

# The new pbl\_met: an open-source library for building custom meteorological processors for small-scale dispersion modeling

Patrizia Favaron - Servizi Territorio srl - patrizia@serviziterritorio.it

## A bit of history...

The first pbl\_met (then "PBL\_MET") was written by Daniele Fraternali and Roberto Sozzi around 1990, and presented in 1993 at the 2nd HARMO workshop held at Manno (Switzerland). Its original purpose was to help practitioners (who were meteorologists and physicists, and not necessarily professional programmers) building *custom meteorological processors*: the computer codes which transformed the available meteorological data into inputs to pollutant dispersion models and other technical appliances. A nitty-gritty task, which the programming languages and tools of 30 years ago made error prone and deadly tedious.



## World changes, anyway

The former PBL\_MET was designed imagining data coming from a "conventional meteorological station" (left: station at Grigna Vetta, about 2003), that is, equipped with electro-mechanical sensors. The met processors, on these years, had a simple purpose: to *estimate* the quantities which technology did not allow to measure directly. Namely, turbulence indices.

30 years have passed since then.

And technology did evolve. Conventional stations still exist, but are no longer alone: "advanced" stations have joined. "Advanced" stations are characterized by sensors, like the three-dimensional ultrasonic anemometer (on right), which are able to arrive very close to the direct measurement of the quantities formerly estimated.

Advanced stations, in addition, may not be "pointwise": SODARs, LIDARs, compact disdrometric radars are just examples of sensors able to yield vertical profiles of common meteorological variables.



## Which uses may be of meteorological processors, today?

Estimation of non-measured quantities is still *à la page* with the many existing meteorological stations.

With *advanced* stations, the interest shifts to a more diverse set of possible uses. Indeed, producing inputs for dispersion models, using measured data instead of estimates. But also, validating and cross-validating measurements (conventional and advanced), providing quality input data to meteorological and climatological models, unifying data sets coming from diverse sensors and technologies. That is to say, integrating complex data, possibly in real-time.

## The new pbl\_met

The pbl\_met, trying to answer these new requirements, did evolve.

Two new sections deal with ultrasonic anemometry and vertical profile reconstruction.

An effort was made to allow easy computing of common statistical and time-series exploratory data analyses.

And, maybe of greatest importance, the new pbl\_met has been conceived since its inception as *open-source*, with use of the permissive MIT license. Use is then free for both non-commercial and commercial applications.

As of the programming language, the authors decided to stay with Fortran. Not the old FORTRAN 77, however: Fortran 2008 has been used, with a (quite conservative) use of the new object-oriented extensions introduced with Fortran 2003.

Compared to the former PBL\_MET (whose text is included in the new), the new pbl\_met is less "encyclopaedic" and more "practically inclined".

## Current sections of pbl\_met

The new pbl\_met is packaged as a set of Fortran modules, with a header module used to provide a unique interface.

The current modules are:

- **pbl\_base.f90** : Common definitions, and support for INI files and spline functions.
- **pbl\_time.f90** : Treatment of coarse and fine resolution time stamps, and time arithmetics.
- **pbl\_stat.f90** : Basic statistics, auto- and cross-covariance and -correlation, time series.
- **pbl\_thermo.f90** : Thermodynamics, psychrometry, radiometry.
- **pbl\_evtrn.f90** : Evapotranspiration.
- **pbl\_wind.f90** : Basic wind statistics and simple eddy covariance.
- **pbl\_simil.f90** : Monin-Obukhov ("surface layer") similarity, vertical profiles reconstruction.
- **pbl\_depth.f90** : Mixing height estimation.

In addition to these "operational" modules, containing real code, a module collecting them all is also present: **pbl\_met.f90**.

## Accessing and using the pbl\_met

The new (and old) pbl\_met can be downloaded from the public repository

[https://github.com/serv-terr/pbl\\_met](https://github.com/serv-terr/pbl_met)

The library components can be found in directory "/core": to use them, one of the possible ways is to compile the code, then collect all the objects into a static library. This may then be linked to user code.

Alternatively, all the pbl\_met sources may be included as text into user projects' directories, and compiled all together with user code.

The executable generation process is eased by some makefiles (in directory "/core").

The compilers used by this poster author were GNU Fortran ("gfortran") and PGI Fortran Community Edition (the latter being currently under revision: will be issued as an NVIDIA product).

Please don't forget to copy the **.mod** files along with the library or object code: they contain essential information used at compile time.

## What the pbl\_met has been used for?

As mentioned, the MIT open-source license allows commercial use. So, Servizi Territorio srl used it as a basis for developing the **ST-Me** meteorological processor. This is a commercial code, aimed at integrating data from conventional and advanced sensors (to date ultrasonic anemometer and SODAR, SODAR/RASS), and producing meteorological inputs for various dispersion models.

Other uses occurred, ranging from the construction of meteorological processors to the development of meteorological data analysis codes.

## What is missing?

The pbl\_met is an evolving concept and system. But nevertheless it would greatly benefit of a complete manual: the one you may find in github repository is just an embryo.

Another missing point is a set of old PBL\_MET routines which have not yet be implemented in the new pbl\_met.

Besides of this, some important modules are still missing:

- Data gap filling, of statistical type (e.g. based on typical days). That's would be very useful e.g. to feed the highly gap-sensitive Lagrangian models.
- Peak detection algorithms, of great usefulness in the context of odour and toxic chemicals data analysis, and in support of validating meteorological data.
- Stationarity analysis algorithms.
- ...: which, exactly?

But maybe the most dangerous lack is of diversity, and support: to date, basically, the new pbl\_met is a one-woman endeavour. As any one-person object, it lacks variety of views and competences. Maybe **you** are willing to cooperate? (The current pbl\_met author and maintainer would be really glad to include you in the band. Besides, she would claim loud working at the pbl\_met is an excellent and very useful way for "cubs of meteorologists, physicists and mathematicians" to learn programming in modern Fortran: a good professional investment. Available to follow these and the like, too. 😊)

References:

Allen R.G., I.A. Walter, R.L. Elliott, T.A. Howell, D. Itenfisu, M.V. Jensen, R.L. Snyder (eds.), *The ASCE Standardized Reference Evapotranspiration Equation*, ASCE, 2005

Aubinet M., T. Vesala, D. Papale (eds.), *Eddy Covariance - A Practical Guide to Measurement and Data Analysis*, Springer, 2012

Box G.E.P., G.M. Jenkins, *Time Series Analysis - Forecasting and control*, Holden-Day, 1976

Chatfield C., *The Analysis of Time Series - An Introduction*, Chapman & Hall, 2004

Fisher N.I., *Statistical analysis of circular data*, Cambridge University Press, 2000

Foken T., *Micrometeorology*, Springer, 2008

Pal Arya S., *Introduction to Micrometeorology*, 2nd edition, Academic Press, 2001

Pavliotis G.A., *Stochastic Processes and Applications - Diffusion Processes, the Fokker-Planck and Langevin Equations*, Springer, 2014

Sozzi R., D. Fraternali, PBL\_MET library: A software library for Advanced Meteorological and Air Quality Data Processing (Fortran 77 and C versions), *2nd Workshop on European Harmonisation of Atmospheric Dispersion Modelling Systems*, Manno, 2003

Sozzi R., T. Georgiadis, M. Valentini, *Introduzione alla turbolenza atmosferica: Stime, concetti, misure*, Pitagora, 2002

Stull R. *An Introduction to Boundary Layer Meteorology*, Kluwer, 1988

## An example slice of code

```
! Simple example program showing how to compute the
! dew-point, given "regular"
! (dry bulb) temperature and relative humidity - just the stuff you may hope
! to get from a thermo-hygrometer.
!
! By: Patrizia Favaron

program DewPoint
  use pbl_met
  implicit none

  ! Locals
  integer          :: iRetCode
  character(len=16) :: sBuffer
  real             :: temp      ! Temperature, in °C
  real             :: relh      ! Relative humidity, in %

  ! Get temperature and relative humidity
  ! from the command line (or print a helpful
  ! message if something looks wrong)
  if(command_argument_count() /= 2) then
    call helpfulMessage()
    stop
  end if
  call get_command_argument(1, sBuffer)
  read(sBuffer, *, iostat=iRetCode) temp
  if(iRetCode /= 0) then
    ...
  end if
  call get_command_argument(2, sBuffer)
  read(sBuffer, *, iostat=iRetCode) relh
  if(iRetCode /= 0) then
    ...
  end if

  ! Perform the calculation using pbl_met, and print it on the fly
  print *, "Dew point temperature = ", &
    DewPointTemperature(temp + 273.15, relh) - 273.15, " °C"

contains

  subroutine helpfulMessage()...

end program DewPoint
```

This very short program can be found together with some others in the "/example" directory. As its name and source imply, the program is designed to compute the dew point, given temperature and relative humidity.

Other examples include the experimental, short-range Lagrangian particle dispersion model ALAMO, first created by prof. Roberto Sozzi, and then refactored to take advantage of the pbl\_met.

You may find me at my e-mail, [patrizia@serviziterritorio.it](mailto:patrizia@serviziterritorio.it) : I'll be glad to answer you. 😊