

## Table of Contents

Why a postprocessor?	2
WRF4G postprocessor	2

## Why a postprocessor?

The raw files produced by WRF need to be postprocessed to make them [?CF-compliant](#). But this is conveniently carried out after the simulations are finished. However, for users that do not have large storage resources, the whole WRF raw files on an experiment can be too heavy. In fact, by default, WRF saves many variables, and these files are large. Thus, WRF4G provides the possibility of passing the WRF raw output files through a post-processing script as WRF is running, using the local grid resources.

## WRF4G postprocessor

Similar to preprocessor, WRF4G postprocessor is an executable script that can be written in any programming language, provided it does accept one argument, which is one WRF raw file. We usually use a modified version of p\_interp to interpolate to filter out unnecessary variables and interpolate to pressure levels. Note that users with large storage resources may not be interested in using a postprocessor.

Sample postprocessor:

```
#!/bin/bash
#
# Sample postprocessor
#
# This can be any kind of executable accepting one argument:
#
wrfnc_file=$1      # The WRF file to postprocess
#
# and creating a postprocessed file with the same name to be uploaded by register_file
#

function pintnml(){
  idir=$1
  ifile=$2
  cat << EOF > namelist.pinterp
&io
  path_to_input      = '${idir}/',
  input_name         = '${ifile}',
  path_to_output     = '${idir}/',
  fields             = 'RAINTOT,T2,Q2,PSFC,U10,V10,CLT,SFCEVP,QFX,GLW,SWDOWN,RAINC,HGT,SMOIS,PH,PHB,GHT,T,TT,PRES,MSL',
  process            = 'list',
  debug              = .FALSE.,
  grid_filt          = 3,
  ntimes_filt        = 10,
/

&interp_in
  interp_levels      = 1000.0, 925., 850.0, 700., 500.0, 300,
  extrapolate        = 1,
  interp_method      = 1,
  unstagger_grid     = .TRUE.,
/
EOF
}

pintnml . ${wrfnc_file}
p_interp
rm namelist.pinterp

xptdsize=$( echo "print int((stat -c %s ${wrfnc_file})*0.025)" | python )
if test "$(stat -c %s ${wrfnc_file}_PLEV)" -ge ${xptdsize}; then
  mv "${wrfnc_file}_PLEV" "${wrfnc_file}"
fi
```