Project INGV-DPC V5 Diffuse degassing in Italy (2005-2007)
Project Coordinator: Dr. Giovanni Chiodini
(INGV Naples - Osservatorio Vesuviano)

# *TWODEE-2*
# *Computer Code and Related Documentation*
(for internal use only)

Arnau Folch[1], Antonio Costa[1] and Robin Hankin[2]

[1] *INGV Naples-Osservatorio Vesuviano, Italy*
[2] *National Oceanography Centre, Southampton, UK*

June 2007

# Contents

**1 Introduction** **3**

**2 Heavy Gas Transport Model** **3**
2.1 Model variables . . . 4
2.2 Model equations . . . 4
2.3 Wind model . . . 5

**3 Program setup** **6**
3.1 Installation . . . 6
3.2 Folder structure . . . 6
3.3 Program run . . . 6

**4 The TWODEE-2 input files** **7**
4.1 The control file `problemname.inp` . . . 7
4.1.1 BLOCK TIME . . . 7
4.1.2 BLOCK GRID . . . 8
4.1.3 BLOCK PROPERTIES . . . 8
4.1.4 BLOCK METEO . . . 8
4.1.5 BLOCK FILES . . . 9
4.1.6 BLOCK OUTPUT . . . 9
4.1.7 BLOCK NUMERIC . . . 10
4.2 The topography file `topography.dat` . . . 12
4.3 The terrain file `terrain.dat` . . . 13
4.4 The restart file `restart.dat` . . . 13
4.5 The source file `source.dat` . . . 13
4.6 The wind data file `wind.dat` . . . 15
4.7 The track points file `points.dat` . . . 16
4.8 The boxes points file `boxes.dat` . . . 16
4.9 The DIAGNO file `diagno.dat` . . . 16

**5 The TWODEE-2 output files** **17**
5.1 The `problemname.log` file . . . 17
5.2 The GRD file format . . . 17
5.3 The CSV file format . . . 17
5.4 Visualization of TWODEE-2 output files . . . 17

**6 References** **18**

**Abstract**

Here we describe the model TWODEE-2 , a shallow layer time-dependent Eulerian model for dispersion of heavy gases. TWODEE-2 is a FORTRAN 90 code which has been derived from the optimization and improvement of a previous FORTRAN 77 code named TWODEE (Hankin and Britter, 1999a,b,c). The model is based on the solution of a shallow water equations system for fluid depth, depth-averaged horizontal velocities and depth-averaged fluid density. The shallow layer approach used by TWODEE-2 is a compromise between the complexity of CFD models and the simpler integral models. The model can be used for forecasting gas dispersion near the ground and/or for hazard assessment over complex terrains. The input to the model are topography, wind measurements from meteorological stations and gas flow rate from the ground sources.

# 1 Introduction

Many volcanic and non-volcanic areas in Italy emit a huge amount of gas into the atmosphere. One of the most frequent gases discharged from both volcanic (*e.g.*, Solfatara Volcano) and non-volcanic sources (*e.g.*, central Italy vents) is the carbon dioxide ($CO_2$) which has a molecular weight greater than air. For this reason, under stable atmospheric conditions and/or in presence of topographic depressions, $CO_2$ concentration can reach high values resulting in lethal effects to humans or animals. In fact, several episodes of this phenomenon were recorded at different areas in central Italy (Rogie et al., 2000) and worldwide. One of the most tragic example was the 1986 degassing of Lake Nyos, Cameroon, when a dense cloud of carbon dioxide hugging the ground suffocated more than 1700 people in one night (Clarke, 2001).

The cloud dispersion of gases denser than air released from natural sources is governed by gravity and by the effects of lateral eddies which decrease the plume density through the incorporation of sourrounding air. In the initial phase the negative buoyancy controls the gas dispersion and the cloud follows the ground (gravitational phase). In this phase, the dispersion of heavy gas is markedly different from a passive or a positively buoyant gas dispersion. In contrast, when the density contrast becomes less important, gas dispersion is mainly governed by wind and atmospheric turbulence, *i.e.*, passive dispersion phase (e.g., Costa et al., 2005).

Although from a theoretical point of view gas dispersion can be fully studied by solving the transport equations for mass, momentum and energy, in practice and because the demanding computational requests, different simplified models which describe only specific phases are commonly used. Such models range from the simplest analytical Gaussian models to the more complex Computational Fluid Dynamics (CFD) models (e.g., Macedonio and Costa, 2002).

A common approach, given by the Box (or Similarity) models, describe the integral properties of plume. A set of differential equations for averaged mass, momentum and energy balance is solved along the plume using different simplifying similarity assumptions (e.g., Blackmore et al., 1982). SLAB (Ermak, 1990), HEGADAS (Witlox, 1994) and DEGADIS (Spicer and Havens, 1989) are popular examples of these similarity models. The most complete but computationally most expensive models are the three-dimensional CFD models based on the transport theory of mass, momentum, energy and species. This approach is able to simulate the dispersion of a heavy gas accounting for obstacles, topographic effects, variation of atmospheric conditions and wind direction, *etc.*

A compromise between the complexity of CFD models and the simpler integral models is given by the shallow layer approach which uses depth-averaged variables to describe the flow behavior (Hankin and Britter, 1999a; Venetsanos et al., 2003). These models are used to describe gravity driven flows of dense gas over complex topography and TWODEE-2 , the model that will be described below, belongs to this category. TWODEE-2 is a FORTRAN 90 code which has been derived from the optimization and improvement of a previous FORTRAN 77 code named TWODEE developed in Hankin and Britter (1999a,b,c).

# 2 Heavy Gas Transport Model

Depth averaged flow models based on the so-called shallow water equations (SWE) were firstly introduced by De Saint Venant in 1864 and Boussinesq in 1872. At present, applications of the shallow water equations cover a wide range of problems which have important implications for hazard assessment, from flood simulation (Burguete et al., 2002) to propagation of tsunamis (Heinrich et al., 2001).

TWODEE-2 is based on depth-averaged equations obtained by integrating mass, density and momentum

equations over the fluid depth, from the bottom up to the free surface. This approach is valid in the limit $H_*^2/L_*^2 \ll 1$ (where $H_*$ is the undisturbed fluid height and $L_*$ the characteristic wave length scale in the flow direction). This means that we are dealing with very long waves or with "shallow water". Such approach is able to describe the cloud as function of time and of the two-dimensional ground positions, in terms of four variables: cloud depth, two depth-averaged horizontal velocities, and depth-averaged cloud concentration. Thermodynamic effects such as condensation are not included at present, but further development could account for them by introducing an additional equation for gas enthalpy.

## 2.1 Model variables

Since real clouds do not have a definite upper surface it is necessary to define cloud depth in terms of the vertical concentration distribution. In fact, we must point out that the actual vertical concentration profile is not uniform as for fluids usually described by shallow water equations, but characterised by an exponential decay (Hankin and Britter, 1999a). Depth averaged values of density and velocities must therefore be defined in terms of their vertical distribution.
In TWODEE-2 , $h$ is that height below which some fraction $\alpha$ of the buoyancy $g(\overline{\rho} - \rho_a)$ is located:

$$\int_{z=0}^{h} (\rho(z) - \rho_a) dz \equiv \alpha \int_{z=0}^{\infty} (\rho(z) - \rho_a) dz \tag{1}$$

where the choice $\alpha = 0.90$ (or 0.95) is adopted. Concerning the depth-averaged density $\overline{\rho}$ we have:

$$h(\overline{\rho} - \rho_a) \equiv \int_{z=0}^{\infty} (\rho(z) - \rho_a) dz \tag{2}$$

In similar way, the depth-averaged velocities $\overline{u}$ and $\overline{v}$ are given by:

$$h(\overline{\rho} - \rho_a)\overline{u} \equiv \int_{z=0}^{\infty} (\rho(z) - \rho_a) u(z) dz \tag{3}$$

$$h(\overline{\rho} - \rho_a)\overline{v} \equiv \int_{z=0}^{\infty} (\rho(z) - \rho_a) v(z) dz \tag{4}$$

In particular, Hankin and Britter (1999a,b) showed that the vertical distribution for density can be calculated from:

$$\rho(z) = \rho_a + \frac{2}{S_1}(\overline{\rho} - \rho_a) \exp\left(-\frac{2}{S_1}\frac{z}{h}\right) \tag{5}$$

where $S_1$ is a shape parameter. Then, the vertical concentration $c$ (in ppm) results:

$$c(z) = c_b + (10^6 - c_b) \times \frac{\rho(z) - \rho_a}{\rho_g - \rho_a} \tag{6}$$

where $c_b$ is the background concentration. Another useful quantity output by the model is the dose $D$, a temporal integrated variable defined as:

$$D(T, z) = \int_0^T [c(z)]^n \frac{dt}{60\text{s}} \tag{7}$$

where $n$ is the toxicity exponent (specified in the control input file), and $T$ the time interval in seconds.

## 2.2 Model equations

Assuming an incompressible homogeneous fluid and an hydrostatic pressure distribution, the shallow water equations for an uniform or gradually varied flow are given by:

$$\frac{\partial h}{\partial t} + \frac{\partial h\overline{u}}{\partial x} + \frac{\partial h\overline{v}}{\partial y} = u_{entr} \tag{8}$$

$$\frac{\partial h(\overline{\rho} - \rho_a)}{\partial t} + \frac{\partial h(\overline{\rho} - \rho_a)\overline{u}}{\partial x} + \frac{\partial h(\overline{\rho} - \rho_a)\overline{v}}{\partial y} = u_{entr}\rho_a \tag{9}$$

$$\frac{\partial h\overline{\rho}\,\overline{u}}{\partial t}+\frac{\partial h\overline{\rho}\,\overline{u}^2}{\partial x}+\frac{\partial h\overline{\rho}\,\overline{u}\,\overline{v}}{\partial y}+\frac{1}{2}S_1\frac{\partial g(\overline{\rho}-\rho_a)h^2}{\partial x}+S_1 g(\overline{\rho}-\rho_a)h\frac{\partial e}{\partial x}+ \\ \frac{1}{2}\overline{\rho}C_D\overline{u}\,|\mathbf{u}|+V_x+k\rho_a\left[\frac{\partial}{\partial t}+u_a\frac{\partial}{\partial x}+v_a\frac{\partial}{\partial y}\right][h(\overline{u}-u_a)]=u_{entr}\rho_a u_a \tag{10}$$

$$\frac{\partial h\overline{\rho}\,\overline{v}}{\partial t}+\frac{\partial h\overline{\rho}\,\overline{v}^2}{\partial y}+\frac{\partial h\overline{\rho}\,\overline{u}\,\overline{v}}{\partial x}+\frac{1}{2}S_1\frac{\partial g(\overline{\rho}-\rho_a)h^2}{\partial y}+S_1 g(\overline{\rho}-\rho_a)h\frac{\partial e}{\partial y}+ \\ \frac{1}{2}\overline{\rho}C_D\overline{v}\,|\mathbf{u}|+V_y+k\rho_a\left[\frac{\partial}{\partial t}+u_a\frac{\partial}{\partial x}+v_a\frac{\partial}{\partial y}\right][h(\overline{v}-v_a)]=u_{entr}\rho_a v_a \tag{11}$$

where the meaning of all variables is described in Table 1 (for a more detailed description of the physical model see Hankin and Britter, 1999a). TWODEE-2 is based on the numerical solution of the governing equations (8) to (11) by using the algorithm described in Hankin and Britter (1999b).

| Symbol | - | Definition |
|---|---|---|
| $t$ | - | Time |
| $(x, y, z)$ | - | Spatial Coordinates |
| $h$ | - | Cloud Depth |
| $(\overline{u}, \overline{v})$ | - | Depth-Averaged Velocities along $(x, y)$ respectively |
| $u_{entr}$ | - | Entrainment Rate of Air |
| $\overline{\rho}$ | - | Depth-Averaged Cloud Density |
| $\rho_a$ | - | Ambient Fluid Density |
| $\rho_g$ | - | Dense Gas Density |
| $c$ | - | Cloud Dense Gas Concentration |
| $c_b$ | - | Dense Gas Background Concentration |
| $D$ | - | Dose |
| $g$ | - | Gravity acceleration |
| $g(\overline{\rho}-\rho_a)$ | - | Cloud Buoyancy |
| $e = e(x, y)$ | - | Terrain Elevation |
| $S_1$ | - | Shape Parameter |
| $k$ | - | Semi-empirical Parameter |
| $C_D$ | - | Skin Friction Coefficient |
| $(V_x, V_y)$ | - | Turbulent Shear Stress exerted on Cloud |
| $(u_a, v_a)$ | - | Ambient Fluid Velocities along $(x, y)$ respectively |
| $U_a$ | - | Wind Velocity Modulus |
| $z_0$ | - | Roughness Length |
| $u_*$ | - | Friction Velocity |
| $L$ | - | Monin-Obukhov Length |
| $K$ | - | Von Karman Constant (0.4) |
| $\psi_m$ | - | Atmospheric Stability Function |
| $Ri_b$ | - | Bulk Richardson Number |

Table 1: Definitions of the symbols used.

## 2.3 Wind model

Concerning wind field description TWODEE-2 admits two options, uniform wind or spatially variable wind which allows to incorporate terrain effects. For the first option, when the wind is considered horizontally uniform, meteorological data at a height $z = z_{ref}$ are directly read from a wind data file (see section 4.6), commonly provided by a ground-based station. For the second option, when the wind is spatially variable, data at height $z = z_{ref}$ are provided by the program DIAGNO (see section 4.9). The program DIAGNO , also provided with the TWODEE-2 package, is a meteorological processor that reads data ("observations") at a point of the domain and, assimilating terrain information, generates a zero-divergence wind field

$(u_x, u_y, u_z)$ in a terrain following coordinate system $x = x'$, $y = y'$, $z = z' - h(x', y')$. The final products of DIAGNO are therefore a null-divergence wind field consistent with the "observations" together with other meteorological parameters like the friction velocity or the Monin-Obukhov length. In both cases, the vertical wind profile is described by the Monin-Obukhov Similarity Theory as:

$$U_a(z) = \frac{u_*}{K}\left[ln\left(\frac{z}{z_0}\right) - \psi_m\left(\frac{z}{L}\right)\right] \tag{12}$$

where $K$ is the von Karman constant, $z_0$ the roughness length, $u_*$ the friction velocity, $L$ the Monin-Obukhov length, and $\psi_m$ denotes the classical stability function for momentum (e.g., Jacobson, 1999). $L$ and $u_*$ are estimated using the non-iterative method of Louis (1979) based on the bulk Richardson number $Ri_b$ (e.g., Jacobson, 1999).

# 3 Program setup

## 3.1 Installation

- On a Windows OS download and decompress the file `twodee2.tar.gz` on your selected directory. The `twodee2.tar` file already contains Windows executables for TWODEE-2 and DIAGNO programs. The untaring of `twodee2.tar` will create the folders described in the Table 2.
- On a Unix/Linux/Mac X operating system:
  1. Untar file `twodee2.tar` issuing the command "`tar xvf twodee2.tar`". This will generate directory `twodee2` (see Table 2)
  2. Compile the program DIAGNO . Enter the directory `twodee2/Programs/Diagno/Sources`, then issue the command "`make`". You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files.
  3. Compile the program TWODEE-2 . Enter the directory `twodee2/Programs/Twodee2/Sources`, then issue the command "`make`". You can edit the Makefile to select your favourite compiler. After compilation you may issue the command "`make clean`" to remove unneeded files.

## 3.2 Folder structure

Table 2 shows the folder structure. The directory `twodee2/Programs` contains the programs DIAGNO and TWODEE-2 with the corresponding source files. The directory `twodee2/Runs` contains the runs, one within each own folder. An example run named "Example" is provided with the installation.

Table 2: Default TWODEE-2 directory structure.

| | | | |
|---|---|---|---|
| twodee2 | Programs | Diagno | Sources |
| | | Twodee2 | Sources |
| | Runs | Example | infiles |
| | | | outfiles |
| | | ..... | infiles |
| | | | outfiles |

## 3.3 Program run

Both TWODEE-2 and DIAGNO can be launched typing, respectivelly,

- "`Twodee2.exe problemname.inp problemname.log`"
- "`Diagno.exe problemname.inp problemname.log`"

where `problemname.inp` and `problemname.log` are the names (including the paths) of the control input file and of the log file (see sections 4.1 and 5.1). Both filenames are passed as a program call argument. However, it is highly recomended to launch the programs through the script files distributed in the tar folder.

- On a Windows OS enter the folder `twodee2/Runs/Example` and launch the script `Example.win`.
- On a Mac X/Unix/Linux OS enter the folder `twodee2/Runs/Example` and launch the script `Example.unix`.

NOTE: To create a new run simply create a new folder (with subfolders infiles and outfiles), copy the input files and the script into it and modifiy the script line which defines the `problemname`.

# 4 The TWODEE-2 input files

TWODEE-2 needs of the following input files:

- File: `problemname.inp` - Control file that defines a run. Mandatory.
- File: `topography.dat` - Regional ground elevation file. Optional.
- File: `terrain.dat` - Regional roughness height file. Mandatory.
- File: `restart.dat` - Restart (initial conditions) file. Optional.
- File: `source.dat` - Source term (dense gas fluxes) file. Mandatory.
- File: `winds.dat` - Meteorological data file. Mandatory.
- File: `diagno.dat` - DIAGNO meteorological data file. Optional.
- File: `points.dat` - File that defines the tracking points. Optional.
- File: `boxes.dat` - File that defines the tracking boxes. Optional.

NOTE: File names used in this manual are given just for illustrative purposes. Names and paths of input files are absolutely free and can be defined by the user in the control file `problemname.inp`.

## 4.1 The control file `problemname.inp`

The TWODEE-2 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 3 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.*,: $12e7 = 12 \times 10^7$).

### 4.1.1 BLOCK TIME

- `YEAR`: Initial year.
- `MONTH`: Initial month (1-12).
- `DAY`: Initial day (1-31).
- `HOUR`: Initial hour (0-23).
- `MINUTE`: Initial minute (0-59).
- `SIMULATION_INTERVAL_(SEC)`: Simulation time duration (in s).
- `RESTART_RUN`: Flag indicating whether the run is a restart or not. Possibilities are `YES`/`NO`. If `YES`, the run starts from the restart file defined in the FILES BLOCK. If `NO`, the run starts assuming zero dense gas concentration everywhere (*i.e.*, $\rho = \rho_a$, $h = 0$ and $u = v = 0$).

NOTE: The parameters YEAR, MONTH, DAY, HOUR, and MINUTE are used to check the consistence with both the restart and the meteorological files.

### 4.1.2 BLOCK GRID

- `NX`: Number of grid cells along the $x$-direction.
- `NY`: Number of grid cells along the $y$-direction.
- `DX_(M)`: Grid spacing along the $x$-direction (in m).
- `DY_(M)`: Grid spacing along the $y$-direction (in m).
- `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).
- `EXTRACT_TOPOGRAPHY_FROM_FILE`: Flag indicating whether the topography of the computational domain is extracted from a regional ground elevation file or not. Possibilities are `YES`/`NO`. If `YES`, topography is extracted from the regional elevation file defined in the FILES BLOCK. If `NO`, a constant-slope terrain defined by the parameters below is assumed.
- `Z_ORIGIN_(M)`: Elevation (in m) of the grid origin (bottom left corner). This record is read only if `EXTRACT_TOPOGRAPHY_FROM_FILE=NO`.
- `X_SLOPE_(DEG)`: Topography slope (in deg) along the $x$-direction. This record is read only if `EXTRACT_TOPOGRAPHY_FROM_FILE=NO`.
- `Y_SLOPE_(DEG)`: Topography slope (in deg) along the $y$-direction. This record is read only if `EXTRACT_TOPOGRAPHY_FROM_FILE=NO`. Note that, in particular, `X_SLOPE_(DEG)= 0.0` and `Y_SLOPE_(DEG)= 0.0` define a flat terrain.

### 4.1.3 BLOCK PROPERTIES

- `AMBIENT_GAS_DENSITY_20C_(KG/M3)`: Density of the ambient gas (in kg/m$^3$) at 20$^o$C (293$^oK$). A value of 1.204 kg/m$^3$ is used as reference for air.
- `DENSE_GAS_DENSITY_20C_(KG/M3)`: Density of the dense gas (in kg/m$^3$) at 20$^o$C (293 K). A value of 1.839 kg/m$^3$ is used as reference for $CO_2$.
- `AVERAGED_TEMPERATURE_(C)`: Time-averaged temperature (in $^o$C). This is used to estimate the time-averaged densities of air and dense gas according to the perfect gas law: $\rho(T) = \rho(293\text{K}) \times 293\text{K}/T$.
- `DOSE_GAS_TOXIC_EXPONENT`: Exponent for the dose calculation, see eq. (7).

### 4.1.4 BLOCK METEO

- `WIND_MODEL`: Flag indicating the wind model. Possibilities are `UNIFORM` or `DIAGNO`. If `UNIFORM`, meteorological data (constant in space) are read from a wind file defined in the FILES BLOCK. If `DIAGNO`, meteorological data (variable in both space and time) is read from a DIAGNO output file defined in the FILES BLOCK. Note that in the later case the program DIAGNO must necessarily run before the program TWODEE-2 and parameters read from the file `wind.dat` are used as input for DIAGNO .
- `X_STATION_(UTM_M)`: $x$-coordinate of the ground station (UTM coordinates in m). This record is read only by DIAGNO (*i.e.*, used only when `WIND_MODEL = DIAGNO`).
- `Y_STATION_(UTM_M)`: $y$-coordinate of the ground station (UTM coordinates in m). This record is read only by DIAGNO (*i.e.*, used only when `WIND_MODEL = DIAGNO`).
- `Z_REFERENCE_(M)`: Reference height $z_{ref}$ (in m) for temperature. This is the height at which meteorological parameters have been measured.

### 4.1.5 BLOCK FILES

- TOPOGRAPHY_FILE_PATH: Name (including relative or absolute path) of the regional topography file. See section 4.2 for file format details. This record is read only when EXTRACT_TOPOGRAPHY_FROM_FILE = YES.
- ROUGHNESS_FILE_PATH: Name (including relative or absolute path) of the regional terrain file. See section 4.3 for file format details.
- RESTART_FILE_PATH: Name (including relative or absolute path) of the restart file. See section 4.4 for file format details. During a run this file is updated with the current values for variables every OUTPUT_INTERVAL_(SEC) seconds.
- SOURCE_FILE_PATH: Name (including relative or absolute path) of the source file. See section 4.5 for file format details.
- WIND_FILE_PATH: Name (including relative or absolute path) of the wind data file. See section 4.6 for file format details.
- DIAGNO_FILE_PATH: Name (including relative or absolute path) of the DIAGNO output file. This is an unformatted file created by DIAGNO . Only used when WIND_MODEL = DIAGNO.
- TRACK_POINTS_FILE_PATH: Name (including relative or absolute path) of the file that defines the coordinates of the points to be tracked during postprocess. Only used when TRACK_POINTS= YES. See section 4.7 for file format details.
- BOXES_POINTS_FILE_PATH: Name (including relative or absolute path) of the file that defines points and areas around them (boxes) where area-averaged concentration is to be tracked during post-process. Only used when TRACK_BOXES= YES. See section 4.8 for file format details.
- OUTPUT_DIRECTORY: Name (including relative or absolute path) of the folder where TWODEE-2 output files are dumped.

### 4.1.6 BLOCK OUTPUT

- OUTPUT_INTERVAL_(SEC): Time interval to output results (in s).
- OUTPUT_DOMAIN: Flag indicating whether the file containing the topography of the computational domain has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file named topog.grd and containing the topography at the computational domain is dumped in the folder defined by the OUTPUT_DIRECTORY record.
- OUTPUT_SOURCE: Flag indicating whether the source term has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file named source.grd and containing the source term (upward velocity of dense gas in m/s) is dumped in the folder defined by the OUTPUT_DIRECTORY record.
- OUTPUT_U_VELOCITY: Flag indicating whether the cloud velocity $u$ has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file is dumped in the folder defined by the OUTPUT_DIRECTORY record every OUTPUT_INTERVAL_(SEC) seconds.
- OUTPUT_V_VELOCITY: Same than the previous record but for the $v$ cloud velocity component.
- OUTPUT_H: Flag indicatind whether the cloud height $h$ has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file is dumped at the OUTPUT_DIRECTORY every OUTPUT_INTERVAL_(SEC).
- OUTPUT_RHO: Flag indicating whether the averaged cloud density $\overline{\rho}$ has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file is dumped in the folder defined by the OUTPUT_DIRECTORY record every OUTPUT_INTERVAL_(SEC) seconds.
- OUTPUT_DOSE: Flag indicating whether the dose has to be printed or not. Possibilities are YES or NO. If YES, a GRD-format file is dumped in the folder defined by the OUTPUT_DIRECTORY record every OUTPUT_INTERVAL_(SEC) seconds.

- `OUTPUT_CONCENTRATION`: Flag indicating whether the dense gas concentration (in ppm) has to be printed at different user-specified heights or not. Possibilities are `YES` or `NO`. If `YES`, a GRD-format file for each height is dumped in the folder defined by the `OUTPUT_DIRECTORY` record every `OUTPUT_INTERVAL_(SEC)` seconds.
- `CONCENTRATION_BG` : Dense gas background concentration (in ppm). This record is used only when `OUTPUT_CONCENTRATION= YES`.
- `HEIGHTS_(M)`: List of heights (in m) at which dense gas concentration is calculated. This record is used only when `OUTPUT_CONCENTRATION= YES`.
- `OUTPUT_Z_CRITICAL`: Flag indicating whether the height at which a certain critical concentration value is achived has to be printed or not. Possibilities are `YES` or `NO`. If `YES`, a GRD-format file for each critical concentration is dumped in the folder defined by the `OUTPUT_DIRECTORY` record every `OUTPUT_INTERVAL_(SEC)` seconds.
- `CRITICAL_C_(%)`: List of critical concentration values (in %). This record is used only when `OUTPUT_Z_CRITICAL= YES`.
- `TRACK_POINTS`: Flag indicating whether concentration has to be calculates at the different points, defined in the file `points.dat`, or not. Possibilities are `YES` or `NO`. If `YES`, a CSV-format file containing point-concentration every minute is dumped at the floder defined by the `OUTPUT_DIRECTORY` record.
- `TRACK_BOXES`: Flag indicating whether concentration has to be printed at different boxes (rectangular regions), defined in the file `boxes.dat`, or not. Possibilities are `YES` or `NO`. If `YES`, a CSV-format file containing box-averaged concentration every minute is dumped at the folder defined by the `OUTPUT_DIRECTORY` record .

### 4.1.7 BLOCK NUMERIC

- `FRONT_FROUDE_NUMBER`: Front Froude number (usually equal to 1).
- `OPTIMAL_COURANT_NUMBER`: Critical Courant number (usually equal to 0.25).
- `EDGE_ENTRAINMENT_COEFF`: Edge entrainment coefficient. (currently equal to 0.0).
- `DIFFUSION_COEFFICIENT`: Numerical parameter in the flux scheme (usually set to 0.2).
- `SHAPE_PARAMETER`: Shape parameter (usually set to 0.5).
- `ZETA_PARAMETER`: Zeta constant in turbulent shear stress (currently equal to 0.0).
- `ALPHA_2`: Entrainment coefficient (usually set to 0.7).
- `ALPHA_3`: Entrainment coefficient (usually set to 1.3).
- `ALPHA_7`: Entrainment coefficient (usually set to 0.45).
- `VON_KARMAN_CONSTANT`: Von Karman constant (usually set to 0.4).
- `BRITTER_B_CONSTANT`: Britter constant (usually set to 0.11).

NOTE: By default numerical parameters are set to their optimal values and typically there is no need to change them in TWODEE-2 applications.

Table 3: Sample of the input control file `problemname.inp`.

```
TIME
YEAR = 2007
MONTH = 02
DAY = 05
HOUR = 17
MINUTE = 40
SIMULATION_INTERVAL_(SEC) = 54000
RESTART_RUN = NO

GRID
NX = 200
NY = 200
DX_(M) = 5.
DY_(M) = 5.
X_ORIGIN_(UTM_M) = 259567.
Y_ORIGIN_(UTM_M) = 4663175.
EXTRACT_TOPOGRAPHY_FROM_FILE = YES
Z_ORIGIN_(M) = 200.
X_SLOPE_(DEG) = 0.
Y_SLOPE_(DEG) = 0.

PROPERTIES
AMBIENT_GAS_DENSITY_20C_(KG/M3) = 1.204
DENSE_GAS_DENSITY_20C_(KG/M3) = 1.839
AVERAGED_TEMPERATURE_(C) = 0.0
DOSE_GAS_TOXIC_EXPONENT = 2.0

METEO
WIND_MODEL = UNIFORM
X_STATION_(UTM_M) = 500.
Y_STATION_(UTM_M) = 500.
Z_REFERENCE_(M) = 3.45

FILES
TOPOGRAPHY_FILE_PATH = .\infiles\topography.dat
ROUGHNESS_FILE_PATH = .\infiles\terrain.dat
RESTART_FILE_PATH = .\infiles\restart.dat
SOURCE_FILE_PATH = .\infiles\source.dat
WIND_FILE_PATH = .\infiles\winds.dat
DIAGNO_FILE_PATH = .\infiles\diagno.res
TRACK_POINTS_FILE_PATH = .\infiles\points.dat
BOXES_POINTS_FILE_PATH = .\infiles\boxes.dat
OUTPUT_DIRECTORY = .\outfiles\
```

continue →

```
← continued
OUTPUT
OUTPUT_INTERVAL_(SEC) = 7200
OUTPUT_DOMAIN = YES
OUTPUT_SOURCE = YES
OUTPUT_U_VELOCITY = YES
OUTPUT_V_VELOCITY = YES
OUTPUT_H = YES
OUTPUT_RHO = YES
OUTPUT_DOSE = YES
OUTPUT_RHOMAX = YES
OUTPUT_CONCENTRATION = YES
CONCENTRATION_BG = 350.
HEIGHTS_(M) = 1.0 2.0 3.0
OUTPUT_Z_CRITICAL = yes
CRITICAL_C_(%) = 1 5 10
TRACK_POINTS = YES
TRACK_BOXES = YES

NUMERIC
FRONT_FROUDE_NUMBER = 1.0
OPTIMAL_COURANT_NUMBER = 0.25
EDGE_ENTRAINMENT_COEFF = 0.0
DIFFUSION_COEFFICIENT = 0.20
SHAPE_PARAMETER = 0.5
ZETA_PARAMETER = 0.0
ALPHA_2 = 0.7
ALPHA_3 = 1.3
ALPHA_7 = 0.45
VON_KARMAN_CONSTANT = 0.4
BRITTER_B_CONSTANT = 0.11
```

## 4.2 The topography file `topography.dat`

The topography file specifies ground elevation at a regional scale (*i.e.*, in a region typically larger than the computational domain). Topography must be specified on a structurated grid using arbitrary (but constant) grid spacing (*e.g.,* 5 m, 10 m, 100 m, *etc*). Discretizations along $x-$ and $y$-directions can be different. The only necessary requirement is that the computational domain must lay within the bounds of the region where topography is specified. TWODEE-2 reads the topography file and automatically interpolates elevations onto the nodes of the computational grid. The file format is described in Table 4 and the meaning of the used symbols is the following:

- `NTX` : Number of ground elevation points along $x$-direction.
- `NTY` : Number of ground elevation points along $y$-direction.
- `XTO` : x-coordinate (UTM in m) of the bottom left corner point.
- `XTF` : x-coordinate (UTM in m) of the top right corner point.
- `YTO` : y-coordinate (UTM in m) of the bottom left corner point.
- `YTF` : y-coordinate (UTM in m) of the top right corner point.
- `ZT` : Elevation (in m) of each grid point of the regional grid. It consists of an array of `NTX`×`NTY` values stored starting from the bottom-left corner and moving towards right then towards the top of the domain.

| | | | |
|---|---|---|---|
| NTX | NTY | | |
| XTO | XTF | | |
| YTO | YTF | | |
| ZT(i,1) | ... | ... | i=1:NTX |
| ... | ... | ... | |
| ZT(i,j) | ... | ... | i=1:NTX |
| ... | ... | ... | |
| ZT(i,NTY) | ... | ... | i=1:NTX |

Table 4: Format of the topography file `topography.dat`.

### 4.3 The terrain file `terrain.dat`

The terrain file specifies the roughness height $z_0$ at a regional scale. The file format is analogous to that of the regional topography file (Table 4). However, these two files can cover different regions and/or have different spatial discretizations. Instead of the elevation, the terrain file specifies the terrain roughness height $z_0$ (in m), a parameter related to the vertical wind profile. Values for $z_0$ range from about $10^{-5}$ m over an iced surface, 0.005 m over naked soil, 0.05 m over soil covered by tall grass, or up to 1 m or more over forest or urban areas.

### 4.4 The restart file `restart.dat`

The restart file can be used to start a new run from the end of a previous simulation. It is automatically created each time TWODEE-2 prints the results, *i.e.,* at every `OUTPUT_INTERVAL_(SEC)`, or at the end of a run. Any restart file previously created is destroyed whenever a new restart file is printed. The file format is described in Table 5 and the meaning of the used symbols is the following:

- `tstart`: Simulation time (in s) at which variables are stored. When a simulation starts from a restart file time is automatically advanced to `tstart`, that is, the simulation begins at `t=tstart` (and not at `t=0` as in a no restart run) and ends at `t= SIMULATION_INTERVAL_(SEC)`.
- `NX` : Number of grid points along $x$-direction.
- `NY` : Number of grid points along $y$-direction.
- `DX` : grid spacing along $x$-direction (in m).
- `DY` : grid spacing along $y$-direction (in m).
- `XO` : x-coordinate (UTM in m) of the bottom left corner.
- `YO` : y-coordinate (UTM in m) of the bottom left corner.
- `h u v rho` : Arrays of variables stored starting from the bottom-left corner and moving towards right then towards the top of the domain.

### 4.5 The source file `source.dat`

The source file specifies dense gas fluxes (in mass flow rate or mass flow rate per unit area) from different rectangular areas or point sources. TWODEE-2 reads this file and automatically calculates the upward source velocity and to interpolates the mass flow rate onto the nodes of the computational domain. Interpolation is done ensuring mass conservation. The advantadge of this approach is that the source file becomes independent of the computational mesh (*i.e.*, the source file is created only once and is the same regardless the location and/or the spatial resolution of the computational grid). The file format is described in Table 6 and the meaning of the used symbols is the following:

- `X_s` : $x$-coordinate (UTM in m) of the source (center of the area for extended sources).
- `Y_s` : $y$-coordinate (UTM in m) of the source (center of the area for extended sources).

| comments | | | | | | | (7 lines) |
|---|---|---|---|---|---|---|---|
| tstart | NX | NY | DX | DY | XO | YO | |
| h(i,1) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| h(i,j) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| h(i,NY) | ... | ... | ... | ... | ... | ... | i=1:NX |
| u(i,1) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| u(i,j) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| u(i,NY) | ... | ... | ... | ... | ... | ... | i=1:NX |
| v(i,1) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| v(i,j) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| v(i,NY) | ... | ... | ... | ... | ... | ... | i=1:NX |
| rho(i,1) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| rho(i,j) | ... | ... | ... | ... | ... | ... | i=1:NX |
| ... | ... | ... | ... | ... | ... | ... | |
| rho(i,NY) | ... | ... | ... | ... | ... | ... | i=1:NX |

Table 5: Format of the restart file `restart.dat`.

- `PHI_s`: Flux associated to the area or to the point source. Units are defined in the flag UNITS.
- `DX_s` : `s`-source extension (in m) along the $x$-direction.
- `DY_s` : `s`-source extension (in m) along the $y$-direction.
- `UNITS` : Flux units. Character flag. Possible values are:
  - KG_M2_SEC ($\mathrm{kg\,m^{-2}s^{-1}}$), GR_M2_SEC ($\mathrm{g\,m^{-2}s^{-1}}$), TN_M2_DAY ($\mathrm{ton\,m^{-2}day^{-1}}$), KG_M2_DAY ($\mathrm{kg\ m^{-2}\,day^{-1}}$), GR_M2_DAY ($\mathrm{g\ m^{-2}\,day^{-1}}$) when the flux is given as a mass flow rate per unit area. In this case the user has to specify the area around the point source coordinate where the flux can be assumed uniform. This is common for diffuse degassing sources.
  - KG_SEC ($\mathrm{kg\,s^{-1}}$), GR_SEC ($\mathrm{g\,s^{-1}}$), TN_DAY ($\mathrm{ton\,day^{-1}}$), KG_DAY ($\mathrm{kg\,day^{-1}}$), GR_DAY ($\mathrm{g\,day^{-1}}$) when the total flux is given as a mass flow rate. In this case the user has to specify the area (usually smaller than the computational grid) from which the gas is expelled. This is common for punctual sources where the emission area is smaller than computational grid size. In this case, it is possible also to specify directly the upward gas velocities on all the points specified by the user setting the label M_S ($\mathrm{m\,s^{-1}}$). In this last case the area is not used and the user has to set `DX_s=0` and `DY_s=0`.

| X_1 | Y_1 | PHI_1 | DX_1 | DY_1 | UNITS_1 |
|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... |
| X_s | Y_s | PHI_s | DX_s | DY_s | UNITS_s |
| ... | ... | ... | ... | ... | ... |
| X_ns | Y_ns | PHI_ns | DX_ns | DY_ns | UNITS_ns |

Table 6: Format of the source file `source.dat`.

## 4.6 The wind data file `wind.dat`

The wind data file contains meteorological data at different time slices. If the record `WIND_MODEL`, in the control input file, is `UNIFORM`, TWODEE-2 reads this file and estimates the Atmospheric Surface Layer parameters. Otherwise, if the record `WIND_MODEL` is `DIAGNO`, this file is instead read by DIAGNO which uses values as input and TWODEE-2 simply uses the DIAGNO output (file `diagno.dat`). The file format is described in Table 7 (if `code=CUP`) and Table 8 (if `code=SONIC`) and the meaning of the used symbols is the following:

- `iyr` : Measurement year.
- `imo` : Measurement month (1-12).
- `idy` : Measurement day (1-31).
- `ihr` : Measurement hour (0-23).
- `imi` : Measurement minute (0-59).
- `code` : Flag that indicates the type of anemometer. Possibilities are CUP or SONIC.
- `t_1` : Time slice starting time (in s after initial time). Initial time for measurements is assumed to be at day `idy`, hour `ihr`, and minute `imi`.
- `t_2` : Time slice ending time (in s after initial time). Initial time for measurements is assumed to be at day `idy`, hour `ihr`, and minute `imi`. A data time slice spans during the interval [`t_1`,`t_2`].
- `wx` : $x$-component of wind speed (in m/s).
- `wy` : $y$-component of wind speed (in m/s).
- `T_z0` : Temperature (in $^oC$) at the ground level.
- `T_zref` : Temperature (in $^o$C) at $z_{ref}$ (reference height defined at the `Z_REFERENCE_(M)` record of the control input file).
- `p` : Atmospheric pressure (in HPa).
- `ustar` : Friction velocity (in m/s).
- `L` : Monin-Obukhov lenght (in m).

NOTE: The records `iyr` to `imi` are used just to check consistency of meteorological data files with the input control file. Initial time must concide in both cases.

| iyr | imo | idy | ihr | imi | code | | (if code=CUP) |
|---|---|---|---|---|---|---|---|
| t_1 | t_2 | wx | wy | T_z0 | T_zref | p | |
| ... | ... | ... | ... | .... | ... | ... | |
| t_(nt-1) | t_nt | wx | wy | T_z0 | T_zref | p | |

Table 7: Format of the wind file `wind.dat` if `code=CUP`.

| iyr | imo | idy | ihr | imi | code | | (if code=SONIC) |
|---|---|---|---|---|---|---|---|
| t_1 | t_2 | wx | wy | T_zref | ustar | L | |
| ... | ... | ... | ... | .... | ... | ... | |
| t_(nt-1) | t_nt | wx | wy | T_zref | ustar | L | |

Table 8: Format of the wind file `wind.dat` if `code=SONIC`.

## 4.7 The track points file `points.dat`

This file defines the coordinates of the tracked points (points where time evolution of concentration is output). There is no limit on the number of points. The file format is described in Table 9 and the meaning of the used symbols is the following:

- `x_p` : point $x$-coordinate (UTM in m).
- `y_p` : point $y$-coordinate (UTM in m).
- `z_p` : point $z$-coordinate (in m). This is the elevation at which concentration is calculated according to eq. (6).

| | | |
|---|---|---|
| x_1 | y_1 | z_1 |
| ... | ... | ... |
| x_p | y_p | z_p |
| ... | ... | ... |
| x_np | y_np | z_np |

Table 9: Format of the file `point.dat`.

## 4.8 The boxes points file `boxes.dat`

This file defines the coordinates of the tracked boxes (areas where evolution of averaged concentration is output). There is no limit on the number of boxes. The file format is described in Table 10 and the meaning of the used symbols is the following:

- `x_b` : $x$-coordinate of the box center (UTM in m).
- `y_b` : $y$-coordinate of the box center (UTM in m).
- `z_b` : Box z-coordinate (in m). This is the elevation at which concentration is calculated according to eq. (6).
- `DX_b` : Box dimension (in m) along $x$.
- `DY_b` : Box dimension (in m) along $y$.

| | | | | |
|---|---|---|---|---|
| x_1 | y_1 | z_1 | DX_1 | DY_1 |
| ... | ... | ... | | |
| x_b | y_b | z_b | DX_b | DY_b |
| ... | ... | ... | | |
| x_nb | y_nb | z_nb | DX_nb | DY_nb |

Table 10: Format of the file `boxes.dat`.

## 4.9 The DIAGNO file `diagno.dat`

The wind field at the user defined reference height $z_{ref}$ produced by DIAGNO is stored in the unformatted file `diagno.dat`.

# 5 The TWODEE-2 output files

At each user-specified time (see `OUTPUT_INTERVAL_(SEC)` record in the control file) TWODEE-2 can generate 2D contour-files written in GRD-format for the following variables: $h$, $u$, $v$, $\rho$, $c$, and $D$. In addition, in can also output CSV-format files with concentration at defined points and/or boxes every minute. All output files are stored in the directory defined by the `OUTPUT_DIRECTORY` record of the TWODEE-2 control file.

## 5.1 The `problemname.log` file

This file, passed as a program call argument, contains information concerning the run (summary of input data, run time error messages, CPU time, *etc.*). It also outputs some basic indicators of the cloud evolution every minute.

## 5.2 The GRD file format

The structure of a GRD format file is described in Table 11 and the meaning of the used symbols is the following:

- `NX` : Number of grid points along $x$-direction.
- `NY` : Number of grid points along $y$-direction.
- `XO` : $x$-coordinate (UTM in m) of the grid bottom left corner.
- `XF` : $x$-coordinate (UTM in m) of the grid top right corner point.
- `YO` : $y$-coordinate (UTM in m) of the grid bottom left corner point.
- `YF` : $y$-coordinate (UTM in m) of the grid top right corner point.
- `VAL` : Value at each grid point. It consists of an array of `NX`×`NY` values stored starting from the bottom-left corner and moving towards right then up towards the top of the domain.

| | | | |
|---|---|---|---|
| NX | NY | | |
| XO | XF | | |
| YO | YF | | |
| MAX(v) | MIN(v) | | |
| VAL(i,1) | ... | ... | i=1:NX |
| ... | ... | ... | |
| VAL(i,j) | ... | ... | i=1:NX |
| ... | ... | ... | |
| VAL(i,NY) | ... | ... | i=1:NX |

Table 11: Format of a GRD file `file.grd`.

## 5.3 The CSV file format

The CSV ("comma separated variables") is a free ASCII format in which variables are stored in columns separated by commas.

## 5.4 Visualization of TWODEE-2 output files

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 ). Files in CSV format can be read directly by Excel© or by any text editor.

## Acknowledgments

This work was supported by the Department of Civil Protection of Italy.

## 6 References

Blackmore, D., Herman, M., Woodward, J., 1982. Heavy gas dispersion models. J. Hazard. Mater. 6, 107–128.

Burguete, J., Garcia-Navarro, P., Aliod, R., 2002. Numerical simulation of runoff from extreme rainfall events in a mountain water catchment. Nat. Haz. Earth Syst. Sci. 2, 109–117.

Clarke, T., 2001. Taming Africa's killer lake. Nature 409, 554–555.

Costa, A., Chiodini, G., Granieri, D., Folch, A., Hankin, R., Caliro, S., Avino, R., Cardellini, C., 2007. A shallow layer model for heavy gas dispersion from natural sources: application and hazard assessment at Caldara di Manziana, italy. Earth Planet. Sci. Lett. Submitted.

Costa, A., Macedonio, G., Chiodini, G., 2005. Numerical model of gas dispersion emitted from volcanic sources. Annals of Geophyisics 48, 805–815.

Ermak, D., 1990. User's manuals for SLAB: An atmospheric dispersion model for denser-then-air releases. Tech. rep., UCRL-MA-105607, Lawrence Livermore National Laboratory, CA.

Hankin, R., Britter, R., 1999a. TWODEE: the Health and Safety Laboratory's shallow layer model for heavy gas dispersion. Part 1. Mathematical basis and physical assumptions. J. Hazard. Mater. A66, 211–226.

Hankin, R., Britter, R., 1999b. TWODEE: the Health and Safety Laboratory's shallow layer model for heavy gas dispersion. Part 2. Outline and validation of the computational scheme. J. Hazard. Mater. A66, 227–237.

Hankin, R., Britter, R., 1999c. TWODEE: the Health and Safety Laboratory's shallow layer model for heavy gas dispersion. Part 3. Experimental validation (Thorney island). J. Hazard. Mater. A66, 237–261.

Heinrich, P., Piatanesi, A., Hébert, H., 2001. Numerical modelling of tsunami generation and propagation from submarine slumps: the 1998 Papua New Guinea event. Geophys. J. Int. 145, 97–111.

Jacobson, M., 1999. Fundamentals of atmospheric modelling, 1st Edition. Cambridge University Press, New York.

Louis, J., 1979. A parametric model of vertical eddy fluxes in the atmosphere. Boundary Layer Meteor. 17, 187–202.

Macedonio, G., Costa, A., 2002. Finite Element modeling of gas dispersion in the atmosphere. In: Buccianti, A., Marini, L., Ottonello, G., Vaselli, O. (Eds.), Proceedings of the Arezzo Seminar in Fluids Geochemistry. Pacini Editore, Ospedaletto (Pisa) - Italy, pp. 147–159.

Rogie, J., Kerrick, D., Chiodini, G., Frondini, F., 2000. Flux measurements of nonvolcanic $CO_2$ emission from some vents in central Italy. J. Geophys. Res. 105 (B4), 8435–8445.

Spicer, T., Havens, J., 1989. DEGADIS - DEnse GAs DISpersion Model. Tech. rep., EPA-450/4-89-0019, San Rafael, CA.

Venetsanos, A., Bartzis, J., Wurtz, J., Papailiou, D., 2003. DISPLAY-2: a two-dimensional shallow layer model for dense gas dispersion including complex features. J. Hazard. Mater. A99, 111–144.

Witlox, H., 1994. The hegadas model for ground-level heavy-gas dispersion - i. steady-state model. Atmos. Environ. 28 (18), 2917–2932.