ATSC 595 Air Pollution Modelling

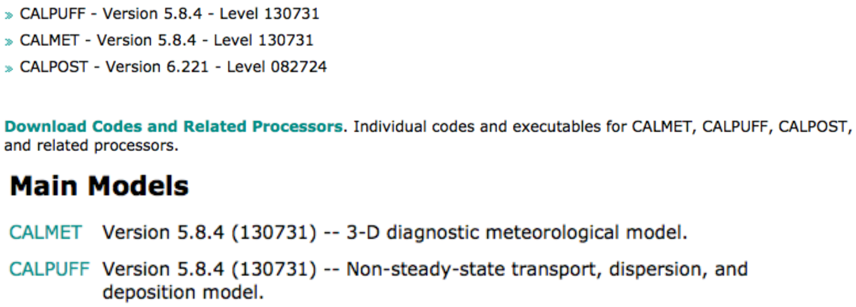
CALPUFF & CALMET Setup Guide

Mar. 2016
M. Fung

This guide will cover compilation and testing of CALPUFF air pollution model and CALMET meteorological pre-processor. (Version 5.8.4) on UNIX-like machines (OSX and Linux). Approximately 250Mb of disk space is required.

---

1. Download CALPUFF and CALMET source code at
   http://www.src.com/calpuff/download/epa_codes.htm
   Under Main Models, click Download for both CALPUFF and CALMET
   For this guide, we will be using the latest version of Calpuff that is approved by USEPA (V. 5.8.4)



» CALPUFF - Version 5.8.4 - Level 130731
» CALMET - Version 5.8.4 - Level 130731
» CALPOST - Version 6.221 - Level 082724

Download Codes and Related Processors. Individual codes and executables for CALMET, CALPUFF, CALPOST, and related processors.

Main Models

CALMET	Version 5.8.4 (130731) -- 3-D diagnostic meteorological model.
CALPUFF	Version 5.8.4 (130731) -- Non-steady-state transport, dispersion, and deposition model.

2. Once the download is complete, unzip both files.
   Open terminal and change directory to '*CALPUFF/CODE*'

```
"cd /path/to/CALPUFF/CODE"
```

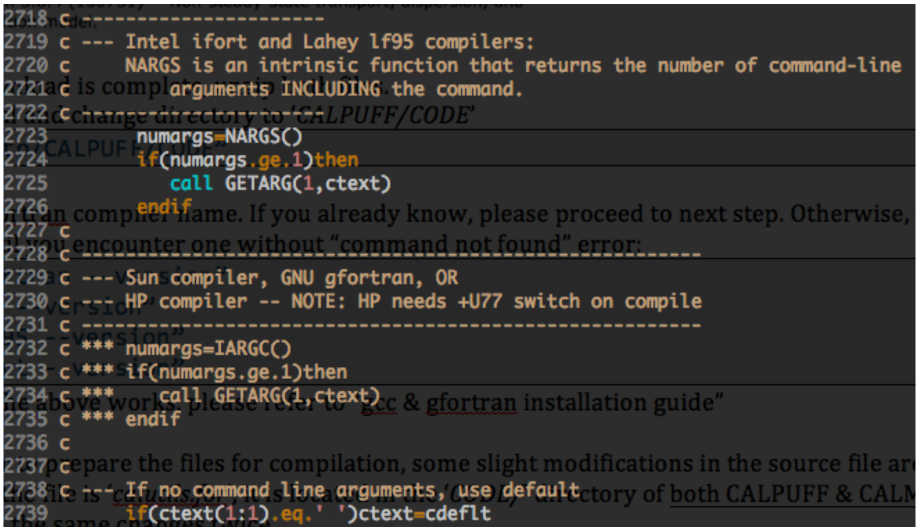
3. Check your fortran compiler name. If you already know, please proceed to next step. Otherwise, type the following until you encounter one without "command not found" error:

```
      "gfortran --version"
      "g95 --version"
      "pgf95 --version"
      "ifort --version"
```

   If neither of the above works, please refer to "gcc & gfortran installation guide"

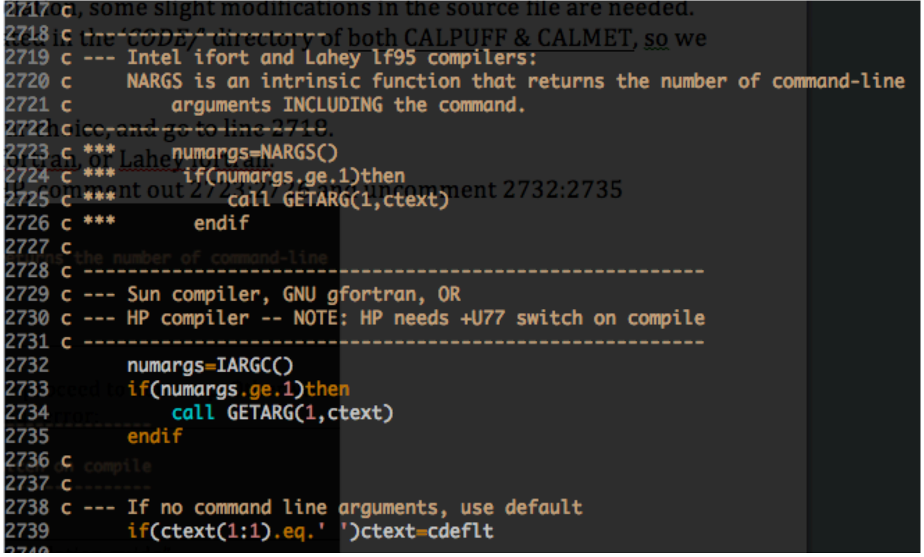
4. Next, in order to prepare the files for compilation, some slight modifications in the source file are needed. The name of the file is '*calutils.for*', it is located in the '*CODE/*' directory of both CALPUFF & CALMET, so we have to make the same changes twice.

   Open '*calutils.for*' with the text editor of your choice, and go to line 2718.
   No change is needed if you are using Intel fortran, or Lahey fortran.



```
2718 c---------------------
2719 c --- Intel ifort and Lahey lf95 compilers:
2720 c     NARGS is an intrinsic function that returns the number of command-line
2721 c         arguments INCLUDING the command.
2722 c
2723        numargs=NARGS()
2724        if(numargs.ge.1)then
2725           call GETARG(1,ctext)
2726        endif
2727 c
2728 c --------------------------------------------------------
2729 c --- Sun compiler, GNU gfortran, OR
2730 c --- HP compiler -- NOTE: HP needs +U77 switch on compile
2731 c ---------------------------------------------------------
2732 c *** numargs=IARGC()
2733 c *** if(numargs.ge.1)then
2734 c ***    call GETARG(1,ctext)
2735 c *** endif
2736 c
2737 c
2738 c --- If no command line arguments, use default
2739       if(ctext(1:1).eq.' ')ctext=cdeflt
```

For those who are using gfortran, or Sun/HP,
comment out 2723:2726 and uncomment 2732:2735 (CALPUFF)
2712:2716 2721:2724 (CALMET)



```
2717 c
2718 c
2719 c --- Intel ifort and Lahey lf95 compilers:
2720 c     NARGS is an intrinsic function that returns the number of command-line
2721 c         arguments INCLUDING the command.
2722 c
2723 c ***     numargs=NARGS()
2724 c ***      if(numargs.ge.1)then
2725 c ***         call GETARG(1,ctext)
2726 c ***       endif
2727 c
2728 c ---------------------------------------------------------
2729 c --- Sun compiler, GNU gfortran, OR
2730 c --- HP compiler -- NOTE: HP needs +U77 switch on compile
2731 c ---------------------------------------------------------
2732       numargs=IARGC()
2733       if(numargs.ge.1)then
2734          call GETARG(1,ctext)
2735       endif
2736 c
2737 c
2738 c --- If no command line arguments, use default
2739       if(ctext(1:1).eq.' ')ctext=cdeflt
```

Now the source code is ready for compilation.

5. Compiling CALPUFF
   In terminal, change directory path to ‘*CALPUFF/*CODE’

```
“cd /location/of/CALPUFF/CODE’
```

Now, to compile the source code, type:

```
“gfortran -Wall -o calpuff calpuff.for 2>compile-calpuff-gfortran.log”
```

You should see an executable file named “*calpuff*” in you *‘CODE/’* folder.
If not, please refer to ‘*compile-calpuff-gfortran.log’* for any error logs.

6. Testing CALPUFF

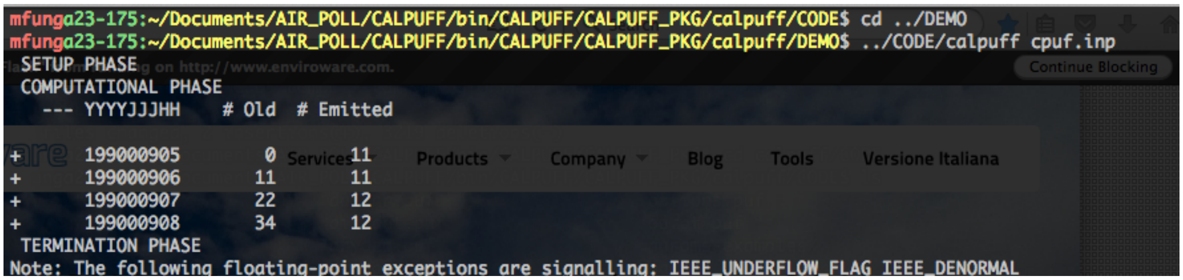
Calpuff runs by running the newly compiled ‘*calpuff*’ executable on an ‘*.inp*’ folder.
The resulting data will be written on a ‘*.dat*’ file and the log will be written on a ‘*.lst’* file. Both of the output file names are specified in the ‘*.inp*’ file.

In terminal, change directory to ‘*CALPUFF/DEMO*’

```
“cd /localtion/of/CALPUFF/DEMO”
```

Run cpuf.inp by typing (while your terminal is in ‘*DEMO/*’)

```
“/location/of/CALPUFF/CODE/calpuff cpuf.inp”
```



```
mfunga23-175:~/Documents/AIR_POLL/CALPUFF/bin/CALPUFF/CALPUFF_PKG/calpuff/CODE$ cd ../DEMO
mfunga23-175:~/Documents/AIR_POLL/CALPUFF/bin/CALPUFF/CALPUFF_PKG/calpuff/DEMO$ ../CODE/calpuff cpuf.inp
 SETUP PHASE
 COMPUTATIONAL PHASE
   --- YYYYJJJHH     # Old   # Emitted

+      199000905       0         11
+      199000906      11         11
+      199000907      22         12
+      199000908      34         12
 TERMINATION PHASE
Note: The following floating-point exceptions are signalling: IEEE_UNDERFLOW_FLAG IEEE_DENORMAL
```

If Calpuff terminates without errors, your Calpuff program is ready to be used.
If can error occurred, please refer to ‘cpuf.lst’ for error messages.
As well, the ‘common mistakes’ section at the end of this guide may be helpful.

7. Compiling CALMET
Compiling CALMET is a lot like compiling CALPUFF. The instructions are pretty much like step 5&6, with slight variations,

In terminal, change directory to ‘*CALMET/CODE*’

```
“cd /location/of/CALMET/CODE”
```

Proceed to type:

```
“gfortran -Wall -o calmet calmet.for 2>compile-calmet-gfortran.log”
```

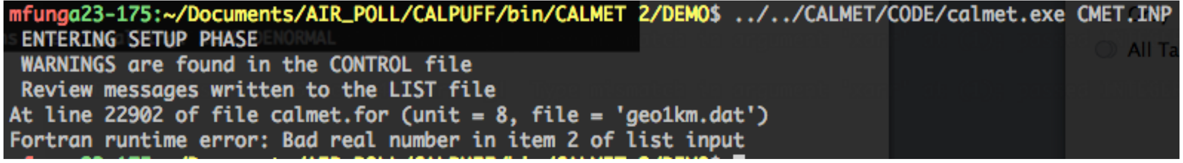
Look for the ‘*calmet*’ executable now in ‘*CALMET/CODE/calmet*’
If ‘*calmet*’ does not exist, please refer to ‘*compile-calmet-gfortran.log*’ for error messages

8. Testing CALMET
Change directory to ‘*CALPUFF_PKG/calmet/DEMO*’

Run ‘*calmet*’ on ‘*cmet.inp*’ by typing:

```
“/path/to/CALPUFF_PKG/calmet/CODE/calmet cmet.inp”
```

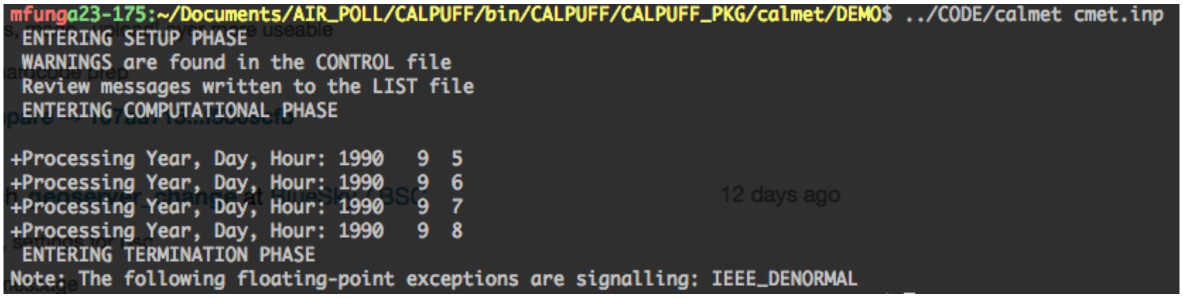
When I first tried to run the test, I encountered the following error:



```
mfunga23-175:~/Documents/AIR_POLL/CALPUFF/bin/CALMET 2/DEMO$ ../../CALMET/CODE/calmet.exe CMET.INP
 ENTERING SETUP PHASE
 WARNINGS are found in the CONTROL file
 Review messages written to the LIST file
At line 22902 of file calmet.for (unit = 8, file = 'geo1km.dat')
Fortran runtime error: Bad real number in item 2 of list input
```

If you do, please go to step 9, otherwise, if you see the message below, you are done compiling. (log file is in CMET.LST, data file is in CMET.DAT)

If you encou3nter another problem, try looking at hints at the appendix.



```
mfunga23-175:~/Documents/AIR_POLL/CALPUFF/bin/CALPUFF/CALPUFF_PKG/calmet/DEMO$ ../CODE/calmet cmet.inp
 ENTERING SETUP PHASE
 WARNINGS are found in the CONTROL file
 Review messages written to the LIST file
 ENTERING COMPUTATIONAL PHASE

+Processing Year, Day, Hour: 1990   9  5
+Processing Year, Day, Hour: 1990   9  6
+Processing Year, Day, Hour: 1990   9  7
+Processing Year, Day, Hour: 1990   9  8
 ENTERING TERMINATION PHASE
Note: The following floating-point exceptions are signalling: IEEE_DENORMAL
```

For any issue, please look into '*cmet.lst*'

9. The above-mentioned issue is caused by 'geo1km.dat' (or whatever file your terminal shows) being in windows 'dos' format. In this case, the calmet compilation is not broken, rather 'geo1km.dat' needs to be converted to UNIX format. (Only geo1km.dat is affected for me, none of the other .dat files need converting, but the process is the same for other files.)

   On Mac and Linux, we will be using a program called 'dos2unix'. If the program is not installed, type:

   ```
   "port install dos2unix" or "sudo port install dos2unix" [FOR OSX]
   "sudo apt-get install dos2unix" [FOR LINUX]
   ```

   for the respective operating systems.

   With your terminal opened to '*CALMET/DEMO'*, run dos2unix on geo1km.dat by typing:

   ```
   "dos2unix geo1km.dat
   ```

   Now try to run calmet on cmet.inp again, it should now produce the desired output.