ARPS2PILT V1.0 User's Guide

MICHAEL T. KIEFER

MICHIGAN STATE UNIVERSITY, EAST LANSING, MICHIGAN, 48823

Contents

1. Introduction to ARPS2PILT

ARPS2PILT is a program that converts Advanced Regional Prediction System [ARPS; Xue et al. (2000, 2001)] history files (containing standard ARPS variables) to files that the Pacific Northwest National Laboratory (PNNL) Integrated Lagrangian Transport (PILT) model can readily ingest (containing standard Weather Research and Forecasting [WRF; Skamarock et al. (2005)] model variables). Hereafter, we will refer to these variables as 'PILT variables'. Note that PILT is the name given to the most recent version of the FLEXPART model (Stohl et al. 2005); in contrast to FLEXPART, PILT is specifically designed for use with WRF model output. ARPS2PILT is based on the ARPS program ARPSCVT and uses the 'arpscvt.input' namelist (with minor modifications).

2. ARPS2PILT Formulation

a. Overview of Modifications to ARPSCVT

The following files are modified versions of existing files in the ARPS package and are required in order to compile ARPS2PILT: netio3d.arps2pilt.f90, module_arps_netio_metadata.arps2pilt.f90, initpara3d.arps2pilt.f90, arpscvt.arps2pilt.f90, and globcst.arps2pilt.inc.

In 'netio3d.arps2pilt.f90', the subroutine **NETDUMP** was modified to add a new data output flag ('flexout') to allow users to dump out history files with a typical grouping of ARPS variables (flexout=0) or with variables needed to run the PILT model (flexout=1); recall that PILT is the name given to the most recent version of the FLEXPART model, hence the use of "flex" in various portions of the ARPS2PILT formulation. At this time, if 'flexout' is set to 1, all other data dumping flags (e.g., tkeout) need to be set to 0 in order for the program to work properly. Note that the file 'module_arps_netio_metadata.arps2pilt.f90' is a version of the metadata file used by the NetCDF read/write subroutines, with PILT variables defined. In each of the other files (initpara3d.arps2pilt.f90, arpscvt.arps2pilt.f90, and globcst.arps2pilt.inc), modifications have been made to accommodate the new data output flag and allow the user to control the flag from the namelist file.

b. Calculation of PILT variables from Standard ARPS Output

In the subroutine **NETDUMP** in file ‘netio3d.arps2pilt.f90’, the following variables are computed and then passed to subroutines that write out data in NetCDF format:

i. Times

   Date/time formatted for PILT (YYYY-MM-DD_HH:MM:SS).

ii. P_TOP: pressure at top of model [Pa]

   Set equal to the ARPS base state pressure at k=nz-1 (uppermost non-boundary grid point).

iii. ZNW: eta values at “full" level, i.e. at top of grid box [value between 0 and 1]

   $\eta(k) = \frac{P(k)-P_{TOP}}{P_s-P_{TOP}}$, where $P(k)$ is base state pressure at level k, P_s is surface base state pressure, and P_{TOP} is defined as before. η is computed at $i = 1$, $j = 1$.

iv. ZNU: eta values at “half" level, i.e. at center of grid box [value between 0 and 1]

   ZNW, interpolated to the “half" level on the staggered grid.

v. HGT: terrain height [meters above mean sea level]

   Set equal to the ARPS physical height variable (*zp*) at the ground surface (k = 1).

vi. XLAT: latitude, south is negative [decimal degrees]

   Calculated using ARPS subroutine **xytoll** (for a given i,j pair, the subroutine computes latitude and longitude at that grid point, using the domain center latitude/longitude and map projection information contained in the header of each input file).

vii. XLONG: longitude, west is negative [decimal degrees]

Calculated using ARPS subroutine **xytoll** (see XLAT).

viii. PSFC: surface pressure [Pa]

Computed as sum of base state pressure (*pbar*) and the prognostic variable *pprt* (perturbation pressure).

ix. U10: U at 10 m above ground level (AGL) [m s^{-1}]

If the lowest u-wind point (half the height of the lowest grid cell), is at or below 10 m AGL, the u-wind component at that level is used. Otherwise, a log-wind profile is assumed and the u-wind component at the lowest model level is interpolated to 10 m AGL (using roughness length, originally specified in the ARPS namelist, and friction velocity, computed from the u-component of the vertical momentum flux).

x. V10: V at 10 m AGL [m s^{-1}]

Same as for U10, except for the v-wind component.

xi. T2: Temperature at 2 m AGL [K]

If the lowest scalar point (half the height of the lowest grid cell), is at or below 2 m AGL, then the potential temperature at that lowest point is converted to temperature.

Otherwise, we use K-theory (Stull 1988) to estimate 2 m AGL potential temperature from the lowest–scalar-level potential temperature: $-ptsflx = \langle w'\theta' \rangle_{2m-k2} = K_h \left(\frac{\theta_{2m} - \theta_{k2}}{\delta z} \right)$, where K_h is eddy diffusivity (computed by ARPS); $ptsflx$ is the surface turbulent heat flux computed by ARPS (using a bulk aerodynamic approach); $\langle w'\theta' \rangle_{2m-k2}$ is the vertical turbulent heat flux in the layer between the lowest scalar level (k=2) and 2 m AGL; θ_{k2} and θ_{2m} are potential temperature at the lowest scalar level and 2 m AGL, respectively; and δz is the depth of the layer between the lowest scalar level and 2 m AGL. The critical assumption made here is that the vertical turbulent heat flux does not vary between the ground surface and the lowest scalar level, such that $\langle w'\theta' \rangle_{2m-k2} = -ptsflx$. Finally, potential temperature is converted to temperature. Note: if the grid point is classified as over-water (determined from soil type variable *soiltyp*), or if the vertical component of eddy diffusivity is very small (set somewhat arbitrarily as less than 0.25 m^2 s^{-1}), the 2 m temperature is set equal to the uppermost soil temperature (equivalent to skin temperature in ARPS).

xii. Q2: water vapor mixing ratio at 2 m AGL [kg kg^{-1}]

The procedure is the same as for T2, except for moisture. There is one additional minor step, where the native ARPS variable specific humidity (*SH*) is converted to mixing ratio (*MR*), where $MR = \frac{SH}{1-SH}$.

xiii. RAINNC: Accumulated total grid scale precip [mm]

Set equal to ARPS grid-resolved precipitation, *raing*.

xiv. RAINC: Accumulated total cumulus precip [mm]

Set equal to ARPS cumulus precipitation, *rainc* (this array will be empty if the cumulus parameterization was switched off).

xv. SNOWH: Physical snow depth [m]

Set equal to ARPS snow depth, *snowdpth*.

xvi. HFX : Upward heat flux at surface [W m^{-2}]

This is computed as $-ptsflx * C_p$, where $ptsflx$ is the ARPS surface sensible heat flux [K kg m^{-2} s^{-1}], defined positive downward, and C_p is the specific heat at constant pressure [J K^{-1} kg^{-1}]. Note there is an error in the comment lines of the ARPS source code that has not been corrected: $ptsflx$ units are K kg m^{-2} s^{-1}, not K kg m^{-1} s^{-2}.

xvii. UST : U* in similarity theory [m s^{-1}].

Computed as in Eq. (2.10b) in Stull (1988), from momentum flux and base state density.

xviii. PBLH : PBL height [m]

Computed in existing ARPS subroutine **pbldepth** wherein PBL depth is computed as the level where environmental virtual potential temperature first exceeds the value at the lowest model level. If the atmosphere is stable right above the ground, the PBL depth is set to the thickness of the layer below the first scalar point above the ground (i.e., one-half of the lowest grid cell depth). Note: Since ARPS2PILT is applied regardless of whether the model output is from standard ARPS or ARPS-CANOPY, stable layers inside the canopy can produce an erroneously small PBL depth. Although a modified version of **pbldepth** is used in ARPS-CANOPY in which the model searches upward from the first grid point above the canopy, rather than the lowest model level (typically inside the canopy), that version is not used here.

xix. SWDOWN : Downward short wave radiation flux at ground surface [W m^{-2}]

Set equal to ARPS incoming shortwave radiation, *radsw*.

xx. U : X-wind component [m s^{-1}]

Set equal to ARPS east-west wind component, *u*.

xxi. V : Y-wind component [m s^{-1}]

Set equal to ARPS north-south wind component, *v*.

xxii. W : Z-wind component [m s^{-1}]

Set equal to ARPS vertical velocity, *w*.

xxiii. PH : perturbation geopotential [m^2 s^{-2}]

To compute PH, we use the same procedure as in WRF. First, PH at ground level is set to 0. Second, the model loops over the vertical dimension, from k = 2 to k = nz, computing PH as

$$PH(k) = PH(k-1) - [(ZNW(k) - ZNW(k-1)) * (PSFC - P\_TOP) * \rho]$$

where ρ is density and all other variables are defined earlier in this section. Note: since PH is a three-dimensional variable, the model loops over all points in the x- and y-dimensions, however, this detail is omitted here for simplicity.

xxiv. PHB : Base state geopotential [m^2 s^{-2}]

This is computed as the larger of either “0" or the ARPS geopotential height, *zp*, multiplied by gravitational acceleration [PHB is set to a minimum of “0" because *zp* may be of negative sign at k = 1 (if below sea level, or when using ARPS in idealized mode)].

xxv. P : Perturbation pressure [Pa]

Set equal to ARPS perturbation pressure, *pprt*.

xxvi. PB : Base state pressure [Pa]

Set equal to ARPS base state pressure, *pbar*.

xxvii. ALT : Inverse density [m^3 kg^{-1}]

This is computed by first calculating density from pressure and temperature (ARPS outputs base state density, not total density), and then computing the inverse.

xxviii. T : Perturbation potential temperature [K]

Set equal to ARPS perturbation potential temperature, *ptprt*.

xxix. TKE_MYJ : Turbulent kinetic energy (TKE) [m^2 s^{-2}]

Set equal to the ARPS variable *tke*. Warning: this is the parameterized TKE, not the total TKE (resolved plus parameterized subgrid-scale). Total TKE can be computed in post-processing, but is not computed by ARPS. Also note that MYJ refers to the Mellor-Yamada-Janjic PBL scheme; ARPS does not employ the MYJ scheme, but the name is retained for compatibility with PILT.

xxx. QVAPOR : Water vapor mixing ratio [kg kg^{-1}]

We compute mixing ratio (QVAPOR) from ARPS specific humidity (qv) as QVAPOR $= \frac{qv}{1-qv}$.

xxxi. CLDFRA : Cloud fraction [value between 0 and 1]

At this time, CLDFRA is set to 0.0 everywhere because PILT expects CLDFRA as a 2D variable and WRF provides it as a 3D variable (Ryan Shadbolt, personal communication).

3. ARPS2PILT Namelist

The entries in the ARPS2PILT namelist ('arpscvt_pilt.input') are standard ARPSCVT namelist entries, with the exception of 'flexout', which was added to allow for the dumping of PILT variables. The following entries in the namelist should be reviewed and modified as necessary, each time ARPS2PILT is run:

i. hinfmt

This is the file format of the input data files (recommended: 7 = netcdf or 1 = unformatted binary). Compiling 'arpscvt' with both HDF4 and NetCDF read/write capability is possible (http://www.caps.ou.edu/pipermail/arpssupport/2009-September/010657.html), but success is not guaranteed. Thus, the user may encounter problems when attempting to read in files in HDF4 format, and write files in NetCDF format (as required for PILT).

ii. hdmpfheader

This is the history file name header, including the directory. For example, one might specify '/home/ARPS/RUN/ARPSdump', where the files in directory '/home/ARPS/RUN/' are named 'ARPSdump.net000000', 'ARPSdump.net003600', etc.

iii. tintv_dmpin, tbgn_dmpin, tend_dmpin

Here the user needs to specify the time interval between files, the timestamp of the first input file and the timestamp of the last input file. Alternatively, the user can change option 'hdmpinopt' from 1 to 2, and specify the files individually, using the entries 'grdbasfn', 'nhisfile', and 'hisfile(n)', where there are n files (see namelist file for more information).

iv. dirname

Here the user specifies the directory where the converted files are to be dumped.

v. outrunname

Here the user specifies the header for the converted files. For example, setting 'outrunname' to 'PILTready' would generate files 'PILTready.net000000', 'PILTready.net003600', etc.

Note: There are additional entries in the namelist file, but they do not need to be modified for most ARPS2PILT applications.

4. Step-by-Step Instructions for Running ARPS2PILT

The following instructions assume that the user has a working version of either the ARPS model or the ARPS-CANOPY model [a modified version of ARPS with a canopy sub-model (Kiefer et al. 2013)]. The ARPS2PILT package consists of a number of files that must be integrated with the existing ARPS/ARPS-CANOPY source code before compiling (See Section 2a for more details about the requisite files).

i. In /home/ARPS/include, replace the contents of the include file ‘globcst.inc’ with the contents of ‘globcst.arps2pilt.inc’.

ii. In /home/ARPS/src/arps, replace the contents of ‘netio3d.f90’ with the contents of ‘netio3d.arps2pilt.f90’, replace the contents of ‘module_arps_netio_metadata.f90’ with the contents of ‘module_arps_netio_metadata.arps2pilt.f90’, and replace the contents of ‘initpara3d.f90’ with the contents of ‘initpara3d.arps2pilt.f90’.

iii. In /home/ARPS/src/arpscvt, replace the contents of ‘arpscvt.f90’ with the contents of ‘arpscvt.arps2pilt.f90’.

iv. If NetCDF libraries are not located in /usr/local/netcdf, open file ‘makearps’ in the main ARPS directory and modify line number 1536 to specify the directory where NetCDF libraries are located.

v. Compile ARPSCVT by typing ‘makearps -io net arpscvt’ in the main ARPS directory (i.e., wherever makearps is located).

vi. Modify the namelist file ‘arspcvt_pilt.input’ to specify the ARPS history files you wish to convert, what format they are in (unformatted binary or NetCDF), and the location where you want the PILT-ready NetCDF-formatted files to be dumped in. See Section 3 for more details regarding the namelist file.

vii. Run ARPS2PILT: [path to ARPSCVT executable] < arpscvt_pilt.input

viii. To examine the contents of the converted files in the directory where they were dumped: ncdump [file_name] | more

5. Known Issues and Limitations

i. When running ARPS2PILT, all output options flags in ‘arpscvt_pilt.input’, except ‘flexout’, must be set to 0. Otherwise, an error will occur and ARPS2PILT (i.e., ARPSCVT) will fail.

6. Further Information / Obtaining ARPS2PILT

For questions regarding ARPS2PILT, or to obtain a copy of the latest version of ARPS2PILT, please contact Michael Kiefer at mtkiefer@msu.edu.

7. Acknowledgements

Support for ARPS2PILT development was provided by the U.S. Joint Fire Science Program (Project # 09-1-04-1) and the USDA Forest Service (Research Joint Venture Agreement # 09-JV-11242306-089). The ARPS model was developed by the Center for Analysis and Prediction of Storms (CAPS), University of Oklahoma. This document was generated using version 3.2 of the official AMS LaTeX template.

REFERENCES

Kiefer, M. T., S. Zhong, W. E. Heilman, J. J. Charney, and X. Bian, 2013: Evaluation of an ARPS-based canopy flow modeling system for use in future operational smoke prediction efforts. *J. Geophys. Res.*, **999**, In Press.

Skamarock, W. C., J. B. Klemp, J. Dudhia, G. D. O, D. M. Barker, W. Wang, and J. G. Powers, 2005: A description of the Advanced Regional WRF Version 2. NCAR Tech Notes-468+STR, 100 pp.

Stohl, A., C. Forster, A. Frank, P. Seibert, and G. Wotawa, 2005: Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2. *Atmos. Chem. Phys.*, **5**, 2461–2474.

Stull, R. B., 1988: *An Introduction to Boundary Layer Meteorology*. Kluwer Academic, 666 pp.

Xue, M., K. K. Droegemeier, and V. Wong, 2000: The Advanced Regional Prediction System (ARPS) – A multi-scale nonhydrostatic atmosphere simulation and prediction model. Part I: Model dynamics and verification. *Meteorol. Atmos. Phys.*, **75**, 463–485.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, K. Brewster, F. Carr, D. Weber, Y. Liu, and D. Wang, 2001: The Advanced Regional Prediction System (ARPS) – A multi-scale nonhydrostatic atmosphere simulation and prediction tool. Part II: Model physics and applications. *Meteorol. Atmos. Phys.*, **76**, 143–165.