Hazmap

Computer Code and Related Documentation

Arnau Folch, Antonio Costa and Giovanni Macedonio

Istituto Nazionale di Geofisica e Vulcanologia
Sezione "Osservatorio Vesuviano"
Via Diocleziano 326 -80124 Napoli, Italy

February 2008

Contents

1 Introduction

HAZMAP is a FORTRAN90 code for the solution of the equation of diffusion, transport and sedimentation of small particles, in order to model the dispersion of ash generated by a convective column. Under the approximations of a constant horizontally uniform wind field and negligible vertical advection and diffusion, this equation reduces to:

$$\frac{\partial C_j}{\partial t} + u_x \frac{\partial C_j}{\partial x} + u_y \frac{\partial C_j}{\partial y} - \frac{\partial (v_{sj} C_j)}{\partial z} = K \left(\frac{\partial^2 C_j}{\partial x^2} + \frac{\partial^2 C_j}{\partial y^2} \right) + S_j \tag{1}$$

where C_j is the concentration of the particle class j having a settling velocity v_{sj}, $\mathbf{u} = (u_x, u_y)$ is the wind velocity, K is the (constant) horizontal turbulent diffusion coefficient, and S_j is the source term. Equation (1) is solved for each particle class independently, *i.e.* assuming no interaction between particles belonging to different classes during the transport process. The generic particle class j is defined by triplet of values characterizing each particle (d_p, ρ_p, F_p), that are, respectively, diameter, density, and a shape factor. For d_p we use the equivalent diameter d, which is the diameter of a sphere of equivalent volume. For the shape factor F_p we choose the sphericity ψ, which is the ratio of the surface area of a sphere with diameter d to the surface area of the particle. In our approximation, each triplet (d, ρ_p, ψ) is sufficient to define the settling velocity value v_{sj}.
Since equation (1) is linear in mass, an instantaneous release of the total mass from the eruption column can be assumed if wind and diffusion parameters do not change significantly with time and only the final deposit is needed. This quasi-steady approach is assumed to hold during each simulation time interval. Considering these approximations, the above equation has a semi-analytical solution as described in (Macedonio et al., 2005). The computational domain is split into thin horizontal layers that fall to the ground together with the particles originally contained in a given initial vertical interval $[z_i, z_{i+1}]$ at time $t = 0$. An analytical solution is then found for each layer. Since the whole treatment is done separately for each class of particles and no vertical diffusion and wind advection takes place, all particles falling from the same initial height remain at all times at the same altitude. While the centre of each cloud is translated by wind, the cloud spreads horizontally due to diffusion and settles by gravity until it reaches the ground where it forms the deposit. The model outputs therefore accumulations on the ground for each granulometric class. For further details see (Macedonio et al., 2005) and (Pfeiffer et al., 2005).
There are several semi-empirical parameterizations for the particle settling velocity v_s if one assumes that particles settle down at their terminal velocity:

$$v_s = \sqrt{\frac{4g\,(\rho_p - \rho_a)\,d}{3 C_d \rho_a}} \tag{2}$$

where ρ_a and ρ_p denote air and particle density, respectively, d is the particle equivalent diameter, and C_d is the drag coefficient. C_d depends on the Reynolds number, $Re = d v_s / \nu_a$ ($\nu_a = \mu_a / \rho_a$ is the kinematic viscosity of air, μ_a the dynamic viscosity). In several options are possible for estimating settling velocity, such as:

1. ARASTOOPOUR model (Arastoopour et al., 1982):

$$C_d = \begin{cases} \frac{24}{Re}(1 + 15 Re^{0.687}) & Re \le 10^3 \\ 0.44 & Re > 10^3 \end{cases} \tag{3}$$

   valid for spherical particles only.

2. GANSER model (Ganser , 1993):

$$C_d = \frac{24}{Re K_1} \left\{ 1 + 0.1118 \left[Re \,(K_1 K_2) \right]^{0.6567} \right\} + \frac{0.4305 K_2}{1 + \frac{3305}{Re K_1 K_2}} \tag{4}$$

   where $K_1 = 3/(1 + 2\psi^{-0.5})$, $K_2 = 10^{1.84148(-\mathrm{Log}\psi)^{0.5743}}$ are two shape factors, and ψ is the particle sphericity ($\psi = 1$ for spheres).

3. `WILSON` model (Wilson and Huang , 1979) using the interpolation suggested by Pfeiffer et al. (2005):

$$C_d = \begin{cases} \dfrac{24}{Re}\varphi^{-0.828} + 2\sqrt{1.07-\varphi} & Re \le 10^2 \\ 1 - \dfrac{1 - C_d|_{Re=10^2}}{900}(10^3 - Re) & 10^2 \le Re \le 10^3 \\ 1 & Re \ge 10^3 \end{cases} \tag{5}$$

   where $\varphi = (b+c)/2a$ is the particle aspect ratio ($a \ge b \ge c$ denote the particle semi-axes).

4. `DELLINO` model (Dellino et al., 2005):

$$v_s = 1.2605\frac{\nu_a}{d}\left(Ar\,\xi^{1.6}\right)^{0.5206} \tag{6}$$

   where $Ar = d^3(\rho_p - \rho_a)\rho_a/\mu_a^2$ is the Archimedes number, and ξ is a particle shape factor (sphericity to circularity ratio).

Since for Hazmap the primary particle shape factor is the sphericity ψ, for sake of simplicity, it calculates φ in (5) and ξ in (6) approximating particles as prolate ellipsoids.

2 Program setup

The Hazmap package comes with a set of utility programs which can be used to generate input files in the format required by Hazmap or to postprocess the results. The order of execution is the following (see Figure 1): (i) run the program SetGrn to generate the granulometry file, (ii) run the program SetSrc to generate the source term file, (iii) run Hazmap and, finally, (iv) run the program HazmapPostp to postprocess the results. Steps (i) and (ii) can be avoided if the user furnish the granulometry and source files directly.

2.1 Installation

- On a Windows OS download and decompress the file `hazmap-3.1.tar.gz` on your selected directory. The `hazmap-3.1.tar` file already contains Windows executables for Hazmap and other utility programs so that it is not strictly necessary to have a `FORTRAN90` compiler. The untaring of `hazmap-3.1.tar` will create the folders described in Table 1.

- On a Unix/Linux/Mac X operating system:

  1. Decompress and then untar the file `hazmap-3.1.tar` issuing the command "`tar xvf hazmap-3.1.tar`". This will generate directory `Hazmap-3.1` (see Table 1).
  2. Compile the program Hazmap . Enter the directory `Hazmap-3.1/Sources`, then issue the command "`make`" to produce the executable Hazmap.exe. You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files. If you are not going to run on a Windows platform you can also delete the executable Hazmap.win.exe.
  3. Compile the (optional) utility program SetGrn . Enter the directory `Hazmap-3.1/Utilities/SetGrn/Sources`, then issue the command "`make`" to produce the executable SerGrn.exe. You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files. If you are not going to run on a Windows platform you can also delete the executable SetGrn.win.exe.
  4. Compile the (optional) utility program SetSrc . Enter the directory `Hazmap-3.1/Utilities/SetSrc/Sources`, then issue the command "`make`" to produce the executable SerSrc.exe. You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files. If you are not going to run on a Windows platform you can also delete the executable SetSrc.win.exe.

Level 1	Level 2	Level 3	Description
Hazmap-3.1	Documents		Contains this manual.
	Runs	Example	Contains the example run.
	Scripts		Contains the script files.
	Sources		HAZMAP sources.
	Utilities	SetGrn	SETGRN utility program. See section 5.1.
		SetSrc	SETSRC utility program. See section 5.2.
		HazmapPostp	HAZMAPPOSTP utility program. See section 5.3.

Table 1: Default structure of HAZMAP folders.

5. Compile the (optional) utility program HAZMAPPOSTP . Enter the directory `Hazmap-3.1/Utilities/HazmapPostp/Sources`, then issue the command "`make`" to produce the executable HazmapPostp.exe. You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files. If you are not going to run on a Windows platform you can also delete the executable HazmapPostp.win.exe.

2.2 Folder structure

Table 1 shows the folder structure. The directory `Hazmap-3.1/Sources` contains the HAZMAP source files, the directory `Hazmap-3.1/Utilities` contains the programs SETGRN , SETSRC and HAZMAPPOSTP and, finally, the directory `Hazmap-3.1/Runs` contains the runs, one within each own folder. An example run named "Example" is provided with the installation.

2.3 Program run

HAZMAP can be launched typing
"`Hazmap.exe FileInp FileSrc FileGrn FileMet FileLst FileRes`" (on a Unix/Linux/Mac X OS)
or
"`Hazmap.win.exe FileInp FileSrc FileGrn FileMet FileLst FileRes`" (on a Windows OS), where

- `FileInp`: Name (including path) of the control input file (see section 3.1).
- `FileSrc`: Name (including path) of the source input file (see section 3.2).
- `FileGrn`: Name (including path) of the granulometry input file (see section 3.3).
- `FileMet`: Name (including path) of the meteo data input file (see section 3.4).
- `FileLst`: Name (including path) of the output list file (see section 4.1).
- `FileInp`: Name (including path) of the HAZMAP results file (see section 4.2).

Note that filenames (and locations) are passed as a program call argument. It is highly recomended to launch HAZMAP through the script files included in the distribution.

- On a Windows OS enter the folder `Hazmap-3.1/Scripts`, edit the script `Script-Hazmap-Win.bat` to change the "problemname" variable and launch the script.
- On a Mac X/Unix/Linux OS enter the folder `Hazmap-3.1/Scripts`, edit the script `Script-Hazmap-Unix` to change the "problemname" variable and launch the script.

<u>NOTE</u>: To create a new run you can simply create a new folder, copy the control input file of the example (Example.inp) and modifiy the script line which defines the "problemname" variable.

3 The HAZMAP input files

3.1 The control file `FileInp`

The HAZMAP control file is passed to the program as a call argument. This file is made up with a set of blocks that define all the computational and physical parameters needed by the dispersion model (Table 2 shows an example of control file). Parameters within a block are listed one per record, in arbitrary order, and can optionally be followed by one or more blank spaces and a comment. A detailed description of each record is given below. Real numbers can be expressed following the FORTRAN notation (*e.g.*, $12e7 = 12 \times 10^7$).

3.1.1 BLOCK TIME_UTC

This block of data defines variables related to time.

- `YEAR`: Current year.
- `MONTH`: Current month (1-12).
- `DAY`: Current day (1-31).
- `RUN_START_(HOURS_AFTER_00)`: Run start hour (after 0000UTC).
- `RUN_END_(HOURS_AFTER_00)`: Run end hour (after 0000UTC).

3.1.2 BLOCK GRID

This block of data defines variables related to the grid at ground.

- `X_ORIGIN_(UTM_M)`: x-coordinate of the grid bottom left corner (UTM coordinates in m).
- `Y_ORIGIN_(UTM_M)`: y-coordinate of the grid bottom left corner (UTM coordinates in m).
- `CELL_SIZE_(KM)`: Grid spacing (in km).
- `NX`: Number of grid cells along the x-direction.
- `NY`: Number of grid cells along the y-direction.

3.1.3 BLOCK HAZMAP

This block of data defines the rest of variables needed by the program.

- `ZLAYER_(M)`: Heights (in m) of the z-layers in terrain following coordinates, *i.e.* above the vent. It is not necessary to specify the number of vertical layers since it is automatically calculated by the program. Alternatively, for regular z-layering, the user can also specify the initial value (z_o), the final value (z_f), and the increment (Δz) using the format:
  `ZLAYER_(M) FROM` z_o `TO` z_f `INCREMENT` Δz
- `TERMINAL_VELOCITY_MODEL`: Type of terminal settling velocity model. Possibilities are `ARASTOOPOUR` (Arastoopour et al., 1982), `GANSER` (Ganser , 1993), `WILSON` (Wilson and Huang , 1979) and `DELLINO` (Dellino et al., 2005).
- `HORIZONTAL_DIFFUSION_COEFFICIENT_(M2/S)`: Value of the diffusion coefficient K (in m^2/s).
- `POSTPROCESS_TIME_INTERVAL_(HOURS)`: Time interval to output results (in h).

```
TIME_UTC
YEAR = 2007
MONTH = 03
DAY = 01
RUN_START_(HOURS_AFTER_00) = 0.
RUN_END_(HOURS_AFTER_00) = 3.

GRID
X_ORIGIN_(UTM_M) = 450000.
Y_ORIGIN_(UTM_M) = 4125000.
CELL_SIZE_(KM) = 2.0
NX = 51
NY = 51

HAZMAP
Z_LAYER_(M) FROM 0.  TO 5000.  INCREMENT 250.
TERMINAL_VELOCITY_MODEL = Ganser
HORIZONTAL_DIFFUSION_COEFFICIENT_(M2/S) = 2500.
POSTPROCESS_TIME_INTERVAL_(HOURS) = 3.
```

Table 2: Sample of the input control file `FileInp`.

3.2 The source file `FileSrc`

The Hazmap source file is an ASCII file containing the definition of the source term. The source is defined at time intervals during which source values are kept constant. The number, position and values (*i.e.* Mass Flow Rate) of the source points can, however, vary from one time slice to another. There is no restriction on the number and duration of the time intervals. It allows, in practise, to discretize any type of source term. This file can be created by the utility program SetSrc . The file format is described in Table 3 and the meaning of the used symbols is the following:

- `itime1`: Starting time (in sec after 00UTC) of the time slice.
- `itime2`: End time (in sec after 00UTC) of the time slice.
- `nsrc`: Number of source points (can vary from one interval to another).
- `nc`: Number of granulometric classes.
- `MFR`: Mass flow rate (in kg/s).
- `x`: x-coordinate of the source isrc (UTM coordinates in m).
- `y`: y-coordinate of the source isrc (UTM coordinates in m).
- `z`: z-coordinate of the source isrc (terrain following coordinates in m, *i.e.* above the vent).
- `src`: Mass flow rate (in kg/s) of each granulometric class for this point source. It must be verified that $\sum\sum src(isrc, ic) = MFR$.

3.3 The granulometry file `FileGrn`

The granulometry file is an ASCII file containing the definition of the particle classes (a class is characterized by particle size, density and sphericity). This file can be created by the utility program SetGrn . Note that SetGrn only generates distributions which are Gaussian in Φ and linear in ρ and ψ. Hazmap can obviously handle with other distributions but, in this case, the granulometry file must be supplied directly by the user. The file format is described in Table 4 and the meaning of the used symbols is the following:

itime1 itime2
nsrc nc
MFR
x y z src(1,1) ... src(1,nc)
...
x y z src(nsrc,1) ... src(nsrc,nc)

Table 3: Format of the source file `FileSrc`. Repeat thes block for each time slice

.

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

Table 4: Format of the granulometry file `FileGrn`.

- `nc`: Number of granulometric classes.
- `diam`: Class diameter (in mm).
- `rho`: Class density (in kg/m^3).
- `sphe`: Class sphericity.
- `fc`: Class mass fraction (0-1). If must verify that $\sum$ fc = 1.

3.4 The meteo data file `FileMet`

The HAZMAP meteo data file is an ASCII file containing wind velocities and air temperatures at different heights and for each time slice. The vertical layers at which meteo data is provided can be different in number and location from the HAZAMP z-layers defined in the record Z_LAYER_(M) of the control input file. The program automatically interpolates data from the formers to the latters. The file format is described in Table 5 and the meaning of the used symbols is the following:

- `itime1`: Starting time (in sec after 00UTC) of the time slice.
- `itime2`: End time (in sec after 00UTC) of the time slice.
- `nz`: Number of z-layers at which meteo data is provided.
- `z`: z-coordinate of the layer (terrain following coordinates in m).
- `ux`: x-component of the wind velocity (in m/s).
- `uy`: y-component of the wind velocity (in m/s).
- `T`: Temperature (in ^oC).

itime1 itime2
nz
z(1) ux(1) uy(1) T(1)
...
z(nz) ux(nz) uy(nz) T(nz)

Table 5: Format of the meteo file `FileMet`. Repeat this block for each time slice.

4 The HAZMAP output files

4.1 The list file `FileLst`

This file contains information concerning the run (summary of input data, run time error messages, CPU time, *etc.*).

4.2 The results file `FileRes`

This is a binary file with the results from a HAZMAP run. Results are output at the nodes of a regular 2D grid. This file must be processed by the HAZMAPPOSTP utility to produce "human readeable" files (normally in GRD format).

5 The utility programs

5.1 The program SetGrn

The granulometric distribution is defined in the granulometry file (see section 3.3). The program SetGrn is an utility that reads the GRANULOMETRY block from the control input file and generates the granulometry file assuming that the mass fraction of particles follows either a linear or Gaussian distribution in Φ and that the density of particles varies linearly with Φ. Note that other granulometric distributions different from a linear/Gaussian can also be considered. However, in this case, the Hazmap granulometry file can not be generated by SetGrn but must be supplied directly by the user.

5.1.1 Program execution

To run SetGrn (see section 2.1 for installation details) simply type
"`SetGrn.exe FileLog FileInp FileGrn`" (on a Unix/Linux/Mac X OS) or
"`SetGrn.win.exe FileLog FileInp FileGrn`" (on a Windows OS) where

- `FileLog`: Name (including path) of the SetGrn output log file.
- `FileInp`: Name (including path) of the control input file that contains the GRANULOMETRY block. Normally this file coincides with the Hazmap input file.
- `FileGrn`: Name (including path) of the granulometry file. This is the output from SetGrn that is used later by Hazmap as input.

Note that filenames are passed as a program call argument. It is highly recomended to launch SetGrn through the script files included in the distribution.

- On a Windows OS go to the folder `Hazmap-3.1/Scripts`, edit the script `Script-SetGrn-Win.bat` to change the "problemname" variable and launch the script.
- On a Mac X/Unix/Linux OS enter the folder `Hazmap-3.1/Scripts`, edit the script `Script-SetGrn-Unix` to change the "problemname" variable and launch the script.

5.1.2 The GRANULOMETRY block

This block of data (see Table 6) defines the variables needed by the SetGrn program. Commonly this block is appended to Hazmap control input file. The meaning of each record is the following:

- `NUMBER_OF_CLASSES`: Number of granulometric classes.
- `FI_MEAN`: Mean value of Φ (Gaussian distribution).
- `FI_DISP`: Value of σ in the Gaussian distribution.
- `FI_RANGE`: Minimum and maximum values of Φ (Φ_{min} and Φ_{max} respectively).
- `DENSITY_RANGE`: Values of density (in kg/m^3) associated to Φ_{min} and Φ_{max} particles. Lineal interpolation is assumed.
- `SPHERICITY_RANGE`: Values of sphericity associated to Φ_{min} and Φ_{max} particles. Lineal interpolation is assumed.

5.2 The program SetSrc

The distribution of sources is defined in a source file (see section 3.2). The program SetSrc is an utility that reads the SOURCE block from the control input file and generates a source file. The source term is constant for a given time interval but there is no limit on the number and duration of the time intervals. It allows, in practise, to discretize any kind of time-dependency (time-dependent mass flow rate, column height, etc.). The program admits three possibilities: point source (mass is released in a single source point), Suzuki distribution (Suzuki , 1983; Pfeiffer et al., 2005), and buoyant plume model (Bursik , 2001). The last option is more elaborated and involves the solution of the 1D radial-averaged plume

```
GRANULOMETRY
NUMBER_OF_CLASSES = 12
FI_MEAN = 1.5
FI_DISP = 1.5
FI_RANGE = -4.  5.
DENSITY_RANGE = 900.  2600.
SPHERICITY_RANGE = 0.8 0.9
```

Table 6: Sample of the SetGrn input file. Normally this block is appended at the end of the Hazmap input file.

governing equations that describe the convective region of an eruptive column. These equations are intimately coupled with the wind field which, for small to medium size plumes, may induce a substantial plume bent-over and subsequent variations of plume height and mass release location. For this reason, when this option switched on, the program reads the values of the wind field from a meteorological file, computes the averaged wind direction and solves the plume governing equations for each time interval and particle class accounting for wind. Note that it introduces a time dependence in the source term even when all the eruptive parameters (mass flow rate, class fraction, etc.) are kept constant in time.

5.2.1 Program execution

To run SetSrc (see section 2.1 for installation details) simply type
"`SetSrc.exe FileLog FileInp FileSrc FileGrn FileMet`" (on a Unix/Linux/Mac X OS) or
"`SetSrc.win.exe FileLog FileInp FileSrc FileGrn FileMet`" (on a Windows OS) where

- `FileLog`: Name (including path) of the SetSrc output log file.
- `FileInp`: Name (including path) of the control input file that contains the SOURCE block. Normally this file coincides with the Hazmap input file.
- `FileSrc`: Name (including path) of the source file. This is the output from SetSrc that is used later by Hazmap as input.
- `FileGrn`: Name (including path) of the granulometry file (normally generated previously by SetGrn).
- `FileMet`: Name (including path) of the meteo data file (see section 3.4).

Note that filenames are passed as a program call argument. It is highly recomended to launch SetSrc through the script files included in the distribution.

- On a Windows OS go to the folder `Hazmap-3.1/Scripts`, edit the script `Script-SetSrc-Win.bat` to change the "problemname" variable and launch the script.
- On a Mac X/Unix/Linux OS enter the folder `Hazmap-3.1/Scripts`, edit the script `Script-SetSrc-Unix` to change the "problemname" variable and launch the script.

5.2.2 The SOURCE block

This block of data (see Table 7) defines the variables needed by SetSrc . Commonly this block is appended to the Hazmap control input file. The meaning of each record is the following:

- `X_VENT_(UTM_M)`: x-coordinate of the vent (UTM coordinates in m).
- `Y_VENT_(UTM_M)`: y-coordinate of the vent (UTM coordinates in m).
- `ZVENT_(M)`: Height (in m a.s.l.) of the vent.

`SOURCE`	
`X_VENT_(UTM_M) = 500080.`	
`Y_VENT_(UTM_M) = 4177690.`	
`Z_VENT_(M) = 3030.`	
`MASS_FLOW_RATE_(KGS) = 5d4`	(One value for each source time interval)
`SOURCE_TYPE = PLUME`	
`POINT_SOURCE`	(Variables below are used only if `SOURCE_TYPE = POINT`)
`HEIGHT_ABOVE_VENT_(M) = 2000.`	
`SUZUKI_SOURCE`	(Variables below are used only if `SOURCE_TYPE = SUZUKI`)
`HEIGHT_ABOVE_VENT_(M) = 3000.`	(One value for each source time interval)
`A = 4.`	(One value for each source time interval)
`L = 5.`	(One value for each source time interval)
`PLUME_SOURCE`	(Variables below are used only if `SOURCE_TYPE = PLUME`)
`EXIT_VELOCIY_(MS) = 100.`	
`EXIT_TEMPERATURE_(K) = 1073.`	
`EXIT_VOLATILE_FRACTION_(IN%) = 0.`	

Table 7: Sample of the SetSrc input file. Normally this block is appended at the end of the Hazmap input file. In this example a Suzuki source and two time intervals are assumed.

- `MASS_FLOW_RATE_(KGS)`: Values of the mass flow rate (in kg/s). One value for each time interval. The duration of each time interval is constant and given by `RUN_START_(HOURS_AFTER_00)` minus `RUN_END_(HOURS_AFTER_00)` divided by the number of time intervals (automatically computed by the program from the number of values).
- `SOURCE_TYPE`: Type of source distribution. Possibilities are POINT, SUZUKI or PLUME.
- `HEIGHT_ABOVE_VENT_(M)`: Height of the source (in m above the vent). One value for each time interval.
- `A`: Parameter A in the Suzuki distribution. One value for each time interval. Used only if `SOURCE_TYPE = SUZUKI`.
- `L`: Parameter L in the Suzuki distribution. One value for each time interval. Used only if `SOURCE_TYPE = SUZUKI`.
- `EXIT_VELOCIY_(MS)`: Magma exit velocity (in m/s) at the vent. One value for each time interval. Used only if `SOURCE_TYPE = PLUME`.
- `EXIT_TEMPERATURE_(K)`: Magma exit temperature (in ^oK) at the vent. One value for each time interval. Used only if `SOURCE_TYPE = PLUME`.
- `EXIT_VOLATILE_FRACTION_(IN%)`: Magma volatile mass fraction at the vent. One value for each time interval. Used only if `SOURCE_TYPE = PLUME`.

5.3 The program HazmapPostp

The program HazmapPostp is an optional utility that reads the output binary file of Hazmap , calculates some relevant quantities at selected time instants and produces elementary maps in GRD and PS formats. Files in GRD format can be readed directly by several plotting programs like the commercial software `GRAPHER`. Alternativelly, the user may also generate its own plots using functions from several free packages (*e.g.* gnuplot in Fortran).

5.3.1 Program execution

To run HazmapPostp (see section 2.1 for installation details) simply type
"`HazmapPostp.exe FileLog FileInp FileRes PATHRES FileTop FileSym`" (on a Unix/Linux/Mac X OS) or

"`HazmapPostp.win.exe FileLog FileInp FileRes PATHRES FileTop FileSym`" (on a Windows OS) where

- `FileLog`: Name (including path) of the HazmapPostp output log file.
- `FileInp`: Name (including path) of the control input file that contains the POSTPROCESS_MODELS block. Normally this file coincides with the Hazmap input file.
- `FileRes`: Name (including path) of the Hazmap results file. This is the output from Hazmap that is used by HazmapPostp as input.
- `PATHRES`: Path where the HazmapPostp output files are dump.
- `FileTop`: Name (including path) of the reginal topography file (see section 5.3.3).
- `FileSym`: Name (including path) of the symbols file (see section 5.3.4). This file is optionally used by HazmapPostp to plot symbols and legends in the PS map files.

Note that filenames are passed as a program call argument. It is highly recomended to launch Hazmap-Postp through the script files included in the distribution.

- On a Windows OS go to the folder `Hazmap-3.1/Scripts`, edit the script `Script-HazmapPostp-Win.bat` to change the "problemname" variable and launch the script.
- On a Mac X/Unix/Linux OS enter the folder `Hazmap-3.1/Scripts`, edit the script `Script-HazmapPostp-Unix` to change the "problemname" variable and launch the script.

5.3.2 The POSTPROCESS_MODELS block

This block of data (see Table 8) defines the variables needed by HazmapPostp . Commonly this block is appended to the Hazmap control input file. The meaning of each record is the following:

- `OUTPUT_FILES_IN_GRD_FORMAT`: Possibilities are YES or NO. If YES, HazmapPostp plots files in GRD format. Files in GRD format can be readed directly by several plotting programs like the commercial software GRAPHER. Alternativelly, the user may also generate its own plots using functons from several free packages (*e.g.* gnuplot in FORTRAN).
- `OUTPUT_FILES_IN_PS_FORMAT`: Possibilities are YES or NO. If YES, HazmapPostp plots files in PS format.
- `MAP_TOTAL_LOAD`: Possibilities are YES or NO. If YES HazmapPostp plots the total ground load.
- `UNITS`: Units of `MAP_TOTAL_LOAD`. It must be KG/M2.
- `CONTOUR_LEVELS`: Values of the contour levels for `MAP_TOTAL_LOAD`. Only used when `OUTPUT_FILES_IN_PS_FORMAT` is YES.
- `MAP_CLASS_LOAD`: Possibilities are YES or NO. If YES HazmapPostp plots the class ground load.
- `UNITS`: Units of `MAP_CLASS_LOAD`. It must be KG/M2.
- `CONTOUR_LEVELS`: Values of the contour levels for `MAP_CLASS_LOAD`. Only used when `OUTPUT_FILES_IN_PS_FORMAT` is YES.
- `MAP_DEPOSIT_THICKNESS`: Possibilities are YES or NO. If YES HazmapPostp plots total deposit thickness.
- `UNITS`: Units of `MAP_DEPOSIT_THICKNESS`. Possibilities are MM (for *mm*), CM (for *cm*), and M (for *m*).
- `COMPACTATION_FACTOR`: Deposit compactation factor.
- `CONTOUR_LEVELS`: Values of the contour levels for `MAP_DEPOSIT_THICKNESS`. Only used when `OUTPUT_FILES_IN_PS_FORMAT` is YES.

`POSTPROCESS_MODELS`	
`OUTPUT_FILES_IN_GRD_FORMAT = YES`	
`OUTPUT_FILES_IN_PS_FORMAT = NO`	
`MAP_TOTAL_LOAD = YES`	
`UNITS = KG/M2`	
`CONTOUR_LEVELS = 0.1 0.25 0.5 1. 5. 10. 50.`	(Only used if `OUTPUT_FILES_IN_PS_FORMAT=YES`)
`MAP_CLASS_LOAD = NO`	
`UNITS = KG/M2`	
`CONTOUR_LEVELS = 0.1 0.25 0.5 1. 5. 10. 50.`	(Only used if `OUTPUT_FILES_IN_PS_FORMAT=YES`)
`MAP_DEPOSIT_THICKNESS = NO`	
`UNITS = MM`	
`COMPACTATION_FACTOR = 0.7`	
`CONTOUR_LEVELS = 0.1 1. 5. 10. 50. 100. 500.`	(Only used if `OUTPUT_FILES_IN_PS_FORMAT=YES`)

Table 8: Sample of the HazmapPostp input file. Normally this block is appended at the end of the Hazmap input file.

DSAA
nx ny
xo xf
yo yf
zmin zmax
z(1,1) ... z(1,nx)
...
z(ny,1) ... z(ny,nx)

Table 9: Format of the topography file `FileTop`.

5.3.3 The regional topography file `FileTop`

This is a file in GRD format containing the topography, typically at a regional scale. It is used by program to plot topography in the maps (Hazmap does not currently include topography). The computational domain must lay within the bounds of the topography file. The GRD file format is described in Table 9 and the meaning of the used symbols is the following:

- `nx`: Number of cells in the x-direction.
- `ny`: Number of cells in the y-direction.
- `xo`: x-coordinate of the grid bottom left corner (UTM coordinates in m).
- `xf`: x-coordinate of the grid top right corner (UTM coordinates in m).
- `yo`: y-coordinate of the grid bottom left corner (UTM coordinates in m).
- `yf`: y-coordinate of the grid top right corner (UTM coordinates in m).
- `zmin`: Minimum value of z in the domain. Not used (you can use a void value).
- `zmax`: Maximum value of z in the domain. Not used (you can use a void value).
- `z`: Height (in m) of each grid point.

xs ys sname scode 1
...

Table 10: Format of the symbols file `FileSym`. Repeat the line for each symbol to be plotted.

5.3.4 The symbols file `FileSym`

This is a file in ASCII format containing geographic information. It is optional and used by HazmapPostp program when `OUTPUT_FILES_IN_PS_FORMAT = YES` to plot symbols in the PS files. The file format is described in Table 10 and the meaning of the used symbols is the following:

- `xs`: x-coordinate of the symbol (UTM in m).
- `ys`: y-coordinate of the symbol (UTM in m).
- `sname`: Name of the symbol as will appear in the PS file.
- `scode`: Symbol code. See Figure 2.

6 Application example

The Hazmap package includes an application example to check that the installation and compilation of the procedure has been done sucessfully. The example considers an eruption occurring at the first of March 2007. The files needed to run the example are located in the folder `Hazmap-3.1/Runs/Example` and are:

- Example.inp. Problem control input file `FileInp` (see Table 2).
- Example.met. Meteo file `FileMet`. Needed by SetSrc and Hazmap .
- Example.regionaltopo.grd. Regional topography file `FileTop`. Needed by HazmapPostp .
- Example.sym. Symbols file `fileSym`. Optionally used by HazmapPostp .

To run the example it is necesary to proceed as follows (Scripts are located in the folder `Hazmap-3.1/Scripts`):

1. Launch the script `Script-SerGrn-Unix` (or `Script-SetGrn-Win.bat` in Windows OS) to run SetGrn .
2. Launch the script `Script-SerSrc-Unix` (or `Script-SetSrc-Win.bat` in Windows OS) to run SetSrc .
3. Launch the script `Script-Hazmap-Unix` (or `Script-Hazmap-Win.bat` in Windows OS) to run Hazmap .
4. Launch the script `Script-HazmapPostp-Unix` (or `Script-HazmapPostp-Win.bat` in Windows OS) to run HazmapPostp .

Figure 3 shows the Hazmap deposit load at the end of the run.

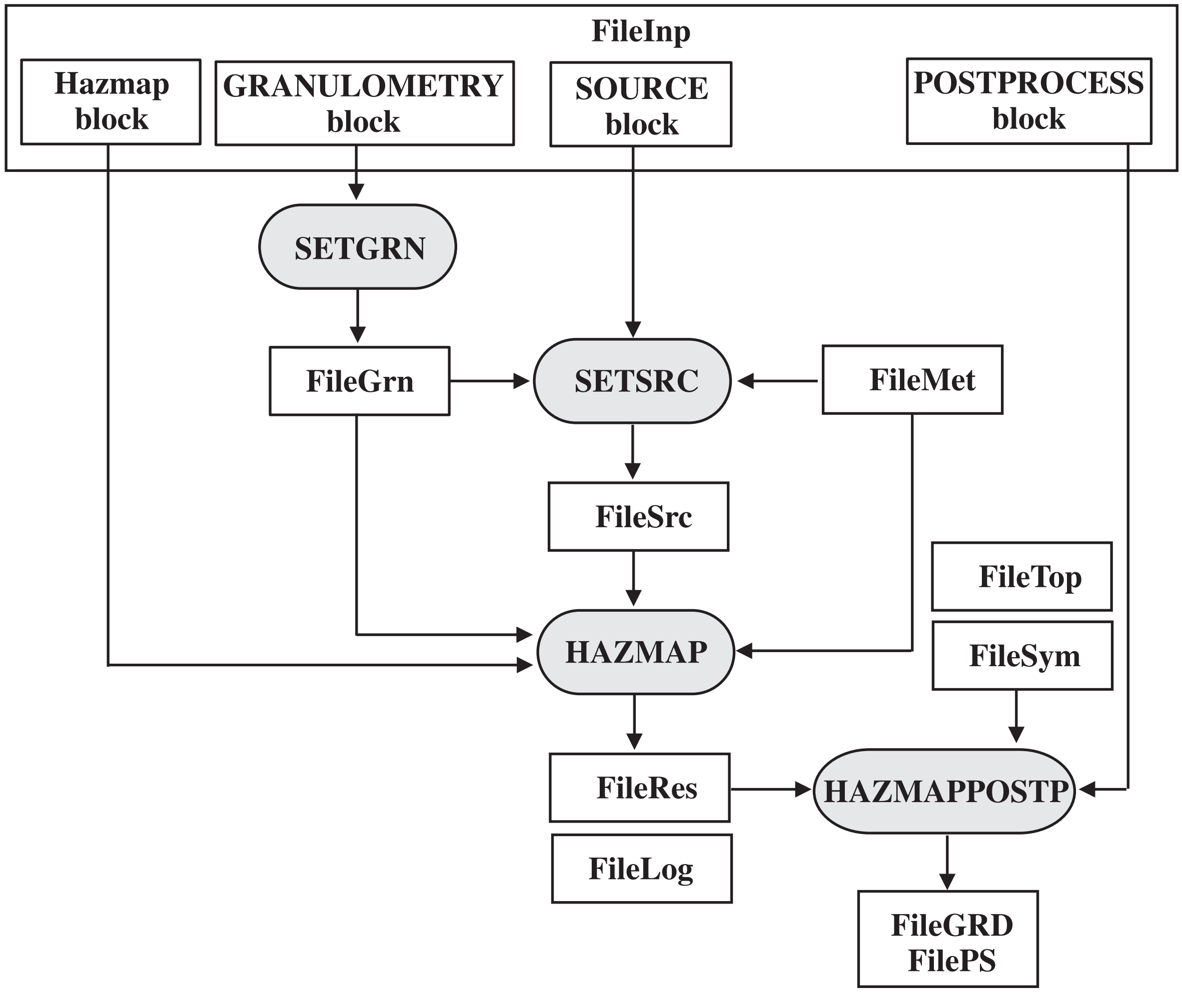

Figure 1: Execution flow for HAZMAP and the utility programs. Boxes indicate I/O files. File names are passed to programs as a call argument.

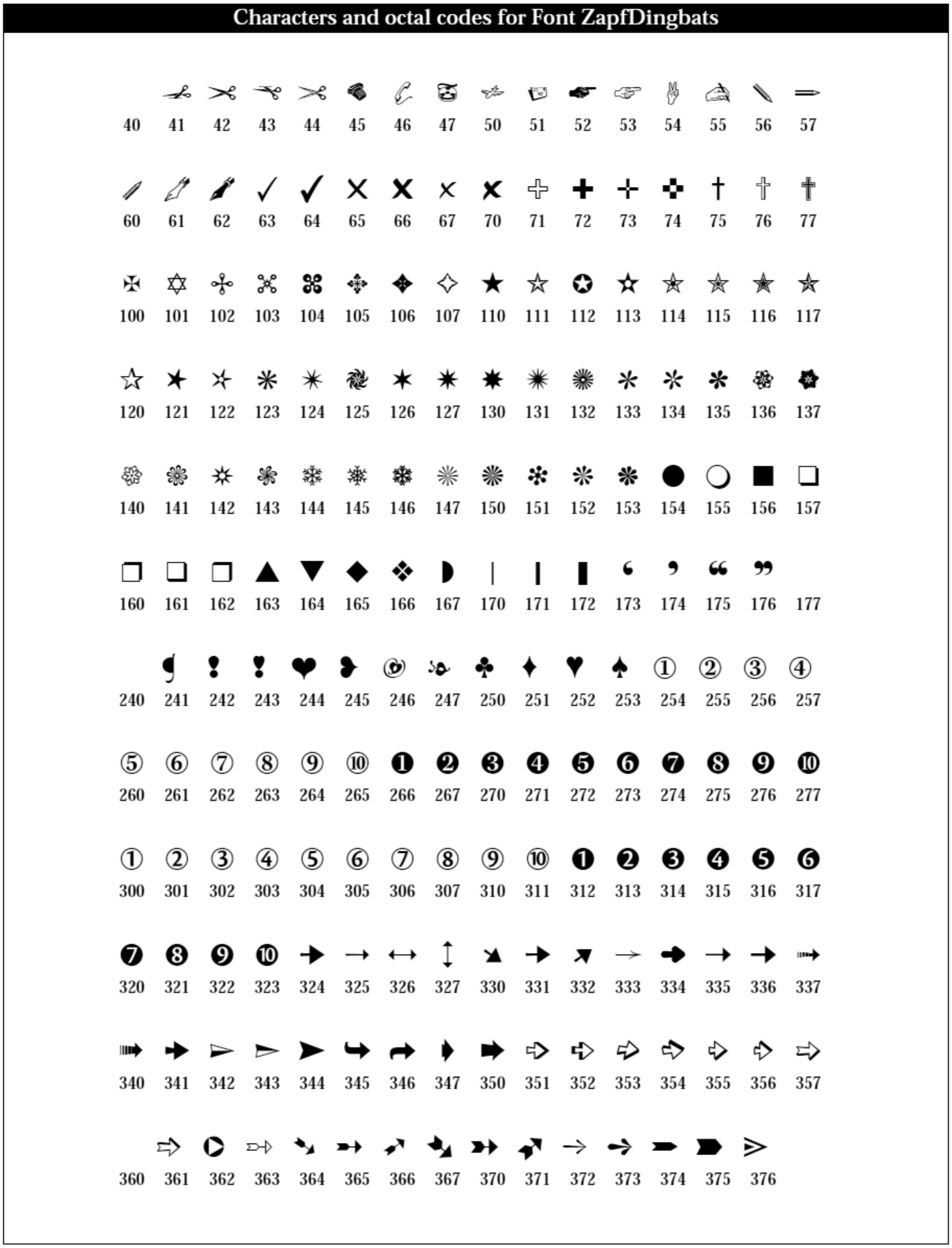

Figure 2: Symbol codes for the file `FileSym`.

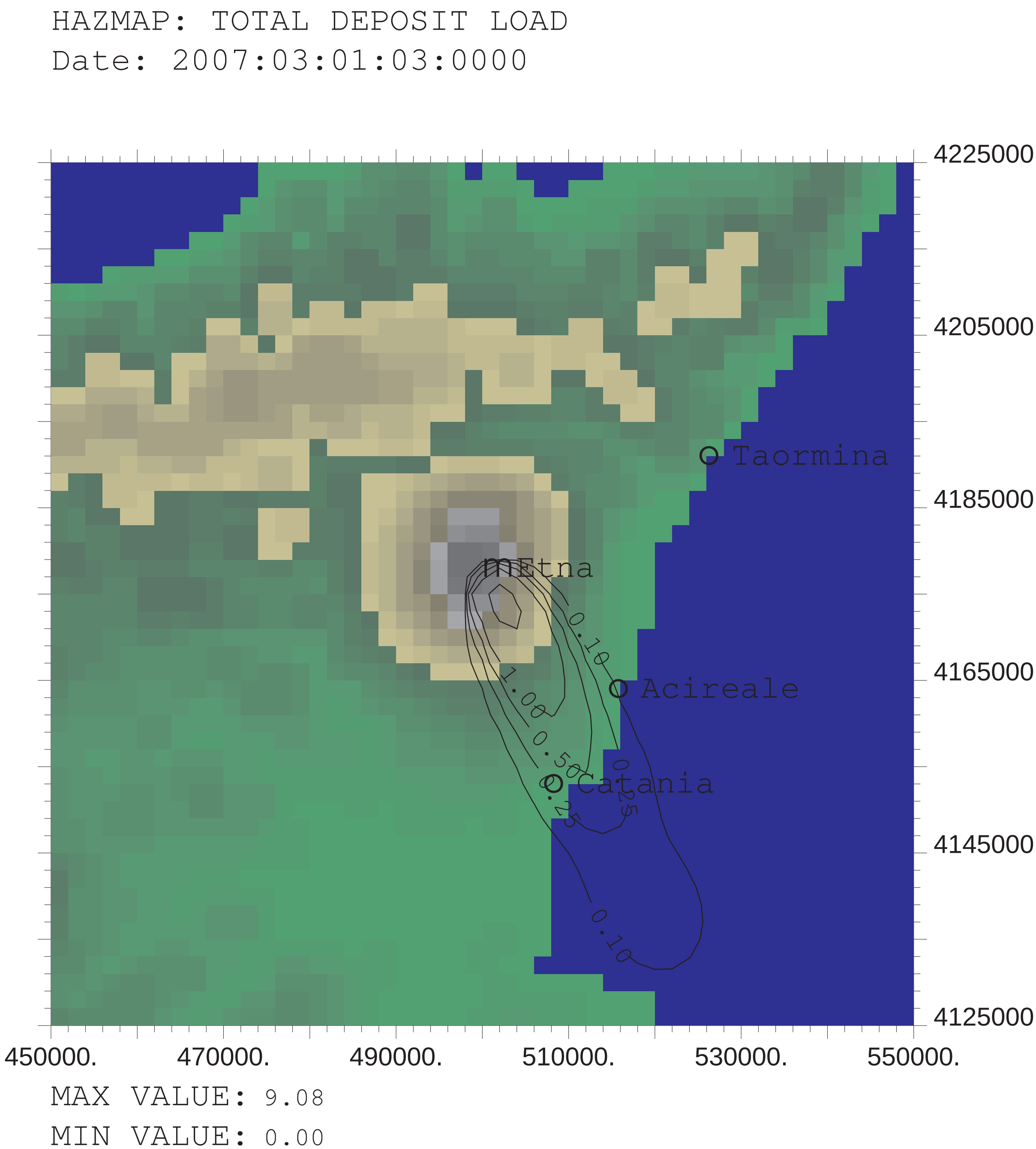

Figure 3: Deposit load (in kg/m^2) at 0300UTC.

7 References

Arastoopour, H., Wang, C., Weil, S., 1982. Particle-particle interaction force in a diluite gas-solid system. Chem. Eng. Sci. 37 (9), 1379–1386.

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

Dellino, P., D. Mele, R. Bonasia, G. Braia, L. La Volpe, R. Sulpizio, 2005. The analysis of the influence of pumice shape on its terminal velocity, Geophys. Res. Lett., 32, L21306.

Ganser, G., 1993. A rational approachto drag prediction spherical and nonspherical particles. Powder Technology 77, 143–152.

Macedonio, G., Costa, A., Longo, A., 2005. A computer model for volcanic ash fallout and assessment of subsequent hazard. Computer and Geosciences 31, 837–845.

Pfeiffer, T., Costa, A., Macedonio, G., 2005. A model for the numerical simulation of tephra fall deposits. J. Volcanol. Geotherm. Res. 140, 273–294.

Wilson, L., and T. C. Huang, 1979. The influence of shape on the atmospheric settling velocity of volcanic ash particles, Earth Planet. Sci. Lett. 44, 311–324.

Suzuki, T., 1983. A theoretical model for dispersion of tephra. In: D. Shimozuru, I. Yokoyama (Eds.), Arc Volcanism: Physics and Tectonics, Terra Scientific Publishing Company (TERRAPUB), Tokyo.