UTYPE_URB2D(I,J) = 2  ! for default. – compact mid-rise
           UTYPE_URB = UTYPE_URB2D(I,J)  !
           FRC_URB2D(I,J) = 0.
         ENDIF
```

module_surface_driver.F

Replace lines

```
        IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) .gt. 31 .or. &  !urban
           IVGTYP(I,J) == 32 .or. IVGTYP(I,J) == 33 ) THEN !urban
```

With lines

```
        IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) .ge. 31) then    !urban
```

This has to be done in two places in the file.

module_sf_noahdrv.F

Similarly, replace lines

```
            IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) == 31 .or. &
             IVGTYP(I,J) == 32 .or. IVGTYP(I,J) == 33) THEN
```

with

```
            IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) .ge. 31 ) THEN
```

This has to be done in three locations.

module_sf_bep.F and *module_sf_bep_bem.F*

Change the value of parameter nurbm from 3 to 10.

Final thing is to modify the value of flag num_land_cat to 40 in the namelist.input file.

```
num_land_cat                    = 40
```

In addition, the LANDUSE.TBL file should be modified to add 40 classes to the approach used (USGS, or MODIS). The actual value given to the different urban classes in the file LANDUSE.TBL is not important because it will not be used, but it is necessary to have the same number of classes indicated in num_land_cat (40).

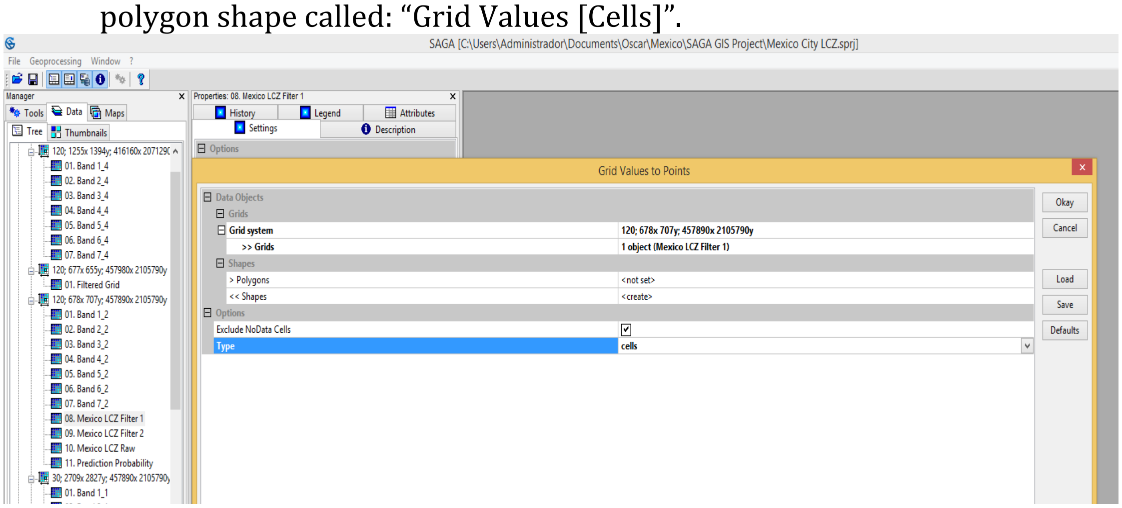
Save and compile the code**.**

3. Defining a domain in WPS, based on the ROI

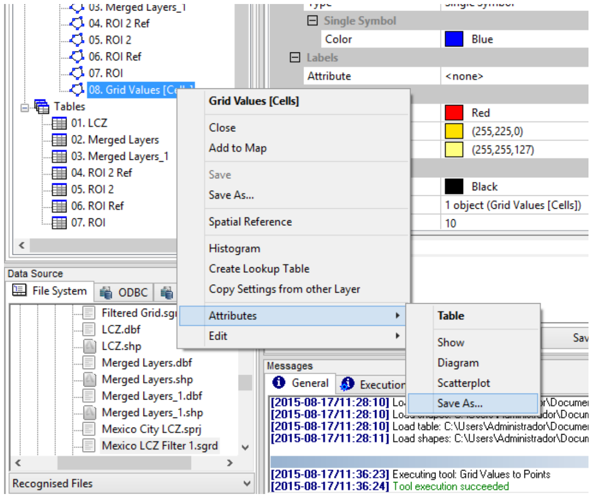
In this section we describe how the LCZ map produced for the ROI with SAGA GIS can be saved in an appropriate format, and transformed in an input for WPS (the preprocessor of WRF).

3.1 Extract the Value Table you obtained with the *Random Forest Classification* in SAGA GIS for the ROI in a text format that can be read by the Fortran program that will prepare the inputs for the WPS. This can be done as follows:

- Using the *Grid Values to Points* tool, implement your *Random Forest Classification* grid in the “Grids” folder. Then just change the “Type” to *cells* without changing anything else. You obtain a polygon shape called: “Grid Values [Cells]”.



- Extract it as a [*name of the city*].txt file of your city by right clicking on it, go to Attributes / Save As…



- Put this text file in the folder where the rd_wr_binary.f90 code is.

3.2 The WRF preprocessor WPS needs, in input, data distributed on a regular array, with a constant distance (in degrees) in the latitudinal and longitudinal direction. The Fortran program *rd_wr_binary.f90* reads the text file created in the previous step, and projects/interpolates the data on a regular grid as needed by WPS, and it also outputs the values in the binary format needed by WPS. To do this, some parameters in the *rd_wr_binary.f90* Fortran program need to be adapted to your case. In particular:

- The number of points in the text file of the ROI, np=*[number_of_points_extracted]* this is equal to the number of lines of the "[name of the city].txt" minus 1 (because there is an header in the file).
- The parameters of the regular grid where the data will be interpolated. These are: nx=*[number_of_X_points]*, ny=*[number_of_Y_points]*, and the coordinates of the grid: *latmin, longmin, latmax & longmax*. These values are in WGS 84 system, the default projection of WRF. Even if the user has the freedom to choose the domain he/she wants, what we usually do is to define this grid as close as possible to the finer domain of the WRF simulation, and to fix the values of nx and ny equal to those of the WRF simulation, so that the resolution of the output file (the regular grid) will be close to the one used by WRF (in the finer domain). However, other options can be chosen. It is important to remember that the program will assign a "missing value" to all the points that are not urban, or that are outside of the ROI.
- Then change the "igsize" value depending on the interpolation you want to process, either closest value, or majority. The resolution of the file produced by SAGA gis is in general 120m, so that if your resolution of

WRF is much coarser than this, it is more appropriate to choose the majority.

If you have extracted your Value Table in UTM, please follow these steps. If not, jump to the next point

- Change the iutm value to 1
- Change the iizone value in reference to the UTM zone where your domain is (i.e. Spain = 30, France = 31, Mexico = 14, etc…). This online map could help you : http://www.dmap.co.uk/utmworld.htm

The next point

- Change the name of file=***'[name of the city].txt'***
- Save your changes
- Compile with ./make.
- Run typing ./rd_wr_binary.exe
- When you run the executable the dx and dy values (distance in degrees in the longitudinal and latitudinal direction) will appear on the screen. Save those values.
- The program will create a binary file called *landuse_urban*. Copy this file in a specific folder (that you can call for example *landuse_[name of the city])* and rename it as *00001-[number_of_X_points].00001-[number_of_Y_points]* For example if nx=240, and ny=270, then it would be : 00001-00240.00001-00270
- In the same folder place the *index* file with the description of the field. You can copy it from another landuse directory (for example landuse_30s in the *geog* folder of WPS), and then modify the different entries following the instructions given below:

```
type=categorical
category_min=31
category_max=40
  projection=regular_ll
  missing_value=0.
  dx=          # <- edit. This is the value given by the rd_wr_binary.exe above
  dy=          # <- edit. This is the value given by the rd_wr_binary.exe above
  known_x=1.0
  known_y=1.0
  known_lat =          # <- edit. This is latitude of the SW corner
  known_lon =          # <- edit. This is longitude of the SW corner
  wordsize = 1
  tile_x=          # <- edit. This is the number of points in x (nx of the Fortran code)
  tile_y=          # <- edit. This is the number of points in x (nx of the Fortran code)
  tile_z=1
```

```
  units="category"
 description="10-category UCZ "
```

In the *WPS* folder

- Leave *landuse_[name of the city]* and enter in the *geogrid* folder in WPS.
- Once in *geogrid* open the *GEOGRID.TBL* and add the lines

```
===============================
name=LANDUSEF
        priority=2
        dest_type=categorical
        z_dim_name=land_cat
        interp_option =   default:nearest_neighbor
        abs_path = default: [full path to the landuse_[name_of_the_city] folder]
```

In this way the WPS will modify the default landuse in one point *if, and only if*, there is a value different than zero in the new landuse file prepared based on the LCZ, and it will assign a value equal to 30+number of UCZ class (UCZ classes go from 1, compact high rise, to 10). So the new UCZ classes will have numbers form 31 to 40 in the landuse field called lu_index.

- Leave *geogrid*
- Run ./geogrid.exe
- Verify the results looking at field lu_index in geo_em.d0*.nc. If you use the graphical package "ferret" you can tape *fill lu_index*
- Once you have the geo_em files, you can run the metgrid.exe executable to obtain the met files needed to run WRF.

References

Brousse O, Martilli A, Foley M, Mills G, Bechtel B, 2016: WUDAPT, an efficient land use producing data tool for mesoscale models: Integration of urban LCZ in WRF over Madrid. Submitted to Urban Climate.

Martilli A., A. Clappier, and M Rotach, 2002: An urban surface exchange parameterizations for mesoscale models, Bound Lay Meteorol (104) 261-304

Salamanca F., A. Krpo, A. Martilli, A Clappier, 2010: A new building energy model coupled with an urban canopy parameterization for urban climate simulations – Part I Formulation, verification and sensitivity analyses of the mode. Theor Appl Climatol (99) 331-344.

Salamanca F., and A. Martilli, 2010: A new building energy model coupled with an urban canopy parameterization for urban climate simulations – Part II validation with one-dimensional off-line simulations. Theor Appl Climatol (99) 335-356.

Salamanca F., A. Martilli, M. Tewari, and F. Chen, 2011: A study of the urban boundary layer using different urban parameterizations and high resolution urban canopy parameters with WRF, J Appl Meteor and Clim (50) 1107-1128.

Skamarock, W., Klemp, J Dudhia, D Gill, D Barker, M. Duda, X Huang, J Powers, 2008: A description of the advanced research WRF version 3, NCAR Tech Note. Note NCAR/TN-475+STR, 125 pp., NCAR, Boulder, CO.

Stewart, I, T. Oke, 2012: Local climate zones for urban temperature studies, Bull Am Meteorol Soc., (93) 1879-1900.

Stewart, I, T. Oke, and E. S. Krayenhoff, 2014: Evaluation of the 'local climate zone' scheme using temperature observations and model simulations, Int'l. J. Climatol (34) 1062-1080.