

# AERMOD/AERMET/AERMAP 2024

## UBC ATSC 595D

### For Linux

- **Bolded entries are individual commands to be placed on the command line; they should be written and entered as a single line in the terminal**
- Ensure you have gfortran installed
  - **which gfortran**
- Main AERMOD site:  
<https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models>
- Sample run instructions:  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/AERMOD\\_Sample\\_Run\\_Instructions.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/AERMOD_Sample_Run_Instructions.pdf)

## Installing AERMOD

- Full AERMOD user's guide:  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/aermod\\_userguide.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/aermod_userguide.pdf)
  - AERMET:  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/met/aermet/aermet\\_userguide.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/met/aermet/aermet_userguide.pdf)
  - AERMAP:  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aermap/aermap\\_userguide\\_v18081.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aermap/aermap_userguide_v18081.pdf)
- Load up the necessary gfortran compiler (use modules on Optimum)
  - **module load GCC/8.3/0**

- Make a new directory and enter the directory
  - `mkdir AERMOD`
  - `cd AERMOD`
- Download the source code from the website:  
<https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models#aermod>
  - `wget --no-check-certificate`  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/aermod\\_source.zip](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/aermod_source.zip)
- Unzip the source
  - `unzip aermod_source.zip`
- Enter the newly unzipped directory and list out its contents
  - `cd aermod_source_code_23132`
  - `ls`
- You should see a whole bunch of source Fortran files (.f) and several batch (.bat) files
- .bat files are for Windows (DOS) users to instantly compile the needed code; but .bat files won't work for Linux users, so we'll have to make some edits
- Create write and execute permissions for gfortran-aermod.bat, then make a copy of it suffixed as .sh
  - `chmod 755 gfortran-aermod.bat`
  - `cp gfortran-aermod.bat gfortran-aermod.sh`
- Use vim to edit the gfortran-aermod.sh to look like the following, or copy from the following; for mass-editing %COMPILE\_FLAGS% to \$COMPILE\_FLAGS, use:
  - Linux:

```

■ sed -i
  s/"%COMPILE_FLAGS%"/'$COMPILE_FLAGS'/g
  gfortran-aermod.sh

```

- Mac [similar, but not the same, in case you want to do this again on a mac]:

```

■ sed -i ''
  s/"%COMPILE_FLAGS%"/'$COMPILE_FLAGS'/g
  gfortran-aermod.sh

```

- Edit gfortran-aermod.sh with vim:
  - **vi gfortran-aermod.sh**
- Use vim (command mode) to convert all upper-case letter to lower-case
  - `:%s/[A-Z]/L&/g`
- Fix the capitalization of the flags “-Wuninitialized” and “-O2”
- Convert all of the DOS commands to their bash equivalents (e.g. del to rm)
- Also change %link\_flags% to \$link\_flags
- Your file, after the required edits, should look like:

```
#!/bin/bash

which gfortran

export compile_flags="-fbounds-check -Wuninitialized -O2"
export link_flags="-O2"

gfortran -c $compile_flags modules.f
gfortran -c $compile_flags grsm.f
gfortran -c $compile_flags aermod.f
gfortran -c $compile_flags setup.f
gfortran -c $compile_flags coset.f
gfortran -c $compile_flags soset.f
gfortran -c $compile_flags reset.f
gfortran -c $compile_flags meset.f
gfortran -c $compile_flags ouset.f
gfortran -c $compile_flags inpsum.f
gfortran -c $compile_flags metext.f
gfortran -c $compile_flags iblval.f
gfortran -c $compile_flags siggrid.f
gfortran -c $compile_flags tempgrid.f
gfortran -c $compile_flags windgrid.f
gfortran -c $compile_flags calc1.f
gfortran -c $compile_flags calc2.f
gfortran -c $compile_flags prise.f
gfortran -c $compile_flags arise.f
gfortran -c $compile_flags prime.f
gfortran -c $compile_flags sigmas.f
gfortran -c $compile_flags pitarea.f
gfortran -c $compile_flags uninam.f
gfortran -c $compile_flags output.f
gfortran -c $compile_flags evset.f
gfortran -c $compile_flags evcalc.f
gfortran -c $compile_flags evoutput.f
gfortran -c $compile_flags rline.f
gfortran -c $compile_flags bline.f

gfortran -o aermod.exe $link_flags modules.o grsm.o aermod.o setup.o coset.o soset.o re
set.o meset.o ouset.o inpsum.o metext.o iblval.o siggrid.o tempgrid.o windgrid.o calc1.
o calc2.o prise.o arise.o prime.o sigmas.o pitarea.o uninam.o output.o evset.o evcalc.o
evoutput.o rline.o bline.o

rm -f *.o
rm -f *.mod
~
~
```

```
#!/bin/bash
```

```
which gfortran
```

```
export compile_flags="-fbounds-check -Wuninitialized -O2"
```

```
export link_flags="-O2"
```

```
gfortran -c $compile_flags modules.f
```

```
gfortran -c $compile_flags grsm.f
```

```
gfortran -c $compile_flags aermod.f
```

```
gfortran -c $compile_flags setup.f
```

```
gfortran -c $compile_flags coset.f
```

```
gfortran -c $compile_flags soset.f
```

```
gfortran -c $compile_flags reset.f
```

```
gfortran -c $compile_flags meset.f
```

```
gfortran -c $compile_flags ouset.f
```

```
gfortran -c $compile_flags inpsum.f
```

```
gfortran -c $compile_flags metext.f
```

```
gfortran -c $compile_flags iblval.f
```

```
gfortran -c $compile_flags siggrid.f
```

```
gfortran -c $compile_flags tempgrid.f
```

```
gfortran -c $compile_flags windgrid.f
```

```
gfortran -c $compile_flags calc1.f
```

```
gfortran -c $compile_flags calc2.f
```

```
gfortran -c $compile_flags prise.f
```

```
gfortran -c $compile_flags arise.f
```

```
gfortran -c $compile_flags prime.f
```

```
gfortran -c $compile_flags sigmas.f
```

```
gfortran -c $compile_flags pitarea.f
```

```
gfortran -c $compile_flags uninam.f
```

```
gfortran -c $compile_flags output.f
```

```
gfortran -c $compile_flags evset.f
```

```
gfortran -c $compile_flags evcalc.f
```

```
gfortran -c $compile_flags evoutput.f
```

```
gfortran -c $compile_flags rline.f
```

```
gfortran -c $compile_flags bline.f
```

```
gfortran -o aermod.exe $link_flags modules.o grsm.o aermod.o setup.o coset.o \
soaset.o reset.o meset.o ouset.o inpsum.o metext.o iblval.o siggrid.o tempgrid.o \
windgrid.o calc1.o calc2.o prise.o arise.o prime.o sigmas.o pitarea.o uninam.o \
output.o evset.o evcalc.o evoutput.o rline.o bline.o
```

```
rm -f *.o
```

```
rm -f *.mod
```

- Note that the final gfortran command (the big one) must be all on one line when the script is viewed unwrapped (or use \ to escape new lines, as copied above)
- In Vim, convert the file format from DOS to Unix; see <https://unix.stackexchange.com/questions/32001/what-is-m-and-how-do-i-get-rid-of-it>
  - **:set ff=unix**
- Save and quit with **:wq**
- If that's all worked out, run the script
  - **./gfortran-aermod.sh**
- Once it's finished running, you should see aermod.exe with **ls**
  - **./aermod.exe**
  - You should get the error: +Error Opening Runstream Input File!  
Aborting.

## Installing AERMET

- Full AERMET user's guide:  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/met/aermet/aermet\\_userguide.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/met/aermet/aermet_userguide.pdf)
- Go back to top level AERMOD directory
  - `cd ~/AERMOD`
- Download the source code from the website:  
<https://www.epa.gov/scram/meteorological-processors-and-accessory-programs#aermet>
  - `wget --no-check-certificate https://gaftp.epa.gov/Air/aqmg/SCRAM/models/met/aermet/aermet_source.zip`
- Unzip the source and enter unzipped directory
  - `unzip aermet_source.zip`
  - `cd aermet_source`
- Create write and execute permissions for `gfortran-aermet-64bit.bat`, then make a copy of it suffixed as `.sh`
  - `chmod 755 gfortran-aermet_allwarn.bat`
  - `cp gfortran-aermet_allwarn.bat gfortran-aermet.sh`
- Use vim to edit the `gfortran-aermet.sh` to look like the following, or copy from the following (hint: sed is your friend)

```

1 #!/bin/bash
2
3 export COMPILE_FLAGS="-fbounds-check -Wuninitialized -O2"
4 export LINK_FLAGS="-O2"
5
6 gfortran -c $COMPILE_FLAGS mod_file_units.f90
7 gfortran -c $COMPILE_FLAGS mod_main1.f90
8 gfortran -c $COMPILE_FLAGS mod_upperair.f90
9 gfortran -c $COMPILE_FLAGS mod_surface.f90
10 gfortran -c $COMPILE_FLAGS mod_onsite.f90
11 gfortran -c $COMPILE_FLAGS mod_pbl.f90
12 gfortran -c $COMPILE_FLAGS mod_read_input.f90
13 gfortran -c $COMPILE_FLAGS mod_reports.f90
14 gfortran -c $COMPILE_FLAGS mod_misc.f90
15 gfortran -c $COMPILE_FLAGS aernet.f90
16
17 gfortran -o aernet.exe $LINK_FLAGS mod_file_units.o mod_main1.o mod_upperair.o mod_surface.o mod_onsite.o mod_
pbl.o mod_read_input.o mod_reports.o mod_misc.o aernet.o
18
19 rm -fv *.o
20 rm -fv *.mod
~
~

```

```
#!/bin/bash
```

```
export COMPILE_FLAGS="-fbounds-check -Wuninitialized -O2"
```

```
export LINK_FLAGS="-O2"
```

```
gfortran -c $COMPILE_FLAGS mod_file_units.f90
```

```
gfortran -c $COMPILE_FLAGS mod_main1.f90
```

```
gfortran -c $COMPILE_FLAGS mod_upperair.f90
```

```
gfortran -c $COMPILE_FLAGS mod_surface.f90
```

```
gfortran -c $COMPILE_FLAGS mod_onsite.f90
```

```
gfortran -c $COMPILE_FLAGS mod_pbl.f90
```

```
gfortran -c $COMPILE_FLAGS mod_read_input.f90
```

```
gfortran -c $COMPILE_FLAGS mod_reports.f90
```

```
gfortran -c $COMPILE_FLAGS mod_misc.f90
```

```
gfortran -c $COMPILE_FLAGS aernet.f90
```

```
gfortran -o aernet.exe $LINK_FLAGS mod_file_units.o mod_main1.o \
mod_upperair.o mod_surface.o mod_onsite.o mod_pbl.o mod_read_input.o \
mod_reports.o mod_misc.o aernet.o
```

```
rm -fv *.o
```

```
rm -fv *.mod
```



- Note that the final gfortran command (the big one) must be all on one line when the script is viewed unwrapped (or use \ to escape new lines, as copied above)
- In Vim, convert the file format from DOS to Unix; see <https://unix.stackexchange.com/questions/32001/what-is-m-and-how-do-i-get-rid-of-it>
  - `:set ff=unix`
  - `:wq`
- If that's all worked out, run the script
  - `./gfortran-aermet.sh`
- Once it's finished running, you should see aermet.exe with `ls`
  - `./aermet.exe`
  - You should get the error: Input file aermet.inp not found

## Installing AERMAP

- Go to top-level AERMOD directory
  - `cd ~/AERMOD`
- Make a new directory and enter the directory
  - `mkdir aermap_source`
  - `cd aermap_source`
- Download the source code from the website:  
<https://www.epa.gov/scram/air-quality-dispersion-modeling-related-model-support-programs#aermap>
  - `wget --no-check-certificate`  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aermap/aermap\\_source.zip](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aermap/aermap_source.zip)
- Unzip
  - `unzip aermap_source.zip`
- Create write and execute permissions for gfortran-aermap-64bit.bat, then make a copy of it suffixed as .sh
  - `chmod 755 gfortran-aermap-64bit.bat`
  - `cp gfortran-aermap-64bit.bat gfortran-aermap.sh`
- ...You know the drill...don't forget `:set ff=unix`

```

1 #!/bin/bash
2
3 export COMPILE_FLAGS="-fbounds-check -Wuninitialized -O2"
4 export LINK_FLAGS="-O2"
5
6 gfortran -m64 -c $COMPILE_FLAGS mod_main1.f
7 gfortran -m64 -c $COMPILE_FLAGS mod_tifftags.f
8 gfortran -m64 -c $COMPILE_FLAGS aermap.f
9 gfortran -m64 -c $COMPILE_FLAGS sub_calchc.f
10 gfortran -m64 -c $COMPILE_FLAGS sub_chkadj.f
11 gfortran -m64 -c $COMPILE_FLAGS sub_chkext.f
12 gfortran -m64 -c $COMPILE_FLAGS sub_demchk.f
13 gfortran -m64 -c $COMPILE_FLAGS sub_nedchk.f
14 gfortran -m64 -c $COMPILE_FLAGS sub_cnrcnv.f
15 gfortran -m64 -c $COMPILE_FLAGS sub_demrec.f
16 gfortran -m64 -c $COMPILE_FLAGS sub_demsr.c.f
17 gfortran -m64 -c $COMPILE_FLAGS sub_domcnv.f
18 gfortran -m64 -c $COMPILE_FLAGS sub_initer_dem.f
19 gfortran -m64 -c $COMPILE_FLAGS sub_initer_ned.f
20 gfortran -m64 -c $COMPILE_FLAGS sub_nadcon.f
21 gfortran -m64 -c $COMPILE_FLAGS sub_reccnv.f
22 gfortran -m64 -c $COMPILE_FLAGS sub_recelv.f
23 gfortran -m64 -c $COMPILE_FLAGS sub_srccnv.f
24 gfortran -m64 -c $COMPILE_FLAGS sub_srcelv.f
25 gfortran -m64 -c $COMPILE_FLAGS sub_utmgeo.f
26 gfortran -m64 -c $COMPILE_FLAGS sub_read_tifftags.f
27
28 gfortran -m64 -o aermap.exe $LINK_FLAGS mod_main1.o mod_tifftags.o aermap.o sub_calchc.o sub_chkadj.o sub_chke
xt.o sub_demchk.o sub_nedchk.o sub_cnrcnv.o sub_demrec.o sub_demsr.c.o sub_domcnv.o sub_initer_dem.o sub_initer_ned
.o sub_nadcon.o sub_reccnv.o sub_recelv.o sub_srccnv.o sub_srcelv.o sub_utmgeo.o sub_read_tifftags.o
29
30 m -fv *.o *.mod

```

#!/bin/bash

```

export COMPILE_FLAGS="-fbounds-check -Wuninitialized -O2"
export LINK_FLAGS="-O2"

```

```

gfortran -m64 -c $COMPILE_FLAGS mod_main1.f
gfortran -m64 -c $COMPILE_FLAGS mod_tifftags.f
gfortran -m64 -c $COMPILE_FLAGS aermap.f
gfortran -m64 -c $COMPILE_FLAGS sub_calchc.f
gfortran -m64 -c $COMPILE_FLAGS sub_chkadj.f
gfortran -m64 -c $COMPILE_FLAGS sub_chkext.f
gfortran -m64 -c $COMPILE_FLAGS sub_demchk.f
gfortran -m64 -c $COMPILE_FLAGS sub_nedchk.f
gfortran -m64 -c $COMPILE_FLAGS sub_cnrcnv.f
gfortran -m64 -c $COMPILE_FLAGS sub_demrec.f
gfortran -m64 -c $COMPILE_FLAGS sub_demsr.c.f
gfortran -m64 -c $COMPILE_FLAGS sub_domcnv.f
gfortran -m64 -c $COMPILE_FLAGS sub_initer_dem.f

```

```

gfortran -m64 -c $COMPILE_FLAGS sub_initer_ned.f
gfortran -m64 -c $COMPILE_FLAGS sub_nadcon.f
gfortran -m64 -c $COMPILE_FLAGS sub_reccnv.f
gfortran -m64 -c $COMPILE_FLAGS sub_recelv.f
gfortran -m64 -c $COMPILE_FLAGS sub_srcnv.f
gfortran -m64 -c $COMPILE_FLAGS sub_srcelv.f
gfortran -m64 -c $COMPILE_FLAGS sub_utmgeo.f
gfortran -m64 -c $COMPILE_FLAGS sub_read_tifftags.f

```

```

gfortran -m64 -o aermap.exe $LINK_FLAGS mod_main1.o mod_tifftags.o \
aermap.o sub_calchc.o sub_chkadj.o sub_chkext.o sub_demchk.o sub_nedchk.o \
sub_cnrcnv.o sub_demrec.o sub_demsrc.o sub_domcnv.o sub_initer_dem.o \
sub_initer_ned.o sub_nadcon.o sub_reccnv.o sub_recelv.o sub_srcnv.o \
sub_srcelv.o sub_utmgeo.o sub_read_tifftags.o

```

```
rm -fv *.o *.mod
```

- ^big gfortran block above needs to be in one line or escaped (as copied) with  
\
- Run the **gfortran-aermap-64bit.sh** file
  - **./gfortran-aermap.sh**
- Run aermap.exe
  - **./aermap.exe**
  - You should get the error: +Error Opening Input File: AERMAP.INP Aborting.

## Running AERMOD TEST CASE - AERMAP

- Return to the top-level AERMOD directory
  - `cd ~/AERMOD`
- Grab the sample run and unzip
  - `wget --no-check-certificate`  
<https://gaftp.epa.gov/Air/aqmg/SCRAM/models/preferred/aermod/SampleRun.zip>
  - `unzip SampleRun.zip`
  - `cd SampleRun`
- The unzipped file contains three directories (AERMAP, AERMET, AERMOD) corresponding to the three portions of the software, alongside sample input files (e.g. aermod.inp; in AERMAP/AERMET/AERMOD the .inp files double as both namelists and input data files)
- Let's start by setting up our model domain (specifically, the source and receptor locations; in [UTM coordinates](#)) with AERMAP, and copy in our compiled aermap.exe executable
  - `cd AERMAP`
  - `cp ../../aermap_source/aermap.exe .`
  - ^Don't forget about the space and the period after aermap.exe
- Next, we need to edit aermap.inp in Vim to turn the input digital elevation model (DEM) names to lowercase
  - CO STARTING
  - TITLEONE MARTIN'S CREEK AERMAP RUN
  - DATATYPE DEM1
  - DATAFILE **newark-w.dem**
  - DATAFILE **newark-e.dem**
  - TERRHGTS PROVIDED
  - DOMAINXY 485000 4510000 18 497000 4520000 18
  - ANCHORXY 495510 513680 495510 4513680 18 1

- RUNORNOT RUN
- CO FINISHED
- You can see that aermap.inp is also used to specify source (stacks) and receptor locations
- Run aermap
  - **./aermap.exe**
- The output can be found in aermap.out; if you run **cat aermap.out**, you should see

```
*****  
*** AERMAP Finishes Successfully ***  
*****
```
- With **ls** you should also be able to see the files **aermap.rec** and **aermap.src**
  - From the sample guide: “Note: The current set of files has a known error that will leave the “aermap.src” file blank. This is not the same result as used in other applications, but this will not hinder any further steps. The user can safely ignore the empty “aermap.src” file.”

## Running AERMOD TEST CASE - AERMET

- Go into the AERMET sample run directory, and copy in our compiled aermet.exe executable
  - `cd ../AERMET`
  - `cp ../../aermet_source/aermet.exe .`
  
- AERMET is in charge of pre-processing the meteorological data used for running AERMOD; in particular, it creates a surface file (.sfc) and a profile file (.pfl) containing atmospheric variables
  - Typically ingests NWS data; can use MMIF software if WRF input is desired:  
<https://www.epa.gov/scram/air-quality-dispersion-modeling-related-model-support-programs#mmif>
  
- Stage 1: Extraction of surface and sounding data, and creation of quality-assurance (.oqa and .iqa) files
  - Edit **aermet1.inp** to capitalize the .DAT names (text-formatted input data):
    - ...
    - xdates      1992/5/1 to 1993/5/19
    - qaout        alb92-93.oqa
    - 
    - surface
    - \*\*            Surface data for Allentown-Beth-Easton, PA in  
CD144 format
    - data         **14737.DAT** CD144
    - extract     14737.iqa
    - qaout        14737.oqa
    - location    14737 40.65N 75.43W 0
    - xdates      1992/5/1 TO 1993/5/19
    - 
    - onsite
    - data         **MCOSPFL.DAT**

- 
- ...
- Run `aermet.exe` with the Stage 1 input file
  - `./aermet.exe ./aermet1.inp`
  - Should see AERMET FINISHED SUCCESSFULLY
- `ls`
  - Should now see several `.iqa` and `.oqa` files
- Stage 2: Creation of `.sfc` and `.pfl` files for input into AERMOD; estimates PBL parameters
  - Requires `alb92-93.oqa` (upper-air), `14737.oqa` (surface), `mcospfl.oqa` (onsite, i.e. non-NWS met tower) QA files to have been created in Stage 1
  - Run `aermet.exe` with the Stage 2 input file
    - `./aermet.exe ./aermet2.inp`
    - Should see AERMET FINISHED SUCCESSFULLY
  - `ls`
    - Should now see `aermet.pfl` and `aermet.sfc`
- There used to be an intermediate Stage to merge the `.oqa` files, but looks like now that's all handled in Stage 2 (see AERMET user's guide for this change)



## Running AERMOD TEST CASE - AERMOD

- Go into the AERMOD sample run directory, and copy in our compiled aermod.exe executable
  - `cd ../AERMOD`
  - `cp ../../aermod_source_code_23132/aermod.exe .`
  
- AERMOD runs the actual model itself, and requires input from the files we produced in AERMAP (source+receptor info) and AERMET (sfc+pfl meteorological info); it also needs an emissions file aermod.emi (already provided here)
  
- Edit aermod.inp to get the paths correct for our AERMAP and AERMET files [**HOMEWORK: change TITLEONE to include your name**]
  - ...
  - SRCGROUP ALL
  - SO FINISHED
  - 
  - 
  - RE STARTING
  - 
  - INCLUDED ../AERMAP/aermap.rec
  - 
  - RE FINISHED
  - 
  - ME STARTING
  - SURFFILE ../AERMET/aermet.sfc free
  - PROFFILE ../AERMET/aermet.pfl free
  - SURFDATA 14737 1992 Allentown
  - SITEDATA 00001 1992 Martin\_Crk
  - UAIRDATA 14735 1992 Abany
  - PROFBASE 73.2
  - ...

- For plotting, can use AERPLOT (Windows only):  
[https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aerplot/AERPLOT\\_Sample\\_Run\\_Instructions.pdf](https://gaftp.epa.gov/Air/aqmg/SCRAM/models/related/aerplot/AERPLOT_Sample_Run_Instructions.pdf)
- Also add aerplot.plt to aermod.inp to create an easy-to-read data file to plot data at the receptors
  - \*\*\*
  - OU STARTING
  - 
  - RECTABLE allave first-second
  - MAXTABLE allave 400
  - SUMMFILE aermod.sum
  - **PLOTFILE period all ./aerplot.plt**
  - OU FINISHED
  - \*\*\*
- Run aermod.exe
  - **./aermod.exe**
  - Should see:
    - +Now Processing Data For Day No. 137 of 1993
    - +Now Processing Data For Day No. 138 of 1993
    - +Now Processing Data For Day No. 139 of 1993
    - +Now Processing Output Options
  - **cat aermod.out**
    - Should see:
 

```
*****
*** AERMOD Finishes Successfully ***
*****
```
- **HOMEWORK: Submit your copy of aermod.inp with TITLEONE containing your name (in the CO control block), and a copy of aermod.plt**

