Guidelines for setting up high-resolution urban simulations over Sydney using WRF

Author: Annette L. Hirsch (annette.l.hirsch@gmail.com)

Version: 1.0

Disclaimer: This guideline uses tools from a variety of sources. Where possible, versions of the software have been noted and acknowledgement of the original author provided. Instances where further code modifications are necessary due to new versions of WRF should be considered.

Assumed knowledge:

- Setup and running WRF on HPC
- The Australian National Computational Infrastructure (NCI)

Files: https://github.com/annettehirsch/hirsch_wrf-urban.git

How to Acknowledge:

The paper presenting the results created using this guideline:

> Hirsch, A. L., J. P. Evans, C. Thomas, B. Conroy, M. A. Hart, M. Lipson, W. Ertler (in review) Resolving the influence of local flows on urban heat amplification during heatwaves. Submitted to Environmental Research Letters.

Other papers you should consult and appropriately acknowledge include:

- Brousse, O., Martilli, A., Foley, M., Mills, G., and Bechtel, B. (2016). WUDAPT, an efficient land use producing data tool for mesoscale models? Integration of urban LCZ in WRF over Madrid. Urban Climate, 17:116-134, doi:10.1016/j.uclim.2016.04.001.
- Lindqvist, J., B. Conroy, M. A. Hart, A. Maharaj, and A. L. Hirsch (in preparation). Classifying land-use in Sydney, Australia for understanding urban climate.
- Stewart, I. D., and Oke, T. R. (2012). Local climate zones for urban temperature studies. Bulletin of the American Meteorological Society, 1879-1900, doi:10.1175/BAMS-D-11-00019.1.
- Su, C.-H., Eizenberg, N., Steinle, P., Jakob, D., Fox-Hughes, P., White, C. J. (2019). BARRA v1.0: the Bureau of Meteorology Atmospheric high-resolution Regional Reanalysis for Australia. Geoscientific Model Development, 12:2049-2068, doi:10.5194/gmd-12-2049-2019.

Contents:

1. WRF code modifications to use 10 class urban

This guideline uses WRF v4.1.3.

See http://climate-cms.wikis.unsw.edu.au/WRF_v4.1.3_installation and https://github.com/coecms/WRF/tree/V4.1.3 for code installation and running at NCI

The following guidelines are based off of those provided by http://www.wudapt.org/wudapt-to-wrf/ last accessed February 2021.

- First git clone the WRF code to your working area at NCI, ensure that you have joined project sx70 where WRF shared data is installed.
- The **`LANDUSE.TBL`** and **`URBPARM.TBL`** files stored in the run directory (e.g. WRF/WRFV3/run) need to be updated. The versions I used are available at the git repo provided at the top of this guideline.
  - For **`LANDUSE.TBL`** update the sections pertaining to **`MODIFIED_IGBP_MODIS_NOAH`** such that the number of land use classes is 40 instead of 30 and add the 10 urban classes to both SUMMER and WINTER sections as follows:

```
31,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Compact high-rise'
32,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Compact midrise'
33,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Compact low-rise'
34,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Open high-rise'
35,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Open midrise'
36,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Open low-rise'
37,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Lightweight low-rise'
38,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Low low-rise'
39,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Sparsely built'
40,  10.,  .10,  .97,  80.,  3.,  1.67, 18.9e5, 'Heavy industry'
```

    The parameter values do not matter as these are read and used in **`URBPARM.TBL`**

  - For **`URBPARM.TBL`** the user needs to update the parameters to have 10 urban classes rather than the default of 3. Best to use the file provided in the git repo. Values for the main parameters for Sydney are as follows:

Table 1: Parameter estimates for each urban class

LCZ	description	building Height [m] median (std dev)	albedo [-]	vegetation fraction [%]	anthropogenic heat [W m-2]
1	compact high-rise	80 (67.58)	0.107	3.7	54
2	compact mid-rise	16 (20.96)	0.134	17.8	20.1
3	compact low-rise	8.5 (0.59)	0.146	25.9	12.4
4	open high-rise	35 (44.46)	0.134	8.9	54
5	open mid-rise	16 (20.96)	0.134	8.9	20.1
6	open low-rise	9 (0.54)	0.148	32.7	10.5
7	lightweight low-rise	4 (1.32)	0.16	60.4	5.4
8	large low-rise	9 (2.18)	0.158	22.4	11.7
9	sparsely built	10 (0.60)	0.156	73.1	3.6
10	heavy industry	11 (2.47)	0.161	23.4	63.9

- The following WRF modules need to be amended to remove the hard-coded urban types of LOW_DENSITY_RESIDENTIAL, HIGH_DENSITY_RESIDENTIAL and HIGH_INTENSITY_INDUSTRIAL:
    - **WRFV3/phys/module_sf_urban.F**
    - **WRFV3/phys/module_surface_driver.F**
    - **WRFV3/phys/module_sf_noahdrv.F**

Note: Here the modifications are only applicable to using the single-layer Urban Canopy Model (SLUCM) and not the BEP-BEM model. If using the latter, additional changes to WRFV3/phys/module_sf_bep.F and WRFV3/phys/module_sf_bep_bem.F will be required.

The modifications are repeated below, but note that the line reference information may be different if a different version of WRF is used. However, generally the modifications follow a similar pattern. Note that I have commented out the old code and included the modifications, so the line numbers refer to the module copies available on the git repo.

WRFV3/phys/module_sf_urban.F Modification over lines 2585-2756

There are 3 if statements relating to the original urban classes starting with:

```
IF( IVGTYP(I,J) == LOW_DENSITY_RESIDENTIAL) THEN
```

```
IF( IVGTYP(I,J) == HIGH_DENSITY_RESIDENTIAL) THEN
```

```
IF( IVGTYP(I,J) == HIGH_INTENSITY_INDUSTRIAL) THEN
```

Comment these if statements and the calculations performed within.

Replace this with:

```
        IF( IVGTYP(I,J) == ISURBAN)  THEN
            UTYPE_URB2D(I,J) = 2  ! for default. high-intensity
            UTYPE_URB = UTYPE_URB2D(I,J)  ! for default. high-intensity
            FRC_URB2D(I,J) = 0. ! ALH Added
            IF (HGT_URB2D(I,J)>0.) THEN
            CONTINUE
            ELSE
            CALL wrf_debug(100, 'USING DEFAULT URBAN MORPHOLOGY')
            LP_URB2D(I,J)=0.
```

```
                LB_URB2D(I,J)=0.
                HGT_URB2D(I,J)=0.
                IF ( sf_urban_physics == 1 ) THEN
                     MH_URB2D(I,J)=0.
                     STDH_URB2D(I,J)=0.
                     DO K=1,4
                     LF_URB2D(I,K,J)=0.
                     ENDDO
                ELSE IF ( ( sf_urban_physics == 2 ) .or. ( sf_urban_physics
== 3 ) ) THEN
                     DO K=1,num_urban_hi
                     HI_URB2D(I,K,J)=0.
                     ENDDO
                ENDIF
                ENDIF
                IF (FRC_URB2D(I,J)>0.and.FRC_URB2D(I,J)<=1.) THEN
                CONTINUE
                ELSE
                WRITE(mesg,*) 'WARNING, FRC_URB2D = 0 BUT IVGTYP IS URBAN'
                call wrf_message(mesg)
                WRITE(mesg,*) 'WARNING, THE URBAN FRACTION WILL BE READ
FROM URBPARM.TBL'
                call wrf_message(mesg)
                FRC_URB2D(I,J) = FRC_URB_TBL(UTYPE_URB)
                ENDIF
                SWITCH_URB=1
           ENDIF

           IF( IVGTYP(I,J) .GE. 31) THEN
                UTYPE_URB2D(I,J) = IVGTYP(i,j)-30  !
                UTYPE_URB = UTYPE_URB2D(I,J)  !
           IF (HGT_URB2D(I,J)>0.) THEN
                CONTINUE
                ELSE
                CALL wrf_debug(100, 'USING DEFAULT URBAN MORPHOLOGY')
                LP_URB2D(I,J)=0.
                LB_URB2D(I,J)=0.
                HGT_URB2D(I,J)=0.
                IF ( sf_urban_physics == 1 ) THEN
                     MH_URB2D(I,J)=0.
                     STDH_URB2D(I,J)=0.
                     DO K=1,4
                     LF_URB2D(I,K,J)=0.
                     ENDDO
                ELSE IF ( ( sf_urban_physics == 2 ) .or. ( sf_urban_physics
== 3 ) ) THEN
                     DO K=1,num_urban_hi
                     HI_URB2D(I,K,J)=0.
                     ENDDO
                ENDIF
                ENDIF
                IF (FRC_URB2D(I,J)>0.and.FRC_URB2D(I,J)<=1.) THEN
                CONTINUE
                ELSE
                WRITE(mesg,*) 'WARNING, FRC_URB2D = 0 BUT IVGTYP IS URBAN'
                call wrf_message(mesg)
```

```
                WRITE(mesg,*) 'WARNING, THE URBAN FRACTION WILL BE READ
FROM URBPARM.TBL'
                call wrf_message(mesg)
                FRC_URB2D(I,J) = FRC_URB_TBL(UTYPE_URB)
                ENDIF
           SWITCH_URB=1
           ENDIF
```

WRFV3/phys/module_surface_driver.F

Search for all instances where there is:

```
IVGTYP(I,J) == LOW_DENSITY_RESIDENTIAL
```

Comment these out and replace as follows:

```
IF( IVGTYP(I,J) == ISURBAN .or. &
!IVGTYP(I,J) == LOW_DENSITY_RESIDENTIAL .or. & !urban
!IVGTYP(I,J) == HIGH_DENSITY_RESIDENTIAL .or.&
!IVGTYP(I,J) == HIGH_INTENSITY_INDUSTRIAL) THEN  !urban
IVGTYP(I,J) .GE. 31 ) THEN  !urban
```

This occurs at lines: 2889-2895, 2912-2918, 3225-3231, 3255-3259, and 3280-3284.

WRFV3/phys/module_sf_noahdrv.F

Search for all instances where there is:

```
LOW_DENSITY_RESIDENTIAL
```

Comment out the if statements and replace as follows:

```
IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) .GE. 31 ) THEN
! IF( IVGTYP(I,J) == ISURBAN .or. IVGTYP(I,J) ==
LOW_DENSITY_RESIDENTIAL .or. &
!   IVGTYP(I,J) == HIGH_DENSITY_RESIDENTIAL .or. IVGTYP(I,J) ==
HIGH_INTENSITY_INDUSTRIAL) THEN
```

This occurs at lines 911-915, 932-936, 952-956, 1251-1255, 3268-3272, 3281-3285, 3310-3314, 3557-3561, 4173-4177, 4194-4198, 4215-4219, 4455-4459.

On lines 3567-3586 comment/replace with the following:

```
!IF(IVGTYP(I,J)==LOW_DENSITY_RESIDENTIAL) UTYPE_URB=1
!IF(IVGTYP(I,J)==HIGH_DENSITY_RESIDENTIAL) UTYPE_URB=2
!IF(IVGTYP(I,J)==HIGH_INTENSITY_INDUSTRIAL) UTYPE_URB=3
IF(IVGTYP(I,J).GE.31) UTYPE_URB=IVGTYP(I,J)-30

! The following should not be hard coded here.
!These values are within the URBPARM.TBL file
! Already provided in INOUT
!IF(UTYPE_URB==1) FRC_URB2D(I,J)=0.5
!IF(UTYPE_URB==2) FRC_URB2D(I,J)=0.9
!IF(UTYPE_URB==3) FRC_URB2D(I,J)=0.95
```

- Once the code modifications have been done, compile the code

2. Local Climate Zone (LCZ) Map of Sydney

This uses a tool from [World Urban Database and Access Portal Tools](#) (WUDAPT).

The LCZ Map of Sydney was started by Jessica Lindqvist a Master's Intern from the University of Gothenburg and updated by Brooke Conroy a CLEX Summer Scholar from the University of Wollongong.

A description of the procedure used as well as challenges and limitations is currently in preparation for the manuscript:

> Lindqvist, J., B. Conroy, M. A. Hart, A. Maharaj, and A. L. Hirsch (in preparation). Classifying land-use in Sydney, Australia for understanding urban climate.

For Sydney, the region of interest was sufficiently large that the region had to be split in two, creating LCZ maps within the files: **`sydney_top.nc`** and **`sydney_bottom.nc`**. These were merged into a single netcdf file: **`sydney_merged.nc`** and also written out to a text file **`sydney_merged.txt`** with the values mapped to the LCZs using the python notebook **`calculate_parameters.ipynb`**. Both the Sydney LCZ netcdf/text file and python notebook are on the git repo.

The python notebook also contains the code used to estimate the parameters for each urban class that are required for the **`URBPARM.TBL`** file.

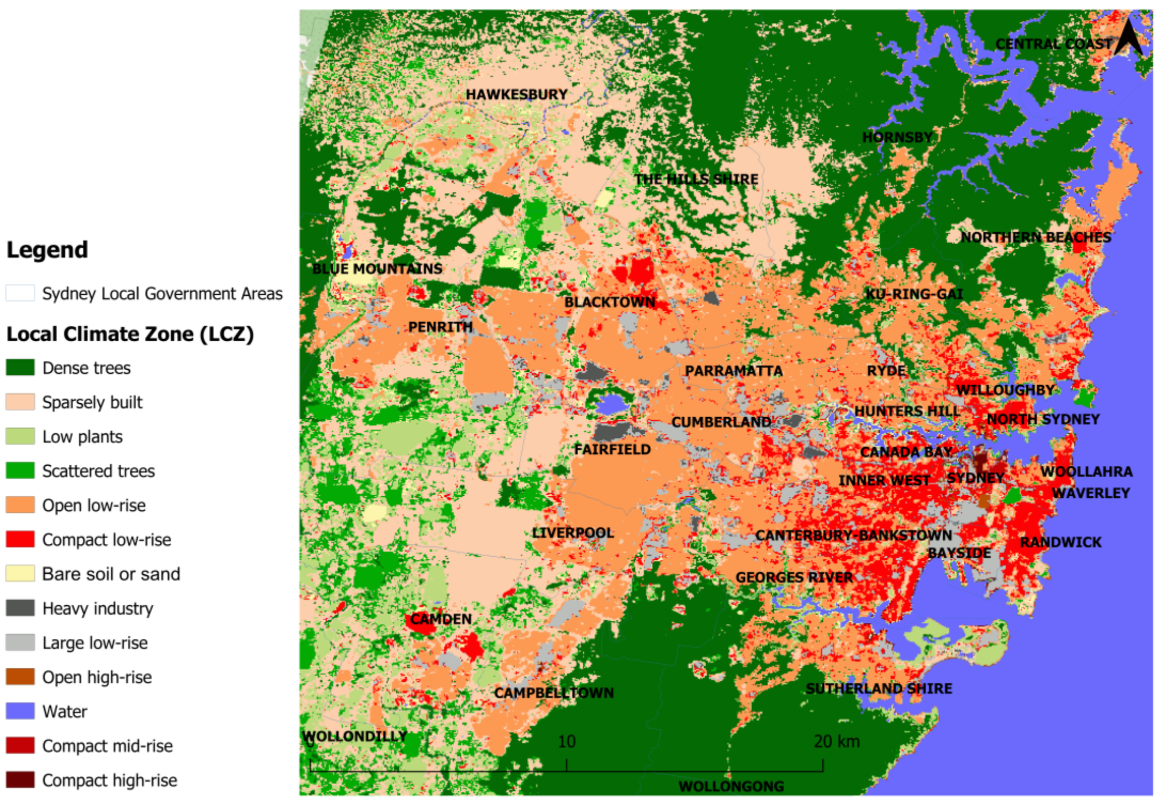

Figure 1: LCZ Map for Sydney

3. WUDAPT to WRF and Domain Configuration

This section describes how to get WPS to replace the land use types with the LCZ map. This is an adaptation of the instructions provided by the World Urban Database and Access Portal Tools (WUDAPT) available at http://www.wudapt.org/wudapt-to-wrf/.

The code is available from: https://sourceforge.net/projects/wudapt2wrf/

The following citation is required for use of this code:

> Brousse, O., Martilli, A., Foley, M., Mills, G., and Bechtel, B. (2016). WUDAPT, an efficient land use producing data tool for mesoscale models? Integration of urban LCZ in WRF over Madrid. Urban Climate, 17:116-134, doi:10.1016/j.uclim.2016.04.001.

Download the wudapt2wrf code to a subdirectory where you have your WRF code.

- In your wudapt2wrf directory, put a copy of the LCZ map in the *.txt file format. Make sure it is named <city>.txt (i.e. sydney.txt)
- The **`rd_wr_binary.f90`** module needs to updated so that the parameters and region of interests are consistent with the number of points and extent of your LCZ map
- On line 36:

```
parameter (nx=962,ny=799,nz=1,np=768638,nurbm=10)
```

- Note that:
    - nx: number of longitudes
    - ny: number of latitudes
    - np: number of points (i.e. nx * ny)
    - These values can be obtained from your LCZ netcdf file (in this case sydney_merged.nc)
- Define the Region of Interest:

```
latmin=-34.2
longmin=150.4
latmax=-33.4
longmax=151.6
```

- In the **`rd_wr_binary.f90`** module update:

```
open(unit=20,file='[name_of_the_city].txt',status='old')
```

with

```
open(unit=20,file='sydney.txt',status='old')
```

- Finally the the **`rd_wr_binary.f90`** module expects the *.txt file to include a gridcell ID. As the sydney.txt file does not have this replace:

```
read(20,*) iip,xx,yy,landuse_i(ip)
```

with

```
read(20,*) xx,yy,landuse_i(ip)
```

- Finally, if there are nan values in the sydney.txt file they need to be replaced with -9999. To do this open the sydney.txt file in your preferred editor do a find and replace for nan with -9999.
- Load the compilers used to compile WRF/WPS to compile the **`rd_wr_binary.exe`**:

```
module load intel-compiler/2019.3.199
make
```

- Check that a new **`rd_wr_binary.exe`** is created and then run:

```
./rd_wr_binary.exe
```

- If successful the following is returned, these values will be determined by your city and LCZ data but look similar in format, make sure you note them down:

```
dx=  1.2474139E-03 dy=  1.0012506E-03
start calculating most frequent value
-8.5420249E+14 -5.8534129E-05
0.0000000E+00    Infinity
```

- There will also be a new file in your directory called **`landuse_urban`**
- Now create a sub-directory in WPS and copy this file across:

```
mkdir -p WPS/landuse_sydney/
scp -p landuse_urban ../WPS/landuse_sydney/.
cd ../WPS/landuse_sydney/
```

- Rename landuse_urban according to: ?????-nx.?????-ny using the nx and ny values defined in the **`rd_wr_binary.f90`** module:

```
mv landuse_urban 00001-00962.00001-00799
```

- Now you need to create an index file in **`WPS/landuse_sydney/`** You need to provide information that is relevant to your LCZ map and includes the dx and dy are the values returned when you executed **`rd_wr_binary.exe`** and that the **`known_lat`**, **`known_lon`**, **`tile_x`** and **`tile_y`** are the values you entered in **`rd_wr_binary.f90`**. The following is what was used for the Sydney LCZ:

```
type=categorical
category_min=31
category_max=40
projection=regular_ll
missing_value=0.
dx=0.00124741 ! value given up rd_wr_binary.exe
dy=0.00100125 ! value given up rd_wr_binary.exe
known_x=1.0
known_y=1.0
known_lat=-34.2 ! Latitude of SW corner i.e. latmin
known_lon=150.4 ! Longitude of SW corner i.e. longmin
wordsize=1
```

```
tile_x=962 ! nx
tile_y=799 ! ny
tile_z=1
units="category"
description="10-category UCZ"
mminlu="MODIFIED_IGBP_MODIS_NOAH"
```

Note: For WPS versions 3.8 and newer the default land use classification is MODIFIED_IGBP_MODIS_NOAH.

- Edit the `GEOGRID.TBL` file found in WPS/geogrid/ and add:

```
name=LANDUSEF
      priority=2
      dest_type=categorical
      z_dim_name=land_cat
      interp_option = default:nearest_neighbor
      abs_path = default:<path to your WRF code>/WRF/WPS/landuse_sydney/
```

- Set up your grid configuration in the `namelist.wps` file.
- For our setup, we have a parent domain (d01) at 4km and two nested domains (d02 and d03) at 800m. Note that d03 has the same extent as d02 but with no urban land use types. These domains retain a parent-nest ratio of 3:1 for the boundary forcing to the parent domain and a ratio of 5:1 for the parent domain to the nested domain. This looks like:

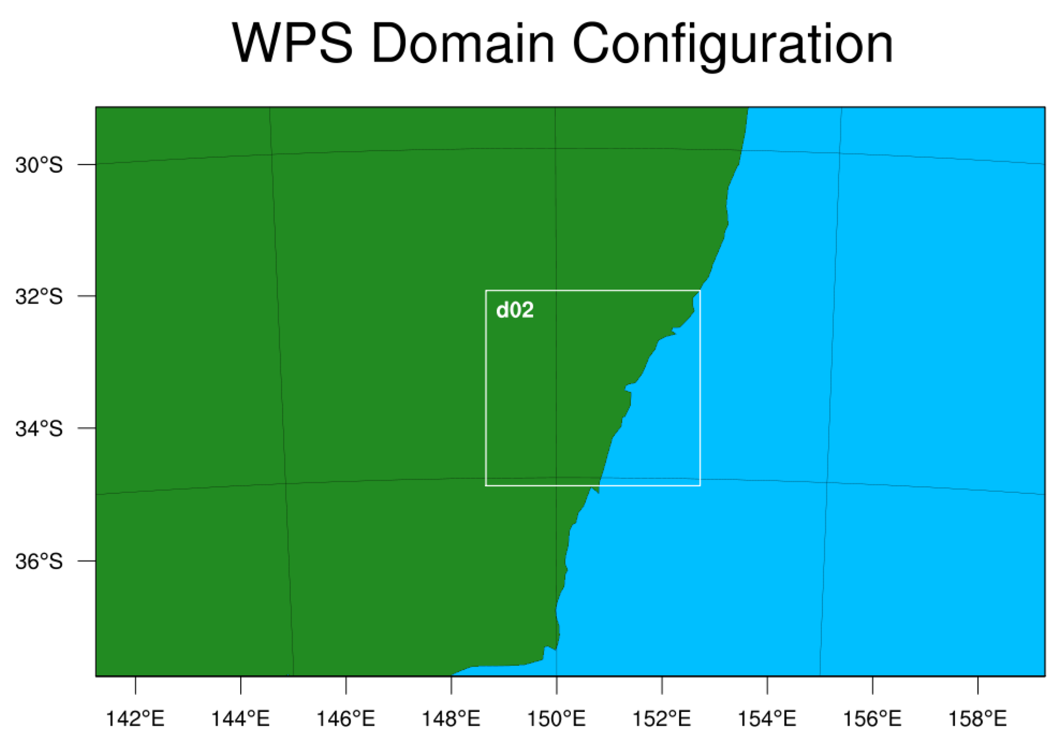

Figure 2: WPS Domain configuration for the Sydney 800m simulations.

A copy of the WPS namelist is available on the git repo and reproduced here:

```
&share
 wrf_core = 'ARW',
 max_dom = 3,
 start_date = '2017-01-01_00:00:00','2017-01-01_00:00:00','2017-01-
01_00:00:00',
 end_date   = '2017-02-28_23:59:00','2017-02-28_23:59:00','2017-02-
28_23:59:00',
 interval_seconds = 21600
 io_form_geogrid = 2,
 opt_output_from_geogrid_path = './',
 debug_level = 0,
/

&geogrid
 parent_id      =    1,   1,   1,
 parent_grid_ratio =  1,   5,   5,
 i_parent_start =    1,  165,  165,
 j_parent_start =    1,  81,  81,
 e_we           =  400, 451, 451,
 e_sn           =  240, 411, 411,
 geog_data_res = 'default','default','default',
 dx = 4000.,
 dy = 4000.,
 map_proj = 'lambert',
 truelat1 = -35.
 truelat2 = -20.
 ref_lat   =  -33.7,
 ref_lon   = 150.26,
 stand_lon = 150.26,
 geog_data_path = '/g/data/sx70/data/WPS_GEOG_v4/'
/

&ungrib
 out_format = 'WPS',
 prefix = 'INVAR',
/

&metgrid
 constants_name = '<path to BARRA surface data in intermediate
format>/BARRA_INV:0000-00-00_00'
 fg_name = '<path to BARRA surface data in intermediate format>/BARRA_SFC',
'<path to BARRA model level data in intermediate format>/BARRA_MDL'
 io_form_metgrid = 2,
 opt_metgrid_tbl_path = '.'
```

- Once you have your domain configuration run **`geogrid.exe`** and check the lu_index in the **`geo_em.d0?.nc`** files. There should now be values 31-40 over your urban area.
- For reference in **`LANDUSE.TBL`**: MODIFIED_IGBP_MODIS_NOAH 13 == 'Urban and Built-Up'

Note: Urban areas outside the LCZ map that are classified as PFT = 13. These are changed to LCZ6 (open low-rise) PFT = 36 which can be done with NCO on the command line for all domains:

```
module load nco
ncap2 -s 'where(LU_INDEX==13) LU_INDEX=36' geo_em.d0?.nc geo_em.d0?.nc
ncap2 -s "LANDUSEF(:,35,:,:)=LANDUSEF(:,35,:,:)+LANDUSEF(:,12,:,:)"
geo_em.d0?.nc geo_em.d0?.nc
ncap2 -s "LANDUSEF(:,12,:,:)=0" geo_em.d0?.nc geo_em.d0?.nc
```

Repeat this for each of your **`geo_em.d0?.nc`** files.

- For our setup we ran a third domain, d03, that has the same extent as d02 but with no urban land use types. This was done so that we could compare urban vs. no-urban over the same domain and resolution to quantify urban heat island effects. To do this, one can copy the geo_em.d02.nc file and replace the LU_INDEX and LANDUSEF as follows:

```
scp -p geo_em.d02.nc geo_em.d03.nc
ncap2 -s 'where(LU_INDEX>30) LU_INDEX=10' geo_em.d03.nc geo_em.d03.nc
ncap2 -s
"LANDUSEF(:,9,:,:)=LANDUSEF(:,9,:,:)+LANDUSEF(:,30,:,:)+LANDUSEF(:,31,:,:)+LANDUSEF
(:,32,:,:)+LANDUSEF(:,33,:,:)+LANDUSEF(:,34,:,:)+LANDUSEF(:,35,:,:)+LANDUSEF(:,36,:
,:)+LANDUSEF(:,37,:,:)+LANDUSEF(:,38,:,:)+LANDUSEF(:,39,:,:)" geo_em.d03.nc
geo_em.d03.nc
ncap2 -s "LANDUSEF(:,30:,:,:)=0" geo_em.d03.nc geo_em.d03.nc
```

4. BARRA Boundary Conditions

This section explains how to use the Bureau of Meteorology Atmospheric high-resolution Regional Reanalysis for Australia (BARRA) as the boundary condition forcing for your WRF domain, provided of course it covers Australia. More information about BARRA is available at: http://www.bom.gov.au/research/projects/reanalysis/ and you will need to appropriately cite the following paper:

> Su, C.-H., Eizenberg, N., Steinle, P., Jakob, D., Fox-Hughes, P., White, C. J. (2019). BARRA v1.0: the Bureau of Meteorology Atmospheric high-resolution Regional Reanalysis for Australia. Geoscientific Model Development, 12:2049-2068, doi:10.5194/gmd-12-2049-2019.

4.1 Get BARRA Data

- The first step is to join project ma05 on NCI via https://my.nci.org.au/mancini/login?next=/mancini/ where the BARRA data is currently stored.
  - File structure: /g/data/ma05/BARRA_R/v1/analysis/$level/$variable/YYYY/MM
  - File name convention: $variable-an-$level-PT0H-BARRA_R-v1-YYYYMMDDTHHmmZ.nc
  - Static Variables are in /g/data/ma05/BARRA_R/v1/static/ and the required variables are the topography, land mask and height_theta
  - Surface Variables required:
    - MSLP, surface temperature and pressure
    - screen level temperature and specific humidity
    - 10m U and V wind components
    - Soil temperature and moisture
  - Pressure Level data requirements: temperature, pressure, U and V wind components, geopotential height and specific humidity
  - Both the surface and model level data is at a 6-hourly time interval. This is adequate for use as WRF boundary conditions.
- The BARRA data collection at NCI doesn't not include data on all 70 model levels due to current storage constraints. It is preferable to use the BARRA data on the model levels as this has better vertical resolution which is preferable when running WRF at convection permitting scales. Therefore you need to contact the BARRA Helpdesk (helpdesk.reanalysis@bom.gov.au) and request the 70 model level data for temperature, pressure, specific humidity, geopotential height, U and V wind components for the time period of interest.
- Once the BARRA Helpdesk has your data ready transfer that to a wps-barra subdirectory that you can create within your WRF working directory

4.2 Convert BARRA data into WPS intermediate format

Christopher Thomas (christopherthomas@cmt.id.au) has developed Fortran code to convert the BARRA data into the intermediate format required by the WPS tool called Metgrid. It basically replaces the ungrid.exe step for WPS.

A copy of these codes is available from: https://bitbucket.org/christopherthomas/barra2wrf

- Download/clone the code to your wps-barra subdirectory within your WRF work area.
- Note that this code automatically calculates the geopotential heights.
- In **`make_intermediate.pbs`** you need to update the directory paths to where you are working and where you put the BARRA 70 model level data provided from BoM. You also need make sure that the for loops for creating all the symbolic links cover your period of interest.
- The compile the code and run:

```
sh compile_barra_to_intermediate.sh
qsub make_intermediate.pbs
```

- When complete, you will have 3 groups of files:
  - **`BARRA_INV:0000-00-00_00`** - contains the invariant variables topography and landmask
  - **`BARRA_MDL:YYYY-MM-DD_HH`** - contains the model level data in WPS intermediate format
  - **`BARRA_SFC:YYYY-MM-DD_HH`** - contains the surface data in WPS intermediate format
- Once these files are ready, you can proceed with running **`metgrid.exe`** making sure that the **`namelist.wps`** has the following:

```
&metgrid
 constants_name = '<path to where you have this file>/BARRA_INV:0000-00-
00_00'
 fg_name = 'BARRA_SFC', 'BARRA_MDL'
 io_form_metgrid = 2,
 opt_metgrid_tbl_path = '<path to where you have WRF code>/WRF/WPS/metgrid'
 opt_output_from_metgrid_path = '<path to your run directory>/run/bdy_data'
```

5. WRF Configuration and Tips for running at <1km

The WRF configuration used for the Sydney 800m simulations is provided in **`namelist.wrf`** available on the git repo.

There are, however, a few gotchas when setting up to run with the SLUCM in WRF at <1km resolution. This includes:

- NCI resource requests to run all 3 domains simultaneously is about 15 days at a time is memory intensive and best if the number of CPUs is a multiple of 48. The PBS request was:

```
#PBS -q normal
#PBS -l walltime=48:00:00
#PBS -l mem=50GB
#PBS -l ncpus=96
```

- The timestep needs to be set correctly to avoid CFL criterion violation, we used a 10 second timestep for the 4km parent domain:

```
time_step = 10,
```

- If you get segmentation faults, that may be related to the CFL error the following page has some useful tips: https://forum.mmm.ucar.edu/phpBB3/viewtopic.php?f=73&t=133
- I found that the output files would get very large quickly, writing the output at 10 min intervals with 6 per file (hourly output files) seemed to work well
- The atmospheric vertical levels have an issue with the building heights, particularly for LCZ1 compact high-rise. Therefore the lowest atmospheric model level needs to be taller than that. To specify these, we ran **`real.exe`** without defining the **`eta_levels`** to find out what they default to. Then we add **`eta_levels`** to the **`namelist.wrf`** before running **`real.exe`** again to create the boundary condition files. We used:

```
eta_levels = 1.00, 0.98, 0.9734183, 0.9570671, 0.9384401, 0.9173963,
     0.8938418,0.8677447, 0.8391488, 0.8081841, 0.7750692, 0.7401074,
     0.7036741,0.6661961, 0.6281269, 0.5899198, 0.5520039, 0.5145977,
     0.4778092,0.4417463, 0.406516, 0.372223, 0.3389684, 0.3068487,
     0.2759545,0.2472012, 0.2209639, 0.1970225, 0.1751762, 0.1552415,
     0.1370513,0.1204529, 0.1053069, 0.09148624, 0.07887501, 0.06736735,
     0.05686668,0.04728488, 0.03854156, 0.03056332, 0.02328323, 0.01664022,
     0.0105785,0.005047212, 0.00
```

6. Python code to conduct analysis

- **calculate_parameters.ipynb** - reads in the WUDAPT LCZ maps for Sydney, merges them, and estimates the urban parameters of building height, vegetation fraction, building roof albedo and anthropogenic heat
- **common_functions.py** - contains functions for calculating ILAMB skill metrics
- **plot_AWS_comparison.ipynb** - using 10 minute AWS data across Sydney, extracts closest WRF grid cell and calculates the skill metrics for a range of variables as well as plots time series over the simulation period to compare observed and modelled time series
- **plot_East_West_comparison.ipynb** - using 10 minute AWS data plots the time series for stations along a similar line of latitude to compare the gradient across the city for different variables, also does this for the corresponding WRF grid cells
- **plot_determine_HW_timing_AWS.ipynb** - determines heatwave timing from AWS data using EHF methodology
- **plot_landuse_WRF.ipynb** - plots the dominant surface types and topography for the WRF domains
- **plot_urban_vs_grass.ipynb** - plots time series and KDEs of AWS data and WRF data for urban and non-urban simulations
- **plot_LCZ_contrast.ipynb** - plots KDEs and diurnal cycle of each of the LCZs where only the grid cells where an LCZ is the dominant type are used. Only the diurnal cycle figure used in the paper
- **plot_contour.ipynb** - produces contour maps of a number of variables coincident with the heatwave events
- **plot_transect.ipynb** - extracts a transect across the domain and plots the vertical profile (HGT vs LONGITUDE) for a number of variables as well as produces the cross city times series used in the paper. Extracting the data is best done on gadi than vdi and saved there. For the vertical cross sections can create figures for every output instance that can then be combined in a mp4/gif
- **plot_lapse_rate.ipynb** - calculate the lapse rate between two points along the same line of latitude and produce time series, also reads in the Abatzoglou et al. 2020 data to determine Foehn days
- **plot_advection.ipynb** - calculate the low-level heat and moisture advection and saves to netcdf. No plotting. - not used in paper. Better to run gadi than vdi.
- **plot_vertical_crosssection.ipynb** - looks at the vertical profile as HGT vs TIME for a number of variables. Data that is extracted is saved, best to run on gadi than vdi. Not used in paper.