Project INGV-DPC V5 Diffuse degassing in Italy (2005-2007)
Project Coordinator: Dr. Giovanni Chiodini
(Istituto Nazionale di Geofisica e Vulcanologia - Sezione Bologna)

TWODEE-2.6
Computer Code and Related Documentation

Arnau Folch[1], Antonio Costa[2] and Robin Hankin[3]

[1] *Barcelona Supercomputing Centre, Jordi Girona 31, Barcelona, Spain*
[2] *Istituto Nazionale di Geofisica e Vulcanologia, Sezione Bologna, Via Donato Creti 12, Bologna, Italy*
[3] *National Oceanography Centre, Southampton, UK*

November 2021

License

How to cite TWODEE-2

Folch, A., Costa, A. and Hankin, R.K.S. (2009). Twodee-2: A shallow layer model for dense gas dispersion on complex topography, Computers & Geosciences, 35(3), 667-674, doi:10.1016/j.cageo.2007.12.017.

Contents

List of Tables

Abstract

Here we describe the model TWODEE-2.6 a shallow layer time-dependent Eulerian model for dispersion of heavy gases. TWODEE-2.6 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, 1999b,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 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 and Woodward, 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, the model that will be described below, belongs to this category. TWODEE 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); Folch et al. (2009).

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 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, h is that height below which some fraction α 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:

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

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

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

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

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 is based on the numerical solution of the governing equations (8) to (11) by using the algorithm described in Hankin and Britter (1999b).

2.3 Wind model

Concerning wind field description TWODEE 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 4.5), 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 4.8). Program DIAGNO 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 ψ_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).
Please, note that starting from TWODEE version 2.6, the program DIAGNO is distributed separately from the TWODEE package. This version of TWODEE is compatible with `diagno-1.1.6`.

Table 1: Definitions of the symbols used

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
ρ_a	-	Ambient Fluid Density
ρ_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)
ψ_m	-	Atmospheric Stability Function
Ri_b	-	Bulk Richardson Number

3 Program setup

3.1 Installation

Untar the gzipped file `twodee-2.6.tar.gz` issuing the command “`tar xvf twodee-2.6.tar.gz`”. This will generate directory `twodee-2.6` (see Figure 1). Enter directory `twodee-2.6` and issue the command “`./configure`”, then execute the command “`make install`”. After compilation you may issue the command “`make clean`” to remove unneeded files.
Please, note that the directory `twodee-2.6` contains subdirectories `doc` (documentation), `src` (sources), `example` (example of input files), README.md, ChangeLog and LICENSE files and other stuff needed for automatic configuration and installation. The automatic configuration is based on `autotools`.

3.2 Folder structure

Figure 1 shows the folder structure. The directory `src` contains the source files of TWODEE, the directory `example` contains two examples and directory `doc` contains this manual. The directories `example/example1/outfiles` or `example/example2/outfiles` are created by the program `twodee` and contain the output files.

3.3 Program run

This version of TWODEE is suited for use under Unix/Linux/MacOS operating systems.

On these systems, TWODEE can be launched typing:

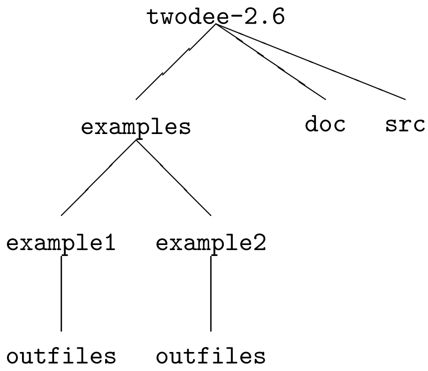

Figure 1: Directory tree of `twodee-2.6`.

`twodee 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. The second argument (log file name) is optional; if not provided, the log file is printed on the screen.

<u>NOTE</u>: To create a new run simply create a new folder, copy the input files and into it and run `twodee`. If you intend to use DIAGNO, you should run `presfc`, `preupr` and `diagno` before running `twodee`. Please, note that starting from TWODEE version 2.6, the program DIAGNO is distributed separately from the TWODEE package.

4 The TWODEE input files

TWODEE needs of the following input files:

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

<u>NOTE</u>: 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 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.*: $12e7 = 12 \times 10^7$).

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 °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 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.

- `Z_ROUGHNESS_(M)` : Terrain roughness height z_o (in m). The surface roughness height is related to the vertical wind profile. Values 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.

4.1.5 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`.

- `RESTART_FILE_PATH`: Name (including relative or absolute path) of the restart file. See section 4.3 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 4.4 for file format details.

- `WIND_FILE_PATH`: Name (including relative or absolute path) of the wind data file. See section 4.5 for file format details.

- `DIAGNO_FILE_PATH`: Name (including relative or absolute path) of the DIAGNO output file. This is an unformatted FORTRAN file created by DIAGNO. Only used when `WIND_MODEL = DIAGNO`.

- `SURF_DATAFILE`: The file surfacedata.txt contains the meteorologica information only used if `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 4.6 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 postprocess. Only used when `TRACK_BOXES= YES`. See section 4.7 for file format details.

- `OUTPUT_DIRECTORY`: Name (including relative or absolute path) of the folder where TWODEE 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.
- `OUTPUT_MAX_CONCENTRATION`: Flag indicating whether the dense gas maximum 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`.
- `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 applications.

4.2 The topography file `topography.grd`

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 reads the topography file and automatically interpolates elevations onto the nodes of the computational grid. The file format is the Golden Surfer Ascii Grid, described in Sec. 5.2, with the following meaning:

- `DSAA` : Magic number for the Golden Surfer Ascii Grids.
- `NX` : Number of ground elevation points along x-direction.
- `NY` : Number of ground elevation points along y-direction.
- `XMIN` : x-coordinate (UTM in m) of the bottom left corner point.
- `XMAX` : x-coordinate (UTM in m) of the top right corner point.
- `YMIN` : y-coordinate (UTM in m) of the bottom left corner point.
- `YMAX` : y-coordinate (UTM in m) of the top right corner point.
- `ZMIN` : Minimum value of the elevations in the grid.
- `ZMAX` : Maximum value of the elevations in the grid.
- `Z` : Elevation (in m) of each grid point of the regional grid. It consists of an array of `NX`×`NY` values stored starting from the bottom-left corner and moving towards right then towards the top of the domain.

4.3 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 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 3 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).
- `X0` : x-coordinate (UTM in m) of the bottom left corner.
- `Y0` : 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.

4.4 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 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 4 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).
- `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 (kg $m^{-2}s^{-1}$), GR_M2_SEC (g $m^{-2}s^{-1}$), TN_M2_DAY (ton $m^{-2}day^{-1}$), KG_M2_DAY (kg m^{-2} day^{-1}), GR_M2_DAY (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 (kg s^{-1}), GR_SEC (g s^{-1}), TN_DAY (ton day^{-1}), KG_DAY (kg day^{-1}), GR_DAY (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 (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`.

4.5 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 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 simply uses the DIAGNO output (file `diagno.out`). The file format is described in Table 5 (if `code=CUP`) and Table 6 (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 °C) at the ground level.
- `T_zref` : Temperature (in °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).

<u>NOTE</u>: 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.

4.6 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 7 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).

4.7 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 8 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.

4.8 The DIAGNO file `diagno.out`

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

5 The TWODEE output files

At each user-specified time (see `OUTPUT_INTERVAL_(SEC)` record in the control file) TWODEE can generate 2D contour-files written in GRD-format for the following variables: h, u, v, ρ, 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 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 GRD files are conform to the ascii version of Golden Software© ASCII grids, which is not-proprietary (see Table 9).

- `DSAA` : Magic number of the Golden Software ASCII grids
- `NX NY` : Number of grid points along x and y directions
- `XMIN XMAX` : x-coordinates (UTM in m) of the grid left and right corners corners
- `YMIN YMAX` : y-coordinates (UTM in m) of the grid bottom and top corners
- `ZMIN ZMAX` : Minimum and maximum values in the matrix (VAL)
- `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.

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 output files

Files in GRD format can be readed directly by several plotting programs like the commercial software Surfer©. Alternativelly, the user may also generate its own plots using functons from several free packages (*e.g.*: gnuplot). 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 (DPC) of Italy. This work does not necessarily represent DPC official opinion and policies.

References

Blackmore, H. and Woodward: Heavy gas dispersion models, J. Hazard Mater., 6, 107–128, doi:10.1016/0304-3894(82)80036-8, 1982.

Burguete, J., Garcia-Navarro, P., and Aliod, R.: Numerical simulation of runoff from extreme rainfall events in a mountain water catchment, Nat. Haz. Earth Syst. Sci., 2, 109–117, doi:10.5194/nhess-2-109-2002, 2002.

Clarke, T.: Taming Africa's killer lake, Nature, 409, 554–555, doi:10.1038/35054609, 2001.

Costa, A., Macedonio, G., and Chiodini, G.: Numerical model of gas dispersion emitted from volcanic sources, Ann. Geophys-Italy, 48, 805–815, 2005.

Ermak, D.: User's manuals for SLAB: An atmospheric dispersion model for denser-then-air releases, Lawrence livermore national laboratory, UCRL-MA-105607, CA, 1990.

Folch, A., Costa, A., and Hankin, R. K. S.: Twodee-2: A shallow layer model for dense gas dispersion on complex topography, Comput. Geosci., 35, 667–674, doi:10.1016/j.cageo.2007.12.017, 2009.

Hankin, R. K. S. and Britter, R. E.: 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, 1999a.

Hankin, R. K. S. and Britter, R. E.: 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, 1999b.

Hankin, R. K. S. and Britter, R. E.: 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, 1999c.

Heinrich, P., Piatanesi, A., and Hébert, H.: Numerical modelling of tsunami generation and propagation from submarine slumps: the 1998 Papua NewGuinea event, Geophy. J. Int., 145, 97–111, doi:10.1111/j.1365-246X.2001.00336.x, 2001.

Jacobson, M. Z.: Fundamentals of atmospheric modelling, Cambridge University Press, New York, first edn., 1999.

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

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

Rogie, J., Kerrick, D., Chiodini, G., and Frondini, F.: Flux measurements of nonvolcanic CO_2 emission from some vents in central Italy, J. Geophys. Res., 105, 8435–8445, doi:10.1016/1352-2310(95)00287-9, 2000.

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

Venetsanos, A., Bartzis, J., Wurtz, J., and Papailiou, D.: DISPLAY-2: a two-dimensional shallow layer model for dense gas dispersion including complex features, J. Hazard. Mater., A99, 111–144, doi:10.1016/S0304-3894(03)00011-6, 2003.

Witlox, H.: The hegadas model for ground-level heavy-gas dispersion i.steady-state model, Atmos. Environ., 28, 2917–2932, doi:10.1016/1352-2310(94)90340-9, 1994.

Table 2: Sample of the input control file `problemname.inp`

```
TITLE
  PROBLEM_NAME = example2

TIME
  YEAR = 2003
  MONTH = 03
  DAY = 31
  HOUR = 00
  MINUTE = 00
  SIMULATION_INTERVAL_(SEC) =  1800
  RESTART_RUN = NO

GRID
  UTM_ZONE  = 33S
  UTM_DATUM = WGS_84

  X_ORIGIN_(UTM_M) = 511300
  Y_ORIGIN_(UTM_M) = 4535600
  NX      = 400
  NY      = 170
  DX_(M) = 3
  DY_(M) = 3

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        = DIAGNO
  Z_REFERENCE_(M)   = 2.0
  X_STATION_(UTM_M) = 511920
  Y_STATION_(UTM_M) = 4535750

  NC_DIM_X_NAME = lon
  NC_DIM_Y_NAME = lat
  NC_DIM_Z_NAME = z
  NC_DIM_T_NAME = time

  NC_VAR_DATE_NAME = DATE
  NC_VAR_HOUR_NAME = HOUR
  NC_VAR_U_NAME    = U-VEL-WRF
  NC_VAR_V_NAME    = V-VEL-WRF
  NC_VAR_T_NAME    = T-WRF

FILES
  OUTPUT_DIRECTORY        = outfiles
  TOPOGRAPHY_FILE         = topography.grd
  TOPOGRAPHY_FILE_FORMAT  = GRD
  ROUGHNESS_FILE          = roughness.grd
  ROUGHNESS_FILE_FORMAT   = GRD
  RESTART_FILE            = restart.dat
  SOURCE_FILE             = source.dat
```

```
  WIND_FILE               = wind.dat
  DIAGNO_FILE             = diagno.out
  SURF_DATA_FILE          = meteo_surface.dat
  MET_NC_FILE             = winds.nc

  TRACK_POINTS            = NO
  TRACK_POINTS_FILE       = points.pts
  TRACK_BOXES             = NO
  TRACK_BOXES_FILE        = boxes.dat

OUTPUT
  OUTPUT_INTERVAL_(SEC)  = 600

  OUTPUT_GRD_FILE_FORMAT = no
  OUTPUT_NC_FILE_FORMAT  = yes

  OUTPUT_DOMAIN          = yes
  OUTPUT_Z0              = yes
  OUTPUT_SOURCE          = yes
  OUTPUT_METEO           = yes

  OUTPUT_VELOCITY        = yes
  OUTPUT_H               = yes
  OUTPUT_RHO             = yes
  OUTPUT_DOSE            = yes
  OUTPUT_CONCENTRATION   = yes
  OUTPUT_MAX_CONCENTRATION = yes
    HEIGHTS_(M)          = 0.05 0.5 1.0 1.5 2.0 2.5 3.0
    CONCENTRATION_BG     = 350.
  OUTPUT_Z_CRITICAL      = yes
    CRITICAL_C_(%)       = 5 10 15
  OUTPUT_IMPACT          = no

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
```

Table 3: Format of the restart file `restart.dat`

comments							(7 lines)
tstart	NX	NY	DX	DY	X0	Y0	
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 4: Format of the source file `source.dat`

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

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

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 7: Format of the file `point.dat`

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

Table 8: Format of the file `boxes.dat`

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 9: Format of a GRD file `file.grd`

DSAA			
NX	NY		
XMIN	XMAX		
YMIN	YMAX		
MIN(VAL)	MAX(VAL)		
VAL(i,1)	...	...	i=1:NX
...	...	...	
VAL(i,j)	...	...	i=1:NX
...	...	...	
VAL(i,NY)	...	...	i=1:NX