FPLUME

USER'S MANUAL

Arnau Folch[1], Antonio Costa[2], Giovanni Macedonio[3]

[1] Barcelona Supercomputing Center (BSC-CNS)
Edifici NEXUS I, 2a planta - c/ Gran Capità 2-4
08034 Barcelona, Spain

[2]Istituto Nazionale di Geofisica e Vulcanologia (INGV)
Via Donato Creti, 12 - 40128 Bologna, Italy

[3]Istituto Nazionale di Geofisica e Vulcanologia (INGV)
Via Diocleziano, 328 - 80124 Napoli, Italy

Version 1.1 - February 2016

Contents

1 Introduction

FPLUME is a steady-state 1D cross-section averaged eruptive plume model which accounts for plume bent over, entrainment of ambient moisture, effects of water phase changes on the energy budget, particle fallout, particle re-entrainment by turbulent eddies, different empirical parameterizations for air entrainment, and particle aggregation in presence of liquid water or ice that depends on plume dynamics, particle properties, and amount of liquid water and ice existing in the plume. The modeling of aggregation in the plume allows for estimating an effective Total Grain Size Distribution (TGSD) depleted in fines with respect to that erupted at the vent. Details on the model are described in Folch et al. (2016).

2 Input files

FPLUME needs the following **input files**:

1. An input file named *problem*.inp specifying the control parameters and options;

2. A granulometry file named *problem*.tgsd describing Total Grain Size Distribution (TGSD), density, and shape of particles. The TGSD is typically obtained from field data analysis of past eruptions. However it can also be generated by the utility `fplume-tgsd`, distributed with this package, assuming either a Gaussian or bi-Gaussian distribution (in Φ-units);

3. A meteorological file named *problem*.met describing atmospheric profiles of density, pressure, temperature, specific humidity, and velocities. This information can be passed through a file or calculated assuming properties of the Standard Atmosphere (Carmichael, 2013).

2.1 Input file `problem.inp`

Here we describe the structure of the *problem*.inp whereas an example is given in Appendix A.
The input file *problem*.inp contains the following blocks:

    TIME_UTC

    SOURCE

    AGGREGATION

    METEO

    CONSTANTS

    MODELLING

2.1.1 Block `TIME_UTC`

This block defines variables related to time, such as the period covered by the meteorology file, and the beginning and end of the eruption. The block has the following format:

- `YEAR`: Database starting year (YYYY).

- `MONTH`: Database starting month (MM).

- `DAY`: Database starting day (DD).

- `ERUPTION_START_(HOURS_AFTER_00)`: Time slice of the eruption start (in h after 00UTC of the starting day). These are nt values (nt XXXXX 1) indicating the starting times of the different eruptive phases. Transient behavior of the eruption column can be described by adding a sufficient number of intervals. Eruptive conditions (plume height, MFR, etc.) are assumed constant during each phase. The first value must be equal or larger than the value of the record `BEGIN_METEO_DATA_(HOURS_AFTER_00)`.

- `ERUPTION_END_(HOURS_AFTER_00)`: Time slice of the eruption end (in h after 00UTC of the starting day). This is the time slice at which the source term is switched off (i.e. the time at which the last eruptive phase ends).

2.1.2 Block `SOURCE`

This block defines variables related the eruptive vent properties. The block has the following format:

- `LON_VENT`: Vent longitude. Only used in the LON-LAT option.

- `LAT_VENT`: Vent latitude. Only used in the LON-LAT option.

- `VENT_HEIGHT`: Height of the vent a.s.l. (in m).

- `SOLVE_PLUME_FOR`: The two available options are MFR or HEIGHT. In the first case SETSRC solves for the mass flow rate given the column height, whereas in the second does the opposite.

- `MFR_SEARCH_RANGE`: Two values n and m such that 10n and 10m specify the range of MFR values admitted in the iterative solving procedure (i.e. it is assumed that 10n MFR 10m). Only used if `SOLVE_PLUME_FOR=MFR`.

- `MASS_FLOW_RATE_(KGS)`: Values of the mass flow rate (in kg/s) for the nt eruptive phases. Only used if `SOLVE_PLUME_FOR=HEIGHT`.

- `HEIGHT_ABOVE_VENT_(M)`: Heights of the plume (in m above the vent) for the nt eruptive phases. Note that the plume heights must be lower than the top of the computational domain, specified in the variable `ZLAYER_(M)` of the GRID block. Only used if `SOLVE_PLUME_FOR=MFR`.

- `EXIT_VELOCIY_(MS)`: Values of the magma exit velocity (in m/s) at the vent for the nt eruptive phases.

- `EXIT_TEMPERATURE_(K)`: Values of the magma exit temperature (in K) at the vent for the nt eruptive phases.

- `EXIT_WATER_FRACTION_(IN%)`: Values of the magma volatile fraction (in weight percent) at the vent for the nt eruptive phases.

- `TERMINAL_VELOCITY_MODEL`: Type of terminal settling velocity model. The available options are `ARASTOOPOUR`, `GANSER`, `WILSON`, and `DELLINO`.

2.1.3 Block `AGGREGATION`

This block defines variables related aggregation parameterization. The block has the following format:

- `AGGREGATION_MODEL`: Aggregation parameterization. Available options are `NONE`, `PERCENTAGE` (based on Sulpizio et al. (2012)), `CORNELL` (based on Cornell et al. (1983)) or `COSTA` (based on Costa et al. (2010));

- `FI_AGGREGATES`: class of aggregates (i.e. main mode).

- `DENSITY_AGGREGATES`: Density of aggregates;

- `VSET_FACTOR`: Multiplicative correction factor for settling velocity of aggregates;

- `PERCENTAGE_(%)`: Percentage of ash aggregating for classes < `FI_AGGREGATES`. Only read if `AGGREGATION_MODEL = PERCENTAGE`;

- `FRACTAL_EXPONENT`: Fractal exponent (see Costa et al., 2010, for details). Only read if `AGGREGATION MODEL = COSTA`.

2.1.4 Block `METEO`

This block defines the meteorological variables. The block has the following format:

- `METEO_DATA`: Profiles of the meteorological variables (density, pressure, temperature, specific humidity, horizontal velocity components). Available options are:

  - `STANDARD_ATMOSPHERE`: profiles are internally calculated assuming Standard Atmosphere properties;

  - `PROFILE`: profiles are read from the file *problem*`.met`. An example of this file is given in Appendix B.

2.1.5 Block CONSTANTS

This block defines the values of some physical constants. When they are not specified, standard default values are assumed. The block has the following format:

- Ca: Specific heat of air at constant pressure. Default value 1000 J/(kg K);

- Cp: Specific heat of pyroclasts at constant pressure. Default value 1600 J/(kg K);

- Cw: Specific heat of water (vapour,liquid,ice) only if latent_heat=no. Default value 2000 J/(kg K).

2.1.6 Block MODELLING

This block defines variables defining the options for accounting for or neglecting some physical couplings and parameterization. The block has the following format:

- WIND_COUPLING: Options are YES or NO. If NO wind velocity components are assumed null;

- AIR_MOISTURE: Options are YES or NO. If NO, air specific humidity is assumed zero (dry entrained air only);

- REENTRAINMENT: Options are YES or NO. If NO particle reentrainment is neglected;

- LATENT_HEAT: Options are YES or NO. If NO water latent heat contribution is neglected;

- xi: value defining the probability that an individual particle will fall out of the plume, e.g. constant ξ in eq. (3g) of Folch et al. (2016). Default value 0.23 (Bursik, 2001);

- zmin_wind: Value of height below which wind entrainment in the low jet region is ignored. Default value is 0.0 m;

- c_umbrella: Value defining the thickness of the umbrella relative to Hb (¿1). Default value 1.32 (Sparks, 1986);

- a_s: options for the calculation of radial entrainment coefficient. Possible choices are:

  - CONSTANT a_s1 a_s2. In this case entrainment coefficient is assumed constant using a_s1 a_s2 for the value of a_s in the jet and plume regions respectively. In case a_s1 a_s2 are not specified default values of 0.075 and 0.12 are assumed;

  - KAMINSKI-R: based on the formulation of Kaminski et al. (2005) as proposed by Folch et al. (2016) including data of Rouse et al. (1952);

  - KAMINSKI-C: based on the formulation of Kaminski et al. (2005) as calculated by Folch et al. (2016) excluding data of Rouse et al. (1952);

- a_v: options for the calculation of radial entrainment coefficient. Possible choices are:

  - CONSTANT a_v1. In this case entrainment coefficient is assumed constant using a_v1 for the value of a_v. In case a_v1 is not specified default values of 0.3 is assumed;

  - TATE: based on the formulation of Tate (2002) as proposed by Folch et al. (2016).

2.2 The granulometry file problem.tgsd

The granulometry file is an ASCII file containing the characterization of the particle classes. The file format is described in Table 1 and the meaning of the used symbols is the following:

- nc: Total number of particle classes.

- diam: Class diameter (in mm).

- rho: Class density (in kg/m^3).

- sphe: Class sphericity.

- fc: Class mass fraction (0-1). It must verify that $\sum$ fc $= 1$.

- `class`: Label denoting class (particle/gas typology) such particle class, aggregate or gas.

As we mentioned above, file `problem.tgsd` can also be generated by the utility `fplume-tgsd`, distributed with this package, assuming either a Gaussian or bi-Gaussian distribution (in Φ-units).

nc				
diam(1)	rho(1)	sphe(1)	fc(1)	class(1)
...	...	...	...	...
diam(nc)	rho(nc)	sphe(nc)	fc(nc)	class(nc)

Table 1: Format of the granulometry file *problem*.`tgsd`

3 Output files

3.1 The log file `name.log`

The file `name.log` is an ASCII file where critical information about the simulation run is stored. The information written on the `name.log` contains:

- FPLUME copyright;

- starting time of the run and names of the input files;

- a short description atmospheric data used in the simulation;

- a short summary of the characteristics of the simulated plume, including geographical information about vent position, granulometric distribution, and modelling options;

- information about the simulation interval, mass flow rate, total plume height, neutral buoyancy level;

- number of warnings and errors during the simulation.

3.2 The file `name.01.res`

The output file `name.res` reports the profiles of the main variables as function of the height above the vent level.
It contains a header of two comment lines about the name and the unit of the variables, i.e.:

- height above vent level `z(a.v.)` in m;

- plume radius `r` in m;

- plume centre positions `x` and `y` in m;

- plume inclination angle `angle` in deg;

- plume density `rho` in kg/m^3;

- plume density normalized to air density `rho/rhoa`, dimensionless (-);

- plume temperature in degree Celsius ^{o}C;

- plume velocity `vel` in m/s;

- radial entrainment coefficient $\mathbf{a}_s$ (commonly denoted as α) dimensionless (-);

- cross-wind entrainment coefficient $\mathbf{a}_v$ (commonly denoted as β) dimensionless (-);

- percentage of entrained air in the plume `air`;

- percentage of water vapour in the plume `air`;

- percentage of liquid water in the plume `air`;

- percentage of iced water in the plume `air`;

- percentage of water vapour in the plume `air`;

3.3 Other output files

Besides the `name.log` and `name.res` files, Fplume creates two files describing mass distribution inside the plume (`name.01.mass-plume`) and leaving the plume (`name.01.mass-fall`) for the different particle sizes at different `z` levels.

Moreover other files needed for creating the optional gnuplot plots (i.e. `name.prof.pdf` and `name.01.res.pdf`) are also produced, e.g. `name.gdp`, `name.gnuplot`.

4 Program installation and execution

4.1 Installation requirements

Fplume is written in Fortran-90. The code has been tested in UNIX/Linux platforms including MacOS X. For compiling the code you need:

- A Fortran 90 compiler (e.g.: ifort, gfortran, xlf90, etc.)

For plotting the output data you can use `gnuplot`, freely available from `www.gnuplot.info`

4.2 Installation

Sources of Fplume are distributed as a gzipped tar archive in a file named `fplume-1.1.tar.gz`, available at the following URLs:

```
http://bsccase02.bsc.es/projects/fplume.html/
http://datasim.ov.ingv.it/fplume.html
```

After you downloaded the file `fplume-1.1.tar.gz`, copy (or move) it in a directory (eg: your home directory) and unpack the tarball with the command:

```
$ tar zxvf fplume-1.1.tar.gz
```

This creates the installation directory tree with the root directory named Fplume-1.1. The directory tree is shown in Table 2.

Table 2: Default structure of Fplume-1.1 sub-folders.

```
doc                        # Contains this manual
examples                   # Contains examples of input files
src                        # Contains the source files
```

The package comes with a `configure` script for automatically configuring your installation.

The 'configure' shell script attempts to guess correct values for various system-dependent variables used during compilation. Then uses these values to create a 'Makefile' in each directory of the package for compiling and installing the code and the scripts.

Generally, the configuration script automatically sets the compiler flags (eg: within the Linux environment), however, we suggested to check if the compiler optimization flag "-O" is correctly set and, eventually, manually set the environmental variable FCFLAGS=-O or set it in the `configure` command line (using `configure FCFLAGS=-O`).

In brief, to configure, compile and install Fplume it is enough to issue the following commands:

```
$ cd fplume-1.1
$ ./configure
```

```
$ make install
```

By default, the binary files are installed in the directory '`$HOME/bin`'. This default location can be changed by providing proper flags to the '`configure`' script, as described below (see flag `--prefix` or `--exec-prefix`).

It is important that the installation directory is listed in the shell PATH, otherwise the command FPLUME can be executed only by issuing its complete pathname.

4.3   Testing the installation

The installation can be tested by running the examples:

```
$ cd examples
$ fplume weak-wind
```

The following files are generated:

```
$ weak-wind.grn
$ weak-wind.log
$ weak-wind.mass.01
$ weak-wind.res.01
$ weak-wind.res.01.gnuplot
```

I a similar way, you may test the strong plumes example with "`fplume strong-wind`".

C-shell users should remind to issue the command "`rehash`" to recompute the internal hash table of the contents of the directories in the PATH variable.

4.4   Customizing the installation

You may choose different configuration parameters, including the fortran compiler, compilation flags, and the installation directory. The values of the different variables and flags are printed on the screen by "`configure`" at the end of the procedure (if no errors occur). During the configuration, you should pay attention to the information printed by the command "`configure`", also reported in file `config.log`.

A typical output of the `configure` script is shown in Table 3, where the list of the variables that are set during the installation is shown. The user settable variables are: `FC`, `FCFLAGS`, `F77`, `GNUPLOT`. These variables can be changed by defining them as shell environment variables or by passing their value as a argument in the "`configure`" command line. In case you specify a variable both in the environment and in the command line, the value passed in the command line takes the precedence.

Example (see Table 3): the default Fortran compiler found by `configure` is `FC=gfortran`. You can change the default value in the following ways (example for `ifort` compiler):

```
(Bourne shell)
$ export FC=ifort
$ export FCFLAGS=-O
$ ./configure
```

or

```
(C-shell)
$ setenv FC ifort
$ setenv FCFLAGS -O
$ ./configure
```

or

```
(any shell)
$ ./configure FC=ifort FCFLAGS=-O
```

All the variables listed before can be configured in a similar way. For example, you can set the path of the `gnuplot` binary, specified in the environmental variable (`GNUPLOT`) to a different value:

```
$ ./configure GNUPLOT=/usr/bin/gnuplot FCFLAGS=-O
```

Moreover you can specify the installation prefix (root directory for the installation of the binaries) with the flag `--prefix=DIRECTORY`. Example:

```
$ ./configure --prefix=$HOME/bin FC=gfortran F77=gfortran FCFLAGS="-g -O"
```

To disable the use of `GNUPLOT` you may use the directive "`--without-gnuplot` or `GNUPLOT=DISABLED`:

```
$ ./configure --without-gnuplot
```

or

```
$ ./configure GNUPLOT=DISABLED
```

A typical output of the =sc configure script is reported in Table 3.

Table 3: Typical output of the `configure` script

```
[... configure messages ...]
configure: ---------------------------------------------------------
configure: Configuration complete - fplume-1.1
configure:
configure: Fortran 90 compiler:         FC=gfortran
configure: Fortran flags:               FCFLAGS=-g -O
configure: Fortran 77 compiler:         F77=gfortran
configure: Gnuplot:                     GNUPLOT=gnuplot
configure: Install prefix:              --prefix=/home/username
configure: Executables install prefix:  --exec_prefix=${prefix}
configure: Binary directory:            --bindir=${exec_prefix}/bin
configure: ---------------------------------------------------------
```

Appendix A. Example of control input file

This is an example (not from a real case) of control input file. Comments begin with an exclamation symbol.

```
!
!  Example of an input file
!
TIME_UTC
!
!  YEAR : value (YYYY)
! MONTH : value (MM)
! DAY : value (DD)
! ERUPTION_START_(HOURS_AFTER_00) :  ndt values (start time of each eruptive phase)
!  ERUPTION_END_(HOURS_AFTER_00) :  value
!
YEAR = 2014
MONTH = 03
DAY = 01
ERUPTION_START_(HOURS_AFTER_00) = 0
ERUPTION_END_(HOURS_AFTER_00) = 2.5
!
SOURCE
!
!  LON_VENT : value
!  LAT_VENT : value
!  VENT_HEIGHT_(M) : value
!
!  SOLVE_PLUME_FOR options :  MFR/HEIGHT
! MFR_SEARCH_RANGE : Give n1 and n2, where:  10**n1 < MFR < 10**n2.  Only used if SOLVE_PLUME_FOR
= MFR
!
!  HEIGHT_ABOVE_VENT_(M) : ndt values.  Only used if SOLVE_PLUME_FOR = MFR
! MASS_FLOW_RATE_(KGS) : ndt values.  Only used if SOLVE_PLUME_FOR = HEIGHT
! EXIT_VELOCITY_(MS) : ndt values
!  EXIT_TEMPERATURE_(K) : ndt values
!  EXIT_WATER_FRACTION_(%) :  ndt values
!
!  TERMINAL_VELOCITY_MODEL options :  ARASTOOPOUR/GANSER/WILSON/DELLINO
!
LON_VENT = 0.0
LAT_VENT = 0.0
VENT_HEIGHT_(M) = 1500.
!
SOLVE_PLUME_FOR = HEIGHT ! MFR
MFR_SEARCH_RANGE = 8.0 10.0
!
HEIGHT_ABOVE_VENT_(M) = 37000.
MASS_FLOW_RATE_(KGS) = 1.5d9
EXIT_VELOCITY_(MS) = 275.
EXIT_TEMPERATURE_(K) = 1053.
EXIT_WATER_FRACTION_(%) = 5.
!
TERMINAL_VELOCITY_MODEL = ganser
!
AGGREGATION
!
```

```
!  AGGREGATION_MODEL options :  NONE/PERCENTAGE/CORNELL/COSTA
! FI_AGGREGATES : value.  Not used if AGGREGATION_MODEL = NONE
! DENSITY_AGGREGATES : value.  Not used if AGGREGATION_MODEL = NONE
! VSET_FACTOR : value.  Not used if AGGREGATION_MODEL = NONE
! PERCENTAGE_(%) :  value.  Only used if AGGREGATION_MODEL = PERCENTAGE
! FRACTAL_EXPONENT : value.  Only used if AGGREGATION_MODEL = COSTA
!
AGGREGATION_MODEL = NONE
FI_AGGREGATES = 2.
DENSITY_AGGREGATES = 350.
VSET_FACTOR = 1.0
PERCENTAGE_(%) = 20.
FRACTAL_EXPONENT = 2.99
!
METEO
!
!  METEO_DATA : PROFILE/STANDARD_ATMOSPHERE
!
METEO_DATA = PROFILE
!
CONSTANTS
!
! Cp :  value.  Specific heat of pyroclasts at constant pressure.  Default 1600 J/kgK
! Ca :  value.  Specific heat of air at constant pressure.  Default 1000 J/kgK
! Cw :  value.  Specific heat of water (vapour,liquid,ice) only if latent_heat=no.  Default
1000 J/kgK
!
Ca = 1000.0
Cp = 1100.0
Cw = 2000.0
!
MODELLING
!
!  WIND_COUPLING : yes/no.  If NO, Ua=0 is assumed
!  AIR_MOISTURE : yes/no.  If NO, wa=0 is assumed (dry entrained air only)
!  REENTRAINMENT : yes/no.  If NO, particle reentrainment is neglected (f=0)
!  LATENT_HEAT : yes/no.  If NO, latent heat contribution is neglected
!
! xi :  value. Constant in eq (1g).  Default 0.23 (Bursik 2001).
! zmin_wind :  value.  Ignore wind entrainment in the low jet region.  Default 0.0
! c_umbrella :  value.  Thickness of umbrella relative to Hb (>1).  Default 1.32
! ! a_s options :  CONSTANT value jet, value plume / KAMINSKI-R / KAMINSKI-C
! a_v options :  CONSTANT value) / TATE
!
WIND_COUPLING = YES
AIR_MOISTURE = YES
LATENT_HEAT = YES
REENTRAINMENT = NO
!
xi = 0.23
zmin_wind = 100.0
c_umbrella = 1.32
!
a_s = KAMINSKI-C
a_v = TATE
!a_s = CONSTANT 0.1 0.1
!a_v = CONSTANT 0.4
```

References

Bursik, M. I.: Effect of wind on the rise height of volcanic plumes, Geophys. Res. Lett., 18, 3621–3624, 2001.

Carmichael, R.: Definition of the 1976 Standard Atmosphere to 86 km, Public Domain Aeronautical Software (PDAS), Santa Cruz, CA (USA), URL http://www.pdas.com/atmosdef.html, web update 12 February 2013 by Ralph Carmichael, 2013.

Cornell, W., Carey, S., and Sigurdsson, H.: Computer simulation and transport of the Campanian Y-5 ash, J. Volcanol. Geotherm. Res., 17, 89–109, doi:10.1016/0377-0273(83)90063-X, 1983.

Costa, A., Folch, A., and Macedonio, G.: A model for wet aggregation of ash particles in volcanic plumes and clouds: I. Theoretical formulation, J. Geophys. Res., 115, doi:10.1029/2009JB007175, 2010.

Folch, A., Costa, A., and Macedonio, G.: FPLUME-1.0: An integral volcanic plume model accounting for ash aggregation, Geosci. Model Dev., 9, 1–20, doi:10.5194/gmd-9-1-2016, 2016.

Kaminski, E., Tait, S., , and Carazzo, G.: Turbulent entrainment in jets with arbitrary buoyancy, J. Fluid Mech., 526, 361–376, doi:10.1017/S0022112004003209, 2005.

Rouse, H., Yih, C. S., and Humphreys, H. W.: Gravitational convection from a boundary source, Tellus, 4, 201–210, 1952.

Sparks, R. S. J.: The dimensions and dynamics of volcanic eruption columns, Bull. Volcanol., 48, 3–15, doi:10.1007/BF01073509, 1986.

Sulpizio, R., Folch, A., Costa, A., Scaini, C., and P., D.: Hazard assessment of far-range volcanic ash dispersal from a violent Strombolian eruption at Somma-Vesuvius volcano, Naples, Italy: Implications on civil aviation, Bull. Volcanol., 74, 2205–2218, doi:10.1007/s00445-012-0656-3, 2012.

Tate, P. M.: The rise and dilution of buoyant jets and their behaviour in an internal wave field, Phd thesis, University of New South Wales, School of Mathematics, URL http://trove.nla.gov.au/version/19798635, last access: 16 September 2015, 2002.