ARPS-Shift Phase/Position Error Program

Keith Brewster
Center for Analysis and Prediction of Storms
University of Oklahoma
Norman, OK 73019
kbrewster@ou.edu

Rev. 18 November 2004
ARPS version 5.1.2

1. Introduction

The ADAS position error scheme in program *`arpsshift`* is designed to objectively determine phase or position errors in a forecast field, and apply a position correction to the forecast field to smoothly and accurately correct those errors. The position correction vectors are also written to a separate file for use by ARPS to allow for the option to apply position corrections gradually within the ARPS model. Statistics about the shift vectors are reported so that the program may also be used for verification studies. The phase correction concept and an application to a simple observing simulation experiment are documented by Brewster (2003a). A demonstration of the position correction in an actual severe thunderstorm outbreak case is presented by Brewster (2003b).

The program takes an ARPS forecast and matches observations to the forecast. The program can also use an analysis or forecast from another model that has been interpolated to the ARPS model coordinate by *`ext2arps`*. The user must specify the name and format of the forecast file and the names and data types of the input observation files in the namelist input file, `arpsshift.input`. The current version matches data from single level (surface observations, aircraft, etc), vertical profile data (rawinsondes, wind profilers, etc), radar reflectivity and Doppler radar winds. An update to include matching to infrared satellite imagery will be implemented in the near future.

The phase correction matching is done in an iterative process, correcting first for synoptic and meso-β scale features, then making corrections to meso-γ (storm scale) features. The program accomplishes this by dividing the observations and forecast domain into a number of overlapping volumes (shift zones) in the horizontal and vertical. The position error is found for each of the shift zones, the larger the zone, the greater the scale of feature that will be matched. The user specifies the number of zones in each direction, and the observation sources to be used in each iteration of the process. It is important, then, that the user match the scale of the observations (in the `iuse` input switches) to the scale of the zones (in the `nizone` and `njzone` input variables).

2. Using the Program

2.1. Building the executable

Build the `arpsshift` executable using `makearps`:.

```
makearps arpsshift
```

switches for optimization (`-opt` *`n`*), debug (`-d`), and I/O options can be specified in the makearps command as would be done for building any of the other executables in the ARPS model system.

2.2. Setting-up the Data

The phase correction program uses the same data as ADAS. Prepare the data as you would to run ADAS. Specifically, convert surface, upper air and aircraft data into ASCII formatted files and run 88d2arps and/or nids2arps to prepare the radar data files. Currently the satellite data are not used, but that is planned for the near future. The satellite files would be prepared using `mci2arps` for MCIDAS formatted GOES visible and infrared satellite files. There are separate help files describing the data formats and how to run the radar pre-processing programs. Email the author for further information.

The program also utilizes the expected-error ADAS table files in ASCII format for each data source that is read in. These are contained in the directory `./data/adas` and are named *`source`*`.adastab`, where *`source`* is the name of the data source. These files should be set-up with the same expected-error data that one would use for running ADAS. There is a separate help file describing the format of the data source error files.

2.3. Running the Program

Once the data and error tables are prepared, edit the namelists in the input file, `arpsshift.input`. A list of the input variables and their default values follows in Section 5. Then, issue the following command from the ARPS top level directory:

```
bin/arpsshift < input/arpsshift.input >! arpsshift.out
```

2.4. Output Files

The phase correction program produces several output files:

1. Standard output. Provides a trace of the progress of the program. Examine this file for notifications (if any) of problems reading files, reports about observation counts and other statistics.
2. Phase-corrected ARPS history-dump files
   These files are `named` *`runname.fmt`*`000000` and *`runname.fmt`*`grdbas` where *`runname`* is the run name that is supplied in the input file, and *`fmt`* is the ARPS history dump format indicator (`bin`, `hdf`, `grb`, etc). These files can be used as input to plotting programs, as the background file (namelist variable named

`inifile`) for ADAS or ARPS-3DVAR analyses, or as the initial file (`inifile`) for the ARPS forecast model.

3. An ARPS history-dump file containing the phase correction vectors.
   This file is named *`runname.fmt`*`000001` where *`runname`* is the run name that is supplied in the input file, and *`fmt`* is the ARPS history dump format indicator. The only purpose of these files is to provide a file that can be read by the ARPS plotting program, `arpsplt`, to plot the phase correction vectors. The phase correction vectors are stored as the perturbation velocity components (*uprt* and *vprt*) in these history-dump files.
4. A binary file containing the phase correction vectors.
   This file is named *`runname.yyyymmdd.hhmmss`*`.shf,` where *`yyyymmdd.hhmmss`* is a date/time string . It contains a brief header and two 3-D arrays of the shift vector components, *ushift* and *vshift*. It is used as input to the ARPS model for gradually applying the phase corrections during the running of the forecast model. See Section 3.

3. Using the Phase Correction Vectors in Data Assimilation

The phase correction vectors can be used in data assimilation in two ways.

1. The phase-corrected fields can be used as input to ADAS (to apply any further adjustments and introduced additional cloud information) and then used to initialize the ARPS model. This can be done for a period of time, repeating at regular intervals as part of an intermittent data assimilation strategy.
2. The phase correction vectors can be read into ARPS and applied gradually over time. When this is done, the input file to `arpsshift` should be the same as the namelist variable *`inifile`* in ARPS (or use the corresponding restart file).

There is more information about using the phase correction information in data assimilation in the published papers (Brewster, 2003a,b) and in a separate Phase Correction Assimilation help file.

4. References

Brewster, K.A., 2003a: Phase-correction data assimilation and application to storm-scale numerical weather prediction. Part I: Method description and simulation testing. *Mon. Wea. Rev.*, **131**, 480-492.

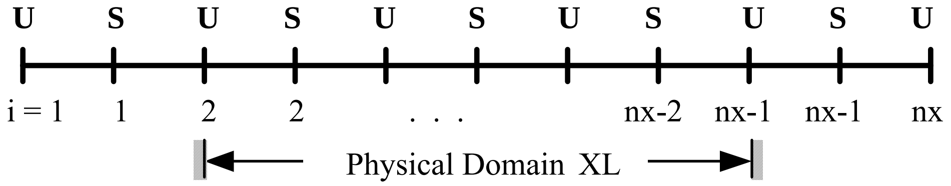
Brewster, K.A., 2003b: Phase-correction data assimilation and application to storm-scale numerical weather prediction. Part II: Application to a severe storm outbreak. *Mon. Wea. Rev.*, **131**, 493-507.

5. Input Parameters

The table in the following pages details the input parameters for the program. There is a separate help document for the phase correction file describing the input parameters for ARPS to drive the phase correction assimilation, if desired.

Model Dimension Parameters (&shfgrid_dims)

Parameter Values	Definition/Purpose	Options/Suggested
nx	Number of grid points including boundary points in the ξ (or x) direction. Physical dimension of the model domain in the x (or ξ) direction is $XL = (nx\text{-}3)\Delta x$. The grid structure in the x or ξ-direction is:	Any value ≥ 4. When run in 2-D *y-z* or 1-D single column mode, set *nx* to 4. In this case, only one grid zone is enclosed between the *physical* boundaries. ***nx* must match the *nx* contained in `inifile`, if that initialization option is used.**



where S indicates the scalar point and U the u-velocity point. Note that the model defines an extra *fake* zone outside the physical boundaries to facilitate the implementation of boundary conditions. Similarly for other boundaries.

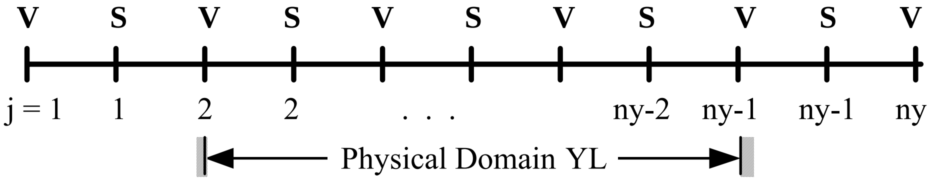
ny

Number of grid points including boundary points in the η (or y) direction.

Physical dimension of the model domain in the y (or η) direction is $YL = (ny-3)\Delta y$. The grid structure in the y or η-direction is:

Any value ≥ 4.
When run in 2-D *x-z* or 1-D single column mode, set *ny* to 4. In this case, only one grid zone is enclosed between the *physical* boundaries.

***ny* must match the *ny* contained in `inifile`, if that initialization option is used**.



where S indicates the scalar point and V the v-velocity point.

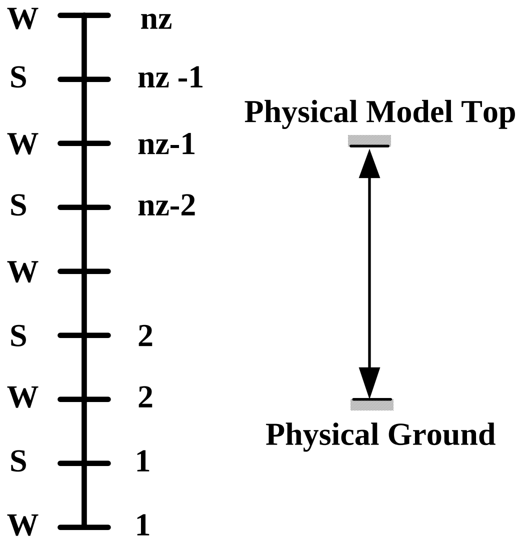
Number of grid points in the ζ (or z) direction.

Any value ≥ 4.

***nz* must match the *nz* contained in `inifile`, if that initialization option is used**.

nz

The model top boundary is at $z = (nz\text{-}3)\Delta\zeta$. where $\Delta\zeta$ is the vertical spacing of the computation grid, and corresponds to parameter *dz* in *arps.input.* The grid structure in the *z* or ζ-direction is:



where S indicates a scalar point and W a *w*-velocity point.

nzsoil

Number of soil levels.

nzsoil = 2

nzsoil should match the *nzsoil* contained in `inifile`, if that initialization option is used.

nstyp

Number of soil types.

nstyp = 4

nstyp should match the *nstyp* contained in `inifile`, if that initialization option is used.

Comment Strings on Current Run (&comment_lines)

Parameter Values	Definition/Purpose	Options/Suggested
nocmnt ***cmnt***	*cmnt* is an array of character strings containing user comments for this run. These comments will accompany the history data dumps. *nocmnt* is the number of comment strings to read in and carried in the data. The maximum array size is 50 and maximum string length is 80. CHARACTER (LEN=80) :: *cmnt(50)*	User specified strings of maximum length of 80.

Model Run Name (&jobname)

Parameter Values	Definition/Purpose	Options/Suggested
runname	A character string containing the identifying information for this run. The initial characters before a blank space or a comma will be used to construct file names of a number of output files, including the history and restart data. This string is also contained within the history data file and placed at the bottom of graphics pages produced by ARPSPLT. CHARACTER (LEN=80) :: *runname*.	Use up to 80 characters to identify this experiment and data files to be produced, *e.g.*, *runname='arpstest'*.

Model Initialization Parameters (&initialization)

Parameter	Definition/Purpose	Options/Suggested
Values		
initime	A character string specifying the calendar day and time (UTC) corresponding to the model clock time zero. *initime = 'yyyy-mn-dd.hh:mm:ss'*, where *yyyy* , *mn*, *dd*, *hh, mm* and *ss* are integers for year, month, day, hours, minutes and seconds. CHARACTER (LEN=19) :: *initime*	Character string in quotes. Choose to match the model time zero, not necessarily the time when the model is started or restarted. *e.g.*, *initime ='2003-01-01.12:00:00'*.
timeopt	Option to check the consistency of user specified time parameters, *initime* and *tstart*, with the time in the history data used initialize ARPS when *initopt*=3, and adjustment when desired.	= 1, warning on inconsistency and continue using *initime* and *tstart* = 2, warning on inconsistency and continue using data time = else, warning on inconsistency and stop. This option is default
initopt	Option for initializing the time dependent fields.	1 = initialize using analytic functions. 2 = initialize with a restart file from a previous run. 3 = initializing from history format initial condition (IC) data set

pt0opt	Option for specifying the initial potential temperature perturbation when *initopt*=1.	0 = no added initial perturbation. 1 = ellipsoidal bubble (defined below). 2 = random perturbations. 3 = random perturbations symmetric about central *x-z* and *y-z* planes. 4 = half vertical wave length bubble (see below). 5 = Soup-can-shaped potential temperature perturbation. 6=ellipsoidal bubble with perturbation specified in *T* rather than $\boldsymbol{\theta}$**.**
ptpert0	The magnitude (K) of the initial potential temperature perturbation for *initopt*=1 case and all options of *pt0opt.*	Typically a few degrees. *ptpert0* = θ_0 in next block.
pt0ctrx ***pt0ctry*** ***pt0ctrz*** ***pt0radx*** ***pt0rady*** ***pt0radz***	The center location (m) and the radii (m) of the initial bubble perturbation in *x, y* and *z* direction. For *pt0opt* =1, the ellipsoidal bubble is defined by $\Delta\theta = \theta_0 \cos^2(\pi\beta/2) \;\; (for\ \beta \le 1)$ where θ_0 (= *ptpert0*) is the amplitude at the center of the disturbance and β is a non-dimensional radius given by: $\beta = \sqrt{\left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{y - y_c}{y_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2}$ where $x_c = pt0ctrx,\ y_c = pt0ctry,\ z_c = pt0ctrz$ $x_r = pt0radx,\ y_r = pt0rady,\ z_r = pt0radz$ For *pt0opt*=4, a 2-D bubble is defined by: $\Delta\theta = \theta_0 \frac{\sin(\pi z/H)}{1+(x-x_c)^2/x_r^2}$ where $H = (nz-3)*\Delta\zeta$ (*dz* in the code).	To place the bubble at the center of model domain, set *pt0ctrx = (nx-3)*Δx/2*, *pt0ctry = (ny-3)*Δy/2.* A negative value of radius implies an infinite extent of the bubble in that direction. Therefore, setting *pt0rady* = -1 gives a bubble perturbation independent of *y*.

sndfile	Name of the sounding file used to initialize the model base state variables when *initopt*=1 and *inibasopt=1*. A sounding file must be defined when *initopt*=1. The sounding file format is defined in Section 8.4.1. CHARACTER (LEN=80) :: *sndfile*.	Character string in quotes. Not to exceed 80 characters in length. *e.g., sndfile='may20.snd'.*
rstinf	Name of the restart file used only when *initopt*=2. The restart file should always be in the machine native binary format. CHARACTER (LEN=80) :: *rstinf*.	Character string in quotes. *e.g.*, *rstinf='arpstest.rst003600'.*
inifmt	Format flag for the initial condition data files *inifile* and *inibgf*. Standard ARPS history data dump formats are used by these initial data files although not all formats are available for the initial condition purpose.	1 = unformatted binary. 2 = ASCII. 3 = NCSA HDF Version 4. 10 = GRIB.
inisplited	Option flag indicating whether the input data files are in single joined form or in split form as created by *splitfiles*. Valid for MPI mode and *initopt* = 3 only.	0 = in single joined form. This option only works when the input files are in binary or HDF format. 1 = in split form
inifile	Name of the initial condition data file containing time-dependent variables, and, possibly, the base-state arrays and grid coordinate arrays depending on the values of flags inside the data file. When both the base state and grid coordinate arrays are present in this file, file *inibgf* will not be read. Used only when *initopt*=3. CHARACTER (LEN=80) :: *inifile*.	Character string in quotes. *e.g.,* *inifile='arpstest.bin003600'* .
inibgf	Name of the initial condition data file containing time-independent base state and grid coordinate arrays. Used only when *initopt*=3. CHARACTER (LEN=80) :: *inibgf*.	Character string in quotes. *e.g.,* *inibgf='arpstest.bingrdbas'* .

inibasopt	Option for initializing the base state variables for *initopt=1* case.	1 = initialize the base state using a single sounding. 2 = isentropic atmosphere. 3 = isothermal atmosphere. 4 = atmosphere with a hard-coded constant static stability (N=10^{-2} s^{-1}). 5 = an analytic thermodynamic sounding profile after Weisman and Klemp (1982) (Section 8.4.5). For *inibasopt* ≠1, additional parameters are hardwired inside subroutine INIBASE.
viniopt	Option for specifying the initial and base-state wind profiles internally when *inibasopt* ≠1.	1 = *ubar=ubar0, vbar=vbar0* (see below). 2 = user-specified profiles for *ubar* and *vbar.* To use option 2, a user has to edit subroutine INIBASE. The default values in the code are 0.0 for *ubar* and *vbar*.
ubar0	Constant u-velocity (m/s) for the initial and base-state wind when *inibasopt* ≠ 1 and *viniopt* = 1.	User specified.
vbar0	Constant v-velocity (m/s) for the initial and base-state wind when *inibasopt* ≠ 1 and *viniopt* = 1.	User specified.

Parameters For Model Terrain Specification (&terrain)

Parameter Values	Definition/Purpose	Options/Suggested
ternopt	Model terrain option. When *ternopt* =0, terrain height is zero and codes related to terrain formulations are switched off to improve efficiency.	0 = no terrain, ground is flat. 1 = analytic mountain. 2 = terrain data read from terrain data file *terndat*.
mntopt	Option for analytic mountain type. The bell-shaped mountain is given by $h = \dfrac{h_m}{1+\left[(x-x_c)/x_r\right]^2+\left[(y-y_c)/y_r\right]^2}$ where h_m is the mountain height, and, x_r and y_r are the half-widths in x and y directions respectively.	1 = Bell-shaped mountain. 2 = User specified mount profile (code change required).
hmount	Maximum height (m) of the analytic mountain above *zrefsfc* level. Used only when *ternopt* = 1. *hmount* = h_m in the above equation.	$0 \le hmount < (nz-3)*d\zeta = z_{top}$.
mntwidx ***mntwidy***	The half-width (m) of the analytic mountain in *x* and *y* directions, respectively. Used only when *ternopt* =1. *mntwidx* = x_r and *mntwidy*=y_r in the above equation.	A negative value indicates infinite extent in that direction, therefore the mountain collapses to 2-D or even 1-D. When the model is run in 2-D or 1-D mode, one or both of these values become irrelevant.
mntctrx ***mntctry***	The center location (m) of the analytic mountain in *x* and *y* directions, respectively. Used only when *ternopt =1.* *mntctrx* = x_c and *mntctry*=y_c in the above equation.	When the model is run in 2-D or 1-D mode, one or both of these values become irrelevant.
terndta	Name of the terrain data file. Used only by option *ternopt*=2. CHARACTER (LEN=80) :: *terndta*	A character string in quotes. *e.g.*, *terndta* = '*arpstern.dat*'.

ternfmt	Integer format flag for terrain data file *terndta*.	1 = binary 3 = HDF

Model Grid Setup Parameters (&grid)

Parameter Values	Definition/Purpose	Options/Suggested
dx	Grid spacing (m) in ξ-direction ($\Delta\xi$). In the code, *dx* is used to represent $\Delta\xi$ for notational convenience.	Problem dependent.
dy	Grid spacing (m) in η-direction ($\Delta\eta$). In the code, *dy* is used to represent $\Delta\eta$ for notational convenience.	Problem dependent.
dz	Grid spacing (m) in the ζ-direction ($\Delta\zeta$) in computational space. In the code, *dz* is used to represent $\Delta\zeta$ for notational convenience. *dz* is also the average physical spacing in the vertical before the terrain transformation.	Problem dependent.
strhopt	Option for vertical grid stretching.	0 = no stretching. Uniform vertical grid used 1 = cubic function used to define *dz* (Eq. (7.3.5.)). 2 = tanh function used to define *dz* (Eq. (7.3.6.)).
dzmin	Minimum grid spacing (m) in the vertical direction in physical space when vertical stretching is invoked *dzmin* and *dz* are used to construct a vertically stretched grid according to Eqs. (7.3.4). *dzmin* is reset to *dz* if *strhopt*=0.	A value less than or equal to *dz*.

zrefsfc	Reference height (m) of the model bottom boundary. *zrefsfc* = z_0 in Eq. (7.3.2).	When the entire model domain is elevated, *zrefsfc* can be chosen to be the minimum elevation of the terrain height. 0.0 m recommended.
dlayer1 ***dlayer2***	*dlayer1* (m) = the depth of layer 1. *dlayer2* (m) = the depth of layer 2. The vertical domain is divided into 3 layers: layer 1 (the lowest level) has a uniform resolution of *dzmin*.; layer 2 (which is just above layer 1) has a grid spacing that stretches from *dzmin* upwards; layer 3 (top layer) has a spacing of approximately that at the top of layer 2. *dlayer1* = D_1 and *dlayer2* = D_2 in Eqs. (7.3.4.). See Fig. 7.3 for further information. Used only when *strhopt*≠0.	$0 \le dlayer1 < z_{top}-z_{0}.$ $0 \le dlayer2 < z_{top}-z_{0},$ $0 \le dlayer1 + dlayer2 < z_{top}-z_{0},$ $dlayer2 = min(dlayer2, z_{top}-dlayer1-z_{0}).$ *dlayer1=0*, *dlayer2*=10^5 m recommended.
strhtune	Tuning factor (α) for stretching option 2. See Eq. (7.3.6).	0.2 ≤ *strhtune* ≤ 5.0. A larger value gives a more linearly stretched grid. Default value is 1.0.
zflat	The height (m) at which grid levels become flat in the terrain-following coordinate transformation (see Fig. 7.6). Note: This parameter has nothing to do with the setup of stretching.	h_m < *zflat* ≤ z_{top}, where h_m is the mountain top height. If one desires the coordinate surface to become flat at the model top, set *zflat* to a value larger than *ztop*, the model will reset *zflat* = *ztop*.
ctrlat ***ctrlon***	The latitude (degrees north) and longitude (degrees east) at the center of model domain.	Anywhere on the globe. -90 ≤ *ctrlat* ≤ 90 -180 ≤ *ctrlon* ≤ 180

Map Projection Parameters
(&projection)

Parameter Values	Definition/Purpose	Options/Suggested
mapproj	Map projection option. In general, polar stereographic is a better choice for hemispheric or polar applications, Lambert is better for mid-latitudes and Mercator better for tropical applications.	0 = no map projection. 1 = northern hemisphere polar stereographic projection. -1 = southern hemisphere polar stereographic projection. 2 = northern hemisphere Lambert conformal projection. -2 = southern hemisphere Lambert conformal projection. 3 = Mercator projection 4 = longitude-latitude grid (not available for model runs). A projection that has a map factor close to 1 is recommended.
trulat1 ***trulat2***	1st and 2nd true latitude (degrees north) of map projection, respectively. *trulat2* is used only by Lambert projection.	True latitudes near the center of model domain are recommended.
trulon	True longitude (degrees east) of map projection. This longitude line corresponds to the -*y* axis of the model grid.	A longitude near the center of the model domain is recommended.
sclfct	Distance scaling factor.	sclfct=1.

Time Integration Control Parameters (&timestep)

Parameter Values	Definition/Purpose	Options/Suggested
tstart	The starting time (s) of a model run. *tstart* is used only when *initopt*=1 or 3. For *initopt*=2 (model restart), *tstart* is reset to the time of restart data.	For a run starting from an external data set (*initopt*=3), *tstart* should be set so that model time zero matches the real time set by string *initime*. 0.0 is the default value. This parameter is only used to verify the data time stamp. See documentation of parameter `timeopt`.

Model I/O Control Parameters for History Dumps (&history_dump)

Parameter Values	Definition/Purpose	Options/Suggested
hdmpopt	Option parameter for specifying the times history dumps are created.	1 = dumps at equal time intervals, start from *tstrdmp*; 2 = dumps at times specified by user.

dmp_out_ joined	Flag indicating if, when the model is run in distributed-memory-parallel mode using MPI, the output fields from different processors will be gathered and joined first before being written out (into single files). NOTE: See the restriction for *max_fopen* flag when *dmp_out_joined* = 1. Joined dumps only work for GrADS, binary and HDF format.	0 = each processor writes out its own portion of data, the output will be joined together using *joinfiles* program. 1 = the output fields from different processors will be gathered and joined first before being written out. The *joinfiles* step is no longer needed.
thisdmp	Time interval (s) between history data dumps when hdmpopt=1. It is reset to the nearest integer multiple of d*tbig* when necessary.	0 ≤ *thisdmp* . Setting *thisdmp* to zero switches off history dump file writing.
hdmpfmt	Format flag for ARPS history data dumps. Format 1 is the fastest. Format 3 with compression is most compact and portable. ASCII is portable but the files are very large. Vis5D and GrADS are formats for 3D visualization and GrADAS graphic plotting, respectively. GRIB format will be dropped and the support for NetCDF and HDF 5 support may be added in the future. See Section 10.1 for detailed descriptions.	0 = no history dump. 1 = unformatted binary (whenever available, compiler options are included in *makearps* so that big endian format is used even on little endian platforms such as those using Intel processors). 2 = ASCII. 3 = NCSA HDF version 4. 6 = binary with grid point skipping in partial domain. 7 = NetCDF (not available) 8 = Packed 16-bit NetCDF (not available). 9 = GrADS. 10 = GRIB 11 = Vis5D format.
thisdmp	Time interval (s) between history data dumps when hdmpopt=1. It is reset to the nearest integer multiple of d*tbig* when necessary.	0 ≤ *thisdmp* . Setting *thisdmp* to zero switches off history dump file writing.

grbpkbit	Number of bits to use in packing GRIB data	= 32, full precision retained on 32 bit machines; = 16, reduced precision to same disk space. 16 is sufficient for general purposes.

hdfcompr	HDF4 compression option (for *hdmpfmt*=3).
	It is reset to the nearest integer multiple of d*tbig* when necessary.
	Note that only options 0-2 work on Cray platforms.
		0 = (default), no compression;
		1 = fast gzip compression;
		2 = high gzip compression;
		3 = adaptive or skipping Huffman compression;
		4-7 = as above plus mapping reals to 16 bit integers.

tstrtdmp	Time (s) at which history dumps start.	User's choice.
numhdmp	The number of history dump times specified by user for *hdmpopt*=2.	Choose 0 if no history data dump is desired.
hdmptim	Real array, specifying the times when history dumps are created. Used when *hdmpopt*=2. The maximum array length permitted is 100.	

Other Model I/O Control Parameters (&output)

dirname	Name of the directory into which output data are written. CHARACTER (LEN=80) :: *dirname*.	Character string in quotes. Not to exceed 80 characters in length. Default ='./' for current directory.
exbcdmp	Option to write external boundary condition files at the time of history dumps.	0 = no EXBC dump. 1 = EXBC dump in binary format 3 = dumps in HDF format.

exbchdfcompr	EXBC HDF format compression option (for *exbcdmp*=3).
	Note that only options 0-2 work on Cray platforms.
		0 = no compression;
		1 = fast gzip compression;
		2 = high gzip compression;
		3 = adaptive or skipping Huffman compression;
		4-7 = corresponding to the above plus mapping real numbers to 16 bit integers.

exdtadmp	Option to write out the history format files that contain the external data arrays interpolated to the current model time. These files will be created at the same time as the history data files.	0 = the data file is not generated. 1 = the data file is created. When *lbcopt* $\neq$ 2, it is automatically reset to 0.
filcmprs	Switch to automatically compress history data files, by using UNIX file compression utility **compress** or **gzip**.	0 = no compression. 1 = compression. Compression can slow down model execution but can reduce the disk usage.
readyfl	Option to create a marker file (same name is the history dump but with "_ready" appended to the end) to indicate the completion of the writing of the history file for the purpose of automatic detection.	0 = do not create a ready file. 1 = create a ready file;
grdout	Option to write out grid coordinate arrays (*x, y, z* and *zp*) together with the time dependent arrays in the history file. Grid arrays are always written into a separate base/grid data file regardless the value of *grdout*.	0 = no. 1 = yes. 0 recommended.
basout	Option to write base state arrays ($\overline{u}$, $\overline{v}$, $\overline{w}$, $\overline{p}$, $\overline{\theta}$ and $\overline{q}_v$) together with the time dependent arrays in the history file. Base state arrays are always written into a separate base/grid data file regardless the value of *basout*.	0 = no. 1 = yes. 0 recommended.

varout	Option to write time dependent model dynamic variables (u, v, w, θ and p) into the history file. Note that the winds are dumped out in the grid-relative instead of ground-relative framework.	0 = no. 1 = yes. 1 recommended.
mstout	Option to write moisture variable arrays (q_v, q_c and q_r) into the history file.	0 = no. 1 = yes. Set to 0 if the model run is dry. When *mstout*=0, ice variables will not be written either.
iceout	Option to write ice variable arrays (q_i, q_s and q_h) into the history file.	0 = no. 1 = yes. Set to 0 if the model run is dry.
tkeout	Option to write turbulent kinetic energy into the history file.	0 = no. 1 = yes. TKE is non-zero only when *tmixopt* = 4.
trbout	Option to write turbulent mixing coefficient array (k_m) into the history file.	0 = no. 1 = yes.
rainout	Option to write surface accumulated rainfall arrays into the history file.	0 = no. 1 = yes. Set to 0 if the model run is dry.
sfcout	Option to write soil model time-dependent arrays into the history file.	0 = no. 1 = yes. Set to 0 if surface physics is off.
landout	Option to write surface characteristics arrays into the history file.	0 = no. 1 = yes.
prcout	Option to write precipitation rates into the history file.	0 = no. 1 = yes.
radout	Option to write radiation arrays in history dump	0 = no. 1 = yes.

flxout	Option to write surface fluxes in history dump	0 = no. 1 = yes.
qcexout	Option for writing array *qc* in EXBC data dump when *exbcdmp*≠0.	0 = no. 1 = yes.
qrexout	Option for writing array *qr* in EXBC data dump when *exbcdmp*≠0.	0 = no. 1 = yes.
qiexout	Option for writing array *qi* in EXBC data dump when *exbcdmp*≠0.	0 = no. 1 = yes.
qsexout	Option for writing array *qs* in EXBC data dump when *exbcdmp*≠0.	0 = no. 1 = yes.
qhexout	Option for writing array *qh* in EXBC data dump when *exbcdmp*≠0.	0 = no. 1 = yes.
sfcdmp	Option to create surface characteristics data file.	0 = no file generated 1 = binary format 3 = (uncompressed) HDF
soildmp	Option to create soil model initial condition format data files.	0 = no file generated 1 = binary format 3 = (uncompressed) HDF
terndmp	Option to write out an ARPS terrain data file.	0 = no file generated 1 = binary format 3 = (uncompressed) HDF
trstout	Time interval (s) between restart data dumps. Restart data are in the machine native binary format.	It is recommended that restart data be written at the end of a model run and at some intermediate times.
tmaxmin	Time interval between the printouts of domain-wide maximum and minimum values of variables. Output can be plotted by program ARPSPLTMAX.	User discretion.
tfmtprt	Time interval (s) between formatted variable printouts in standard output file.	0 ≤ *tfmtprt* . Setting *tfmtprt* to zero switches off formatted printing.

tenergy	Time interval between printouts of domain energy statistics (only partially maintained).	User discretion.
imgopt	Option for the generation of 2-D 8-bit raster image files in NCSA HDF format.	0 = no image file. 1 = image files written. The indices of the slices are hardwired in subroutine OUTPUT.
timgdmp	Time interval (s) between creation of HDF-image files.	User discretion.
pltopt	Option for generating graphic plots during model run.	Not implemented.
tplots	Time interval (s) between graphic plot generation.	Not implemented.

Debug Parameters (&Debug)

Parameter Values	**Definition/Purpose**	**Options/Suggested**
lvldbg	Option for printing debug information. Currently ARPS prints certain arrays in tabular form.	0 = no printing. 1= model variables in large time step. 2 = add forcing terms in large time step. 3 = add variables in small time step. 4 = add forcing terms in small time step. 5 = add individual forcing terms and other miscellaneous information.

ARPSShift Constants (&shift_const)

Parameter	**Definition/Purpose**	**Options/Suggested Values**

nshfpass	Number of iterations of the grid shifting algorithm	Typical: *nshfpassl*=4
sprdist	Super-ob distance (m). Surface observations closer than *sprdist* will be combined into a single-superob before beginning the analysis.	Scale (*dx*) and observation dataset dependent, approximately 15000 m. (15 km)
sfcqcrng	Scale distance (m) for the Barnes weighting function used in the quality control of surface data.	Data spacing dependent. Generally: 30.E03 < *sfcqcrng* < 100.E03 m
wlim	Smallest value of horizontal correlation (ND) that will contribute to the analysis. This serves to define a cut-off radius, outside of which an observation has no influence. When the dataset is very large (10^4 observations), increasing *wlim* (decreasing cut-off radius) can improve execution time.	1.0E-03
nbaksmth	Number of passes of a 9-point filter applied to background field before beginning processing.	nbaksmth=3
nshfsmth	Number of passes of a 9-point filter applied to shift vectors before output and application to forecast fields..	nshfsmth=3
noutsmth	Number of passes of a 9-point filter applied to position-adjusted forecast fields before they before they are updated.	noutsmth=0
minkdat	Minimum number of data points in grid zone to calculate zone shift vectors.	minkdat=3

slnratio	Weighting ratio relative to grid size.	slnratio=0.5
reflmin	Minimum radar reflectivity to affect grid shift algorithm (dBZ).	reflmin=8.0
spradopt	Superadiabatic adjustment option. Option for adjusting temperature to remove any superadiabatic layers. Proceeding from bottom up provides greater fidelity to the surface observations but may overwarm boundary layer.	0=no adjustment 1=check proceeding from top down to bottom 2=check proceeding from the bottom up to top *spradj*=0 recommended for analysis-only applications *spradj*=1 recommended for model initializa
hydradj	Option for adjusting pressure to balance temperature field to create a balanced state for model initialization. Formerly known as *boycor*.	0=no pressure adjust. 1=use model eqs, beginning at bottom 2=use model eqs, beginning at top 3=hydrostatic pressure, beginning at bottom *hydradj*=0 recommended
wndadj	Option for adjusting wind field after horizontal winds have been analyzed.	0=no wind adjust. 1=*w* is set so *wcont*=0 no changes to *u*, *v* 2=*w* is set using horiz divergence and O'Brien method for top bc see *obropt* 3=as in 2, but *u,v* are relaxed to create zero 3-D divergence *windadj*=2 recommended

obropt	Option for distribution of horizontal divergence error in O'Brien adjustment of vertical velocities (to enforce vertical boundary conditions on *w*). Add 10 to values at right to adjust vertical velocities so that *w=0* at physical height *obrzero***.** Otherwise the scheme forces *w=0* at the top of the model physical domain.	0= No O'Brien adjust. 1=linearly in model z 2=linearly in physical z 3=linearly in theta *obropt*=12 recommended
obrzero	Height (m) at which the O'Brien scheme enforces zero *w*. This should roughly correspond to the tropopause, and be consistent with the bottom of the Rayleigh damping layer (*zbrdmp*) if used for model initialization	Grid dependent.

ARPSShift Grid Shift Zones (&shift_backerf)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nizone	Number of zones in the x direction for each grid shift iteration.	Example: *nizone()*=4,8,15,25
njzone	Number of zones in the y direction for each grid shift iteration.	Example: *njzone()*=4,8,15,25
nkzone	Number of zones in the z direction for each grid shift iteration.	Example: *njzone()*=4,4,4,4

ARPSShift VariableWeights (&shift_backerf)

Parameter	**Definition/Purpose**	**Options/Suggested Values**

wgtvar	Relative weighting for each variable (in addition to normalization to account for standard error of observation).	Example:
	index variable	
	1 u-wind	*wgtvar(1)* = 5.0,
	2 v-wind	*wgtvar(2)* = 5.0,
	3 pressure	*wgtvar(3)* = 0.0,
	4 temperature	*wgtvar(4)* = 5.0,
	5 qv	*wgtvar(5)* = 3.0,
	6 radar reflectivity	*wgtvar(6)* = 5.0,
	7 radar radial velocity	*wgtvar(7)* = 1.0,

ARPSShift Background Error File (&shift_backerf)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
backerrfil	Name of background error file character*132	Typical: *backerrfil*= 'ruc3herr.adastab'

ARPSShift Constants (&shift_const)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nshfpass	Number of single-level data files. Single level data include surface data and pilot report data.	Data dependent. Typical: 1 or more *nsngfil* ≤ *mx_sng_file* (*mx_sng_file* is set in **adas.inc**)
sfcqcrng	Scale distance (m) for the Barnes weighting function used in the quality control of surface data.	Data spacing dependent. Generally: 30.E03 < *sfcqcrng* < 100.E03 m

ADAS Adjustment Options
(&adjust)

ADAS Radar Options
(&adas_radaropt)

Parameter	Definition/Purpose	Options/Suggested Values
raduvobs	Option for using Doppler radial winds in analysis step	0 = do not use radial winds 1 = use radial winds
radrhobs	Switch to create pseudo-obs of high relative humidity for use in *qv* analysis (effect will be spread in space) where reflectivity is high	0 = do not create high-RH obs from reflectivity data 1 = create obs *radrhobs*=0 recommended
sfcqcrng	Name of data file used for time consistency check of respective *sngfname* data. Dataset of the same type from a previous time period. Specify a dummy filename if no such data are available. character*132	Typical: *sngtimchk(1)*= 'jun08/951591400.lso'
srcsng (isrc)	Following are repeated for each data source in the single-level data file(s). Note that a single-level file can contain one or more sources. Largest index, *isrc*, used must be less than or equal to *nsrc_sng* (*nsrc_sng* is set in **adas.inc**). Name of data source. These must match the source name(s) in the data file(s). character*8	Typical: *srcsng(1)*='SA'

sngerrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *sngerrfil(1)*= 'saoerr.adastab'
iusesng (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iusesng(1,1)*, *iusesng(1,2)*...	0: Do not use data from this source on this pass. 1: use data Example: *iusesng(1,1)*=0, *iusesng(1,2)*=1, *iusesng(1,3)*=0

ADAS Single-Level Data Specifications (&adas_sng)

Parameter	Definition/Purpose	Options/Suggested Values
nsngfil	Number of single-level data files. Single level data include surface data and pilot report data.	Data dependent. Typical: 1 or more *nsngfil* ≤ *mx_sng_file* (*mx_sng_file* is set in **adas.inc**)
sngfname (ifile)	Name of single-level data file Repeat for *nsngfil*s. character*132	Typical: *sngfname(1)*= 'jun08/951591500.lso'
sngtimchk (ifile)	Name of data file used for time consistency check of respective *sngfname* data. Dataset of the same type from a previous time period. Specify a dummy filename if no such data are available. character*132	Typical: *sngtimchk(1)*= 'jun08/951591400.lso'

srcsng (isrc)	Following are repeated for each data source in the single-level data file(s). Note that a single-level file can contain one or more sources. Largest index, *isrc*, used must be less than or equal to *nsrc_sng* (*nsrc_sng* is set in **adas.inc**). Name of data source. These must match the source name(s) in the data file(s). character*8	Typical: *srcsng(1)=*'SA'
sngerrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *sngerrfil(1)=* 'saoerr.adastab'
iusesng (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iusesng(1,1)*, *iusesng(1,2)*...	0: Do not use data from this source on this pass. 1: use data Example: *iusesng(1,1)*=0, *iusesng(1,2)*=1, *iusesng(1,3)*=0

ADAS Single-Level Data Specifications (&adas_sng)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nsngfil	Number of single-level data files. Single level data include surface data and pilot report data.	Data dependent. Typical: 1 or more *nsngfil* ≤ *mx_sng_file* (*mx_sng_file* is set in **adas.inc**)
sngfname (ifile)	Name of single-level data file Repeat for *nsngfil*s. character*132	Typical: *sngfname(1)=* 'jun08/951591500.lso'

sngtimchk (ifile)	Name of data file used for time consistency check of respective *sngfname* data. Dataset of the same type from a previous time period. Specify a dummy filename if no such data are available. character*132	Typical: *sngtimchk(1)*= 'jun08/951591400.lso'
srcsng (isrc)	Following are repeated for each data source in the single-level data file(s). Note that a single-level file can contain one or more sources. Largest index, *isrc*, used must be less than or equal to *nsrc_sng* (*nsrc_sng* is set in **adas.inc**). Name of data source. These must match the source name(s) in the data file(s). character*8	Typical: *srcsng(1)*='SA'
sngerrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *sngerrfil(1)*= 'saoerr.adastab'
iusesng (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iusesng(1,1)*, *iusesng(1,2)*...	0: Do not use data from this source on this pass. 1: use data Example: *iusesng(1,1)*=0, *iusesng(1,2)*=1, *iusesng(1,3)*=0

ADAS Single-Level Data Specifications (&adas_sng)

Parameter **Definition/Purpose** **Options/Suggested Values**

nsngfil	Number of single-level data files. Single level data include surface data and pilot report data.	Data dependent. Typical: 1 or more *nsngfil* ≤ *mx_sng_file* (*mx_sng_file* is set in **adas.inc**)
sngfname (ifile)	Name of single-level data file Repeat for *nsngfil*s. character*132	Typical: *sngfname(1)*= 'jun08/951591500.lso'
sngtimchk (ifile)	Name of data file used for time consistency check of respective *sngfname* data. Dataset of the same type from a previous time period. Specify a dummy filename if no such data are available. character*132	Typical: *sngtimchk(1)*= 'jun08/951591400.lso'
srcsng (isrc)	Following are repeated for each data source in the single-level data file(s). Note that a single-level file can contain one or more sources. Largest index, *isrc*, used must be less than or equal to *nsrc_sng* (*nsrc_sng* is set in **adas.inc**). Name of data source. These must match the source name(s) in the data file(s). character*8	Typical: *srcsng(1)*='SA'
sngerrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *sngerrfil(1)*= 'saoerr.adastab'

iusesng (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iusesng(1,1)*, *iusesng(1,2)*...	0: Do not use data from this source on this pass. 1: use data Example: *iusesng(1,1)*=0, *iusesng(1,2)*=1, *iusesng(1,3)*=0

ADAS Multiple-Level Data Specifications (&adas_ua)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nuafil	Number of multiple-level data files. Multiple level data include radiosonde data and profiler data.	Data dependent. Typical: 1 or more *nuafil* ≤ *mx_ua_file* (*mx_ua_file* is set in **adas.inc**)
uafname (ifile)	Name of multiple-level data file Repeat for *nuafils*. character*132	Typical: *uafname(1)*= 'jun08/951591500.snd'
srcua (isrc)	Following are repeated for each data source in the multiple-level data file(s). Note that a multiple-level file usually contains one source, but more than one file may have the same source. Largest index, *isrc*, used must be less than or equal to *nsrc_ua*. (*nsrc_ua* is set in **adas.inc**). Name of data source. character*8	Typical: *srcua(1)*="NWS RAOB'
uaerrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *uaerrfil(1)*= 'snderr.adastab'

iuseua (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iuseua(1,1)*, *iuseua(1,2)*...	0: Do not use data from this source on this pass. 1: use data Data dependent Example: *iuseua(1,1)*=1, *iuseua(1,2)*=0, *iuseua(1,3)*=0

ADAS Radar Data Specifications (&adas_radar)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nradfil	Number of remapped radar data files. Remapped radar data files are created by mapping raw data onto a regular grid, which need not match the final analysis grid.	Data dependent. Typical: 0 or more *nradffl* ≤ *mx_rad_file* (*mx_rad_file* is set in **adas.inc**)
radfname (ifile)	Name of radar data file(s) Repeat for nradfils. character*132	Typical: *radfname(1)*= 'KTLX.950507.1756'
srcrad (isrc)	Following are repeated for each data source in the radar data file(s). **Note**: that a radar data file contains one source, but more than one file may have the same source. Largest index, *isrc*, used must be less than or equal to *nsrc_rad* (*nsrc_rad* is set in **adas.inc**). Name of data source. character*8	Typical: *srcrad(1)*='88D-AII' *srcrad(2)*='NIDS'

raderrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *raderrfil(1)*= 'data/adas/88derr.adastab' *raderrfil(2)*= 'data/adas/nidserr.adastab'
iuserad (isrc, ipass)	Integer switch indicating whether the radar data source should be used on each pass. A switch is required for each pass, i.e. *iuserad(1,1)*, *iuserad(1,2)*...	0: Do not use data from this source on this pass. 1: use data Data dependent Example: *iuserad(1,1)*=0, *iuserad(1,2)*=0, *iuserad(1,3)*=1

ADAS Retrieved Radar Data Specifications (&adas_retrieval)

Parameter	**Definition/Purpose**	**Options/Suggested Values**
nretfil	Number of radar retrieved data files. Retrieved data files are created by separate radar retrieval programs.	Data dependent. Typical: 0 or more *nretfil* ≤ *mx_ret_file* (*mx_ret_file* is set in **adas.inc**)
retfname (ifile)	Name of radar retrieval data file Repeat for *nretfil*s. character*132	Typical: *retfname(1)*= 'KTLXret.960526.1700'

srcret (isrc)	Following are repeated for each data source in the retrieval data file(s). Note that a retrieval data file contains one source, but more than one file may have the same source. Largest index, *isrc*, used must be less than or equal to *nsrc_ua*. (*nsrc_ret* is set in **adas.inc**). Name of data source. character*8	Typical: *srcret(1)*='88D-RET'
reterrfil (isrc)	Name of file containing error specification table for each source. character*132	Typical: *reterrfil(1)*= '88dreterr.adastab'
iusesret (isrc, ipass)	Integer switch indicating whether the data source should be used on each pass. A switch is required for each pass, i.e. *iuseret(1,1)*, *iuseret(1,2)*...	0: Do not use data from this source on this pass. 1: use data Data dependent Typical: *iuseret(1,1)*=0, *iuseret(1,2)*=0, *iuseret(1,3)*=1