

## ADAPT User's Guide

11 February 2021

The Atmospheric Dispersion And Parallel Transport (ADAPT) program can be used to evaluate various OMP parallel statements in the context of atmospheric dispersion calculations. Program elements including reading ARL packed meteorological data files, converting meteorological data to computational units, computing particle advection and dispersion, and the accumulation of particle mass on a concentration grid. The current version only supports the use of conformal projection meteorological data on sigma surfaces for transport and TKE fields for mixing. The computer code and data files described here are distributed under the GNU LGPL license, <https://www.gnu.org/licenses/lgpl-3.0.en.html>.

### Compilation

Use the following statements to compile the code with gfortran:

```
OPT="-ffree-form -fconvert=big-endian -frecord-marker=4"  
MAP="libcmapf.a"  
gfortran -oadapt -fopenmp ${OPT} ${MAP} adapt.f
```

The compilation requires the CMAPF routines (version 1.0) to transform between earth and model grid coordinates. Version 1.0 covers the standard conformal map projections centered at the North and South Pole, namely the Polar Stereographic, the Mercator, and the Lambert Conformal projections. See <https://www.arl.noaa.gov/hysplit/cmapf-mapping-routines/>. Version 1.0 is described more fully in an article, "Conformal Map Transformations for Meteorological Modelers", by Albion D. Taylor, published in the February, 1997 issue of Computers and Geosciences (v23, no 1) and can be downloaded from [https://www.arl.noaa.gov/wp\\_arl/wp-content/uploads/utilities/cmap/cmapf.v1\\_0.tar.gz](https://www.arl.noaa.gov/wp_arl/wp-content/uploads/utilities/cmap/cmapf.v1_0.tar.gz) <https://www.iamg.org/documents/oldftp/VOL23/v23-1-5.tar.Z>

### Obtaining meteorological data

Calculations can be made using WRF, NAMS, or HRRR meteorological data processed for use by HYSPLIT. This WRF data archive is described in more detail in "A Long-Term WRF Meteorological Archive for Dispersion Simulations: Application to Controlled Tracer Experiments", by F. Ngan and A.F. Stein, J. Appl. Meteorology and Climatology, Aug. 2017, 2203-2200, Vol. 56 (DOI: 10.1175/JAMC-D-16-0345.1). Daily WRF data files can be downloaded from <ftp://arlftp.arlhq.noaa.gov/pub/archives/{avg|inst}> as instantaneous or time-averaged data files. WRF output that has been converted to the ARL format in these directories follows the file name convention of wrfout\_d01\_ccyymmdd.ARL. Only the time-averaged data files contain the TKE field required for dispersion calculations, otherwise ADAPT will estimate the TKE. Note that the HRRR archived data (a variant of WRF) can also be found on the ARL server. In fact, any conformal sigma level data set should be compatible for computations. The sample calculation discussed below can be performed using the meteorological data provided with the HYSPLIT tutorial (<https://ready.arl.noaa.gov/data/web/workshop/2020/index.html>).

## Processing ADAPT output files

There are four types of output files created by ADAPT and three of them can be processed by software provided with the HYSPLIT package. Air concentrations are output in the HYSPLIT concentration binary format, directly in DATEM format, or as a SVG graphic. Trajectory output is in the HYSPLIT trajectory endpoints file format. Provided with the code is a sample SETUP.CFG that can be used to create the concentration output file for CAPTEX release #2 as discussed in the HYSPLIT tutorial (<https://www.meteozone.com/tutorial/html/index.html>). All the post-processing options described for HYSPLIT output in the Tutorial can be applied to the ADAPT output.

### Creating the simulation name list: SETUP.CFG

The Atmospheric Dispersion and Parallel Transport (ADAPT) calculation uses the maximum number of available cores unless the environment has been adjusted: export `OMP_NUM_THREADS={x}`. The code is run according to the variables set in the name list file `SETUP.CFG`. The following statements describe the file contents in more detail. If the file is not found in the startup directory, a default version of the file will be created that sets the parameters to perform the calculation for CAPTEX release #2. These settings are shown below. All numeric inputs are integers unless indicated as real (**R**).

`back=.false.`

**! when true the integration direction is reversed**

The default setting of false results in a time-forward integration. When true the integration direction is reversed, and backward trajectories or upwind dispersion patterns will be calculated. The values of `datecal` and ***daterel*** should reflect the start time of the backward simulation or ending time of the equivalent forward trajectory. Note that all other time duration variables, such as a ***crun*** should be positive.

`traj=.false.`

**! when true computes a single trajectory**

A setting of false results in a concentration simulation and ***numpar*** particles are released starting at ***daterel*** for a duration of ***emdir*** minutes. When true, only one particle is released with advection, but no turbulent dispersion, and particle endpoint positions are written hourly to the file defined in variable ***trjfile***, which is written in the HYSPLIT trajectory format.

`trjfile='endpts_adapt.txt',`

**! defines the trajectory output file when traj=true**

When ***traj*** is true this file is written with the particle position at hourly intervals in the HYSPLIT trajectory file format.

`zgrid=.false.,`

**! when true configures the vertical grid with each input**

After the first meteorological data read, the pressure and temperature fields are integrated at each grid point to determine the height AGL of each data level. This field is then averaged over the domain and a single height versus index profile is used for all

grid points. When true, the height versus index relationship is recomputed after each meteorological data read.

**fixed=.false.,**

**! when true start with fixed random number index**

An array (2,000,000) random numbers are computed during the model initialization. As each new particle is released, the index within the random number array is assigned to that particle randomly, but when the fixed variable is true, then the index assignment always starts at one. Each new particle is sequentially assigned the next index, or all particles are assigned the next index when a random number is extracted.

**datem=.false.,**

**! when true output in DATEM format**

Particle mass is accumulated on an internal concentration grid that is identical or an even multiple of the meteorology grid. When true, the mass concentrations from the grid are transferred to a one-dimensional array defined by the locations and sampling times in the DATEM file in *datfile* and then output in the DATEM format to the file named in the variable *stnfile*. Both *datem* and *hybin* cannot both be true at the same time.

**hybin=.true.,**

**! when true output in HYSPLIT binary format**

Particle mass is accumulated on an internal concentration grid that is identical or an even multiple of the meteorology grid. When true, these values are interpolated to a latitude-longitude grid of comparable resolution in the HYSPLIT binary format to the file in *confile*. Both *datem* and *hybin* cannot both be true at the same time.

**confile='concout\_adapt.bin',**

**! concentration output file if datem=false**

When *datem* is false the internal concentration grid is remapped to a latitude-longitude grid prior to output in the HYSPLIT concentration binary format to the file named in the variable *confile*. All concentration output requires setting the start time *cstart* and the output averaging time *cinter*.

**datfile='measured\_datem.txt',**

**! datem input file if datem=true**

When *datem* is true this DATEM format file is read and the locations, sample start, and duration times are used as a template to output the model simulation results to an identical formatted file named in the variable *stnfile.txt*.

**stnfile='calculated\_datem.txt',**

**! datem output file if datem=true**

When *datem* is true this DATEM format file is written with the model calculated values at the locations, sample start, and duration times defined in file *datfile*. Although the sampling start and averaging times have been defined, *datem* output still requires setting the start time *cstart* and the output averaging time *cinter* because these values define the internal accumulation grid, which is used to interpolate to the sampling locations. These values also must be of finer temporal resolution than the sampling data and the start and end times should be aligned with the external data.

**html=.false.,**

**! when true concentration graphic output in SVG format**

When true the concentration output is converted to an SVG (scalable vector graphic) that is embedded in an html file, one file for each output time period, where the base

output file name is defined in the variable *svgfile*. Both *datem* and *html* cannot both be true at the same time, but both *hybin* and *html* can be true at the same time.

**svgfile='svgplot',**

**! base name for HTML file with SVG graphics**

When *html* is true, a simple graphic of color filled rectangles corresponding to the concentration grid is produced for each output time period. The output file name consists of the base name defined in *svgfile* plus a two-digit frame number, incremented by one for each new output period, terminated by the .html suffix. For example, the first output frame using the default values would be *svgplot01.html*. Note that the color fill code requires whole number concentration values that range from 20 to 20,000. The value of *cfactor* should be set to accommodate the display range. A circle is drawn to show a radius of 100 km and it represents the minimum size of the graphic.

**thin=.false.,**

**! when true no headers are written to the html output**

The SVG (scalable vector graphic) is embedded in an html file with multiple header records and metadata. When *thin* is true, these records are not written, and the html output file only contains contents between the <svg> </svg> tags. Thin files can be more easily converted to other formats using utilities such as Image Magick.

**mapfile='map\_standard.txt',**

**! base name for the background map file**

When *html* is true, the SVG graphic may contain a map background with political boundaries. This file is a text file, similar to the HYSPLIT map file, which contains the vectors as latitude and longitude points. If the file is not found, no background is added.

**cfactor=1.0,**

**! units' conversion for concentration output (R)**

The mass unit's conversion factor for concentrations prior to output.

**diag=.false.,**

**! when true output diagnostics**

When true, model diagnostic output is turned on such as particle mass profiles and meteorological data profiles. The meteorological data profile is always computed at the position of particle number one, each hour.

**metfile=' /Tutorial/captex/captex2\_wrf27uw.bin',,,,,**

**! meteo data file**

Defines the input meteorological data file in ARL format. Only WRF output is acceptable (actually any conformal projection on sigma surfaces), either instantaneous or time-averaged. The TKE field is required. Only trajectories can be computed if the TKE field is missing. Dispersion calculations will default to the minimum TKE. Up to five meteorological data files can be specified in the order that they should be opened. All files should be identical in grid dimensions and resolution.

**polid='PMCH',**

**! four character pollutant identification**

Label for pollutant released. It is only used for the output identification.

**totrel=201000.0,**

**! total mass released (R)**

The mass released can be any units and it represents the total amount released over the period defined by *emdur*.

**olat= 39.90,** **! source location latitude (R)**  
The latitude of the source location in degrees (positive is north).

**olon=-84.22,** **! source location longitude (R)**  
The longitude of the source location in degrees (positive is east).

**olvl=10.0,** **! starting height mAGL (R)**  
The starting height from which particles will be released (mAGL).

**datecal= 1983,9,25,17,,,** **! computational start time ({cc}yyymmddhhmmff)**  
The computational start time is represented by a two digit year, month, day, and hour. The last two fields are minutes and forecast hour, and usually are left at the default value of zero. The computational start time is start time of the internal elapsed time clock and represents the zero time for variables *crun* and *cstart*. Note that the century designation **{cc}** is optional and if omitted all internal calculations will default to the current century. Because the century is not designated in the ARL formatted meteorological data files, it may not be immediately obvious when this field is omitted that the date field of the internal calculation is incorrect.

**crun=64,** **! computational run length in hours**  
The field represents the total number of simulation hours from the computational start time for reading meteorology and output of results. The value is positive for both forward and backward calculations.

**daterel= 1983,9,25,17,,,** **! particle release start time ({cc}yyymmddhhmmff)**  
The particle release start time is the initial time the particle release starts. Note that if minutes are also defined (the forecast hour field has no meaning in this context), the minutes field should be consistent with the emission time step *dte*.

**emdur=180,** **! particle emission duration in minutes**  
Particles will be emitted over this duration starting at *daterel*.

**maxpar=50000,** **! maximum number of particles released over *emdur***  
The number of particles to be released over duration *emdur*. This value may be adjusted slightly internally so that the number of particles in the integration loop is evenly divisible by the number of threads allocated to the calculation. If the variable is set to zero, *maxpar* will be computed internally.

**cgratio=1,** **! number of concentration cells per WRF cell**  
During the particle integration step, the particle mass is accumulated on a grid cell with the same resolution as the meteorological data. To accommodate finer spatial resolution sampling, the horizontal resolution of the concentration grid can be defined as an even integer multiple ( $\geq 1$ ) of the meteorological data grid.

**conlvl=100,** **! average ground to level for concentration output mAGL**

Only one concentration output level is permitted, and it always starts at the ground and the top is defined by *conlvl*. Computationally the particles in all the meteorological grid cells from the ground to one that ends just below this level are accumulated.

**cstart=60,**

**! concentration output start time (min)**

The accumulation of particle mass to the internal concentration grid starts this many minutes after the computational start time. The value is positive for both forward and backward calculations.

**cinter=180,**

**! concentration output interval (min)**

At this interval after *cstart*, the concentrations are output to the file named in variable *confile* or *stnfile*. Although the sample start and durations are defined when *datem* is true, the concentration data from the internal grid are transferred to either *confile* or accumulated into the *stnfile* station list at the interval *cinter*, which must therefore be at least equal to the smallest sampling duration defined in *datfile*.

**dte=300,**

**! particle emission time step (sec)**

Between *datere1* and *emdur*, particles are emitted at this time step. This emission time step should be evenly divisible into *emdur*.

**dta=1200,**

**! maximum advection time step (sec)**

The advection integration starts with the default time step of *dta*, unless emissions are occurring in which case it is *dte*. For each particle, after the first-pass of the advection computation, the horizontal and vertical displacement grid distance is compared with their respective stability ratios, *tratioh* and *tratiov*. If the criteria are violated, the time step is adjusted downward until satisfied and the first-pass is recomputed. This time step also is the interval at which the multi-thread environment is enabled. A longer step reduces the overhead requirements and makes the code more efficient. However, this time step cannot exceed the time interval of the meteorological data. Also, particle mass is accumulated on the concentration grid during this period and output, grid or station, can only occur at the end of the *dta* integration loop. Therefore, this value should always be less than or equal to *cinter*.

**tratioh=0.50,**

**! ratio for horizontal time step stability (R)**

The time step is adjusted downward if *Udt* exceeds this value.

**tratiov=0.10,**

**! ratio for vertical time step stability (R)**

The time step is adjusted downward if *Wdt* exceeds this value.

**tker=0.18,**

**! ratio of vertical to horizontal turbulence (R)**

This value is used to partition the total turbulence into vertical and horizontal components with  $tke = W^2 / (U^2 + V^2)$ . If the input meteorology does not contain the TKE field, the turbulence partition will still occur but with the minimum TKE (0.05). In this case dispersion calculations are disabled but it is still possible to compute a trajectory.