

A MODEL FOR PASSIVE
DISPERSION OF GAS
AND PARTICLES

DISGAS-2.6.0

User Manual

Antonio Costa¹ and Giovanni Macedonio²

1 Istituto Nazionale di Geofisica e Vulcanologia, Sezione di
Bologna, Via Berti Pichat 6/2, Bologna, Italy

2 Istituto Nazionale di Geofisica e Vulcanologia, Osservatorio
Vesuviano, Via Diocleziano 328, Napoli, Italy

Manual version: 2.6.0 - 20 November 2023

License

DISGAS is a 3-D time-dependent Eulerian model for transport of gas in the atmospheric surface layer.

Copyright (C) 2014, 2016, 2020, 2021, 2022, 2023, Antonio Costa and Giovanni Macedonio

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

A copy of the GNU General Public License can be found in the file COPYING included in the distributed DISGAS package.

How to cite disgas

Costa, A. and Macedonio, G. (2016). DISGAS-2.0: A model for passive dispersion of gas, Rapporti Tecnici n.332, INGV, Italy, ISSN 2039-7941, <https://doi.org/10.13127/rpt/332>

Contents

1	Introduction	5
2	Physical model	6
2.1	Wind field	6
2.2	Turbulence parameterization	7
3	Numerical solution	8
4	Application example	9
5	Fixed bugs	10
A	Program setup	10
A.1	Installation	10
A.2	Folder structure	11
A.3	Program run	12
B	The DISGAS input and output files	12
B.1	The control file <code>problemname.inp</code>	13
B.1.1	BLOCK TIME	13
B.1.2	BLOCK GRID	13
B.1.3	BLOCK PROPERTIES	14
B.1.4	BLOCK TOPOGRAPHY	14
B.1.5	BLOCK METEO	14
B.1.6	BLOCK FILES	15
B.1.7	BLOCK OUTPUT	15
B.2	The topography file <code>topography.grd</code>	16
B.3	The source file <code>source.dat</code>	16
B.4	The roughness length file <code>roughness.grd</code>	16
B.5	The wind data file <code>winds.dat</code>	16
B.6	The DIAGNO file <code>diagno.out</code>	17
B.7	The DISGAS output files	17
B.8	The <code>problemname.log</code> file	17
B.9	The restart file <code>restart.dat</code>	18
C	Format of the GRD files	18
	References	21

List of Figures

1	Average CO ₂ concentration maps at 1.5 m above ground level assuming the mean CO ₂ emission rate of 1067 ton/d (figures a and b on the top) and the maximum CO ₂ emission rate measured in the 1998-2008 period of 1800 ton/d (figures c and d on the bottom), over a whole diurnal cycle (figures a and c) and over nocturnal periods (figures b and d). Modified after Granieri et al. (2013).	11
2	Directory tree of <code>disgas-2.6.0</code>	12

List of Tables

1	Sample of the input control file <code>problemname.inp</code>	19
2	Format of the wind file <code>winds.dat</code> with <code>code=CUP</code>	20
3	Format of the wind file <code>winds.dat</code> with <code>code=SONIC</code>	20
4	Format of a GRD file	20

Abstract

The code DISGAS (model for passive DISpersion of GASEs) is a Eulerian model for passive dispersion of diluted gas and fine dust particles. Turbulent diffusion is based on the K-theory and the wind field can be evaluated assuming either a uniform wind profile based on the Similarity Theory or using a terrain-following mass-consistent wind model. DISGAS can be used to forecast concentration of gas (or dust) over complex terrains. The inputs to the model are topography, wind observations from meteorological stations, atmospheric stability information, and gas flow rate from a discrete number of point sources. DISGAS is written in fortran 90. Here we describe the model and an application example.

1 Introduction

Several volcanic and non-volcanic sources in central Italy and worldwide emit a large amount of gas, such as carbon dioxide (CO₂) and other components, through focused or diffuse degassing, producing consistent gas plumes that are dispersed in the lower part of the atmosphere (e.g., [Chiodini et al., 2005](#); [Costa et al., 2005](#); [Chiodini et al., 2007](#)).

The cloud dispersion of a gas denser than air released from a given source, is governed by the gravity and by the effects of lateral eddies which increase the mixing with air around the edges of the plume, decreasing its density. In the initial phase the negative buoyancy controls the gas dispersion and the cloud follows the ground (gravitational phase). During this phase, the dispersion of heavy gas is markedly different from passive gas dispersion (and even from positively buoyant gas). When density contrast is not important, gas dispersion is basically governed by wind and atmospheric turbulence (passive dispersion).

From a theoretical point of view, dispersion of natural gases can be fully studied by solving the complete equations for mass, momentum, and energy transport. However, in actual practice, different simplified approaches able to describe only specific phases or aspects of gas transport are commonly used. The most general and computationally most expensive models are given by three-dimensional CFD models based on the transport theory of mass, momentum, energy, and species (e.g., [Macedonio and Costa, 2002](#); [Cortis and Oldenburg, 2009](#); [Cerminara et al., 2016](#)). This approach is able to simulate dispersion of both heavy and light gases accounting for topographic effects, variation of atmospheric conditions and wind direction, etc.

For a gas denser than air the shallow layer approach, which uses depth-averaged variables to describe the flow behaviour, is commonly used (e.g., [Hankin and Britter, 1999](#); [Costa et al., 2008](#); [Folch et al., 2009](#)). Models based on this approach are applied to describe gravity driven flows of dense gas over complex topography ([Costa et al., 2008](#); [Chiodini et al., 2010](#)).

For dispersion of a diluted gas, which is passively driven by wind advection and atmospheric turbulence, simpler advection-diffusion equations can be solved (e.g., [Prabha and Mursch-Radlgruber, 1999a](#)), like in the model we describe here. DISGAS ([Costa and Macedonio, 2016](#)) is based on an explicit solver of the advection-diffusion equation which uses a second-order Lax-Wendroff scheme (e.g., [Ewing and Wang, 2001](#)) and can be coupled with a wind profile described by the similarity theory ([De Bruin et al., 2000](#)) or with a mass-consistent wind model ([Douglas and Kessler, 1990](#)).

The generalization of the advection-diffusion equation including a sedimentation term, allows the model to be applied to study even the dispersion of a diluted mixture of very fine particles into the atmospheric surface layer. For instance, with such a generalization the model can describe the dispersion of small fibres like asbestos or small powders like pollen.

Here, firstly we summarize the theory and equations on which the model is based, then we illustrate the numerical method used to solve the equations, and, in the Appendix A, we describe the program setup, folder structure, input/output files, and how to run the code. Simulation results obtained for the dispersal of CO₂ emitted from the Solfatara crater in the Campi Flegrei, in surroundings areas of the city of Naples, Italy, is also presented as an example of application ([Granieri et al., 2013](#)). The format and the description of input and output files is reported in the Appendix B and Appendix C.

2 Physical model

A full description of gas dispersion into the atmosphere requires the solution of mass, momentum, energy and species transport equations (see *e.g.* [Macedonio and Costa, 2002](#)). This implies a relatively high computational effort and sometimes it is impossible to study the gas dispersion in complex large domains, especially in the surface layer where a large range of length scales are involved.

In order to reduce the computational time we introduce some assumptions. Our main objective is to solve the advection-diffusion equation for the gas concentration $c = \langle c \rangle + c'$:

$$\frac{\partial \langle c \rangle}{\partial t} + \frac{\partial u_x \langle c \rangle}{\partial x} + \frac{\partial u_y \langle c \rangle}{\partial y} + \frac{\partial u_z \langle c \rangle}{\partial z} + \frac{\partial \langle u'_x c' \rangle}{\partial x} + \frac{\partial \langle u'_y c' \rangle}{\partial y} + \frac{\partial \langle u'_z c' \rangle}{\partial z} = \langle Q \rangle \quad (1)$$

where $\mathbf{V} = (u_x + u'_x, u_y + u'_y, u_z + u'_z)$ is the wind field and $\langle Q \rangle$ is a source term, i.e. the gas mass flux from the ground. Plain terms in lower case letters represent the average part, terms with the symbol prime ' their turbulent fluctuations and the overbar defines the filtering operation on the product of the fluctuation terms (*e.g.*, [Sagaut, 2006](#)).

Hence, as in other studies (*e.g.*, [Prabha and Mursch-Radlgruber, 1999a,b](#); [Costa et al., 2005](#)), we do not solve the complete set of the coupled equations for mass, momentum, energy, and concentration, but we solve eq. (1) and approximate the wind field as given by a uniform profile (for relatively small domains) or estimate it using a diagnostic wind model which produces a zero three-dimensional divergence velocity field consistent with the measured values, avoiding artificial generation or loss of gas. Turbulent terms are parameterized according with the K-theory as described later. The passive dispersion approach can be used when the gas is diluted enough, and the validity of such an approximation can be assessed by estimating the Richardson number (*e.g.*, [Bitter and McQuaid, 1988](#); [Cortis and Oldenburg, 2009](#); [Costa et al., 2013](#)):

$$R_i = \frac{(g'q)^{2/3}}{V^2 R^{2/3}} \quad (2)$$

where $g' = (g(\rho_g - \rho_a))/\rho_a$ is the reduced gravity acceleration (being g the gravity acceleration, ρ_a and ρ_g the air and the gas density respectively), q is the volumetric flow rate, R the plume size (*e.g.* plume radius), and V is the wind velocity at the reference altitude (*i.e.*, 10 m). For $R_i < 0.25$ transport is substantially passive whereas for $R_i > 1$ is mainly density driven ([Cortis and Oldenburg, 2009](#); [Costa et al., 2013](#)).

2.1 Wind field

For evaluating the wind field there are two options:

1. The simplest choice (SIMILARITY) consists of assuming a horizontally uniform wind profile calculated in accord to the wind measurements and the Monin-Obukhov similarity theory ([Louis, 1979](#); [Monin and Yaglom, 1979](#); [Byun, 1990](#)):

$$u(z) = \frac{u_*}{k} [\ln(z/z_0) - \Psi_m(z/L)] \quad (3)$$

where $\kappa = 0.4$ is the von Karman constant, z_0 is the roughness length, u_* the friction velocity and $\Psi_m(x)$ the atmospheric stability function for the momentum (see *e.g.* [Dyer, 1974](#); [Jacobson, 1999](#)). The friction velocity can be calculated as ([Louis, 1979](#)):

$$\frac{u_*}{k} = \frac{u(z_1)}{\ln(z_1/z_0) - \Psi_m(z_1/L) + \Psi_m(z_0/L)} \quad (4)$$

where z_1 is the height at which the wind velocity $u(z_1)$ is known. Roughness can be assumed uniform or given as a matrix covering the computational domain. Using the SIMILARITY option

implies the wind can be not mass-consistent (e.g. $\nabla \cdot V$ can be locally not null). It is adequate for flat small domains but is not suitable for large domain and complex terrains.

2. Another choice (DIAGNO) consists of using the wind field estimated by the Diagnostic Wind Model (DWM), a model developed by the United States Environmental Protection Agency (EPA), here named DIAGNO. This model generates the wind components (U, V, W) in a terrain following coordinates system.

The DIAGNO needs topographic data, average wind on the computational domain and atmospheric stability information within the scale of the domain (the temperature gradient dT/dz).

In a first step the domain-mean wind is adjusted for the kinematic effects of terrain (lifting and acceleration of the airflow over terrain obstacles), thermodynamically generated slope flows, and blocking effects.

In a second step, wind observations, when available, are added to the first step field, and an objective analysis scheme is used to produce a new gridded field (U, V, W). The scheme is designed so that the observations are used to define the wind field within a user-specified radius of influence while the first step (U, V, W) field is used in subregions in which observations are unavailable.

Finally, a divergence-minimization procedure in terrain following coordinates is iteratively applied until the inequality:

$$\nabla \cdot \mathbf{U} < \epsilon \quad (5)$$

is satisfied ($\mathbf{U} = (U, V, W)$ and ϵ is an arbitrarily user defined small number). The final product of DWM is an approximately null-divergence wind field consistent with the observations (for further information see [Douglas and Kessler, 1990](#)).

The approximation of null-divergence wind field is generally applicable up to a kilometer above ground level ([Dutton and Fichtl, 1969](#)) therefore to our study, since we treat the surface layer, *i.e.* the lowest part of the Planetary Boundary Layer (PBL).

Since terrain-following coordinate are used such an approximation still holds even if ground level varies significantly within the computational domain. DISGAS model interpolates the wind field into its computational grid. Starting from DISGAS version 2.4.2, the wind field generated by DIAGNO (typically every hour) is interpolated also in time.

2.2 Turbulence parameterization

Turbulent fluxes are given by the product of the fluctuation terms, and in agreement with the ‘‘K-theory’’, they are expressed in terms of gradients of average concentration:

$$\langle u'_x c' \rangle = -K_x \frac{\partial}{\partial x} \langle c \rangle \quad \langle u'_y c' \rangle = -K_y \frac{\partial}{\partial y} \langle c \rangle \quad \langle u'_z c' \rangle = -K_z \frac{\partial}{\partial z} \langle c \rangle \quad (6)$$

where $K = (K_x, K_y, K_z)$ is the turbulent diagonal diffusivity tensor. In `disgas` the eddy diffusivity K_z and $K_H = K_x = K_y$ can be assumed either constant or considered variable. In the latter case, the user can specify optional minimum values for both vertical and horizontal diffusion coefficients (default values are $1 \text{ m}^2/\text{s}$ for the horizontal and vertical diffusion coefficients). For the parameterization of the minimum eddy diffusivity see [Lu et al. \(2023\)](#).

Vertical turbulent diffusivity. Inside the atmospheric surface layer, the Monin-Obukhov similarity theory allows to estimate the vertical turbulent diffusivity K_z in terms of friction velocity u_* and the Monin-Obukhov length L :

$$K_z = \frac{\kappa z u_*}{\phi_h} \quad (7)$$

where κ is the von Karman constant ($\kappa = 0.4$), z is the distance from the ground and ϕ_h is the dimensionless potential temperature gradient ([Byun, 1990](#); [Jacobson, 1999](#)):

$$\phi_h = \begin{cases} Pr_t + \beta_h \frac{z}{L} & z/L > 0 \quad \text{stable condition} \\ Pr_t (1 - \gamma_h \frac{z}{L})^{-1/2} & z/L < 0 \quad \text{unstable condition} \\ Pr_t & z/L = 0 \quad \text{neutral condition} \end{cases} \quad (8)$$

with $\beta_h = 7.8$, $\gamma_h = 11.6$ and $Pr_t \approx 0.95$. Therefore, evaluating K_z requires knowledge of the friction velocity u_* and the Monin-Obukhov length L .

From a practical point of view, L is often a quantity difficult to directly estimate. Instead, it is operationally easier to evaluate this length using the ‘‘Bulk Richardson number’’ method (Golder, 1972):

$$Ri_b = \frac{g}{\theta} \frac{\partial \theta}{\partial z} \frac{z_m^2}{u_m^2} \quad (9)$$

where g is the gravity and θ the potential air temperature (Louis, 1979; Jacobson, 1999), z_m the geometric mean thickness of the considered layer and u_m is the logarithmically interpolated wind speed at z_m . The Richardson number Ri_b is a measure of the dynamics on the buoyancy effects: we have atmospheric instability when $Ri_b < Ri_{crit}$ and stability for $Ri_b > Ri_{crit}$ ($Ri_{crit} \simeq 0.2$). The term z/L can be calculated in terms of Ri_b by using the relationship (Byun, 1990):

$$\frac{z}{L} = Ri_b \frac{\phi_m^2}{\phi_h} \quad (10)$$

where ϕ_m is the dimensionless wind shear function (Businger et al., 1971; Jacobson, 1999):

$$\phi_m = \begin{cases} 1 + \beta_m \frac{z}{L} & z/L > 0 \quad \text{stable condition} \\ (1 - \gamma_m \frac{z}{L})^{-1/4} & z/L < 0 \quad \text{unstable condition} \\ 1 & z/L = 0 \quad \text{neutral condition} \end{cases} \quad (11)$$

with $\beta_m = 6.0$ and $\gamma_m = 19.3$ estimated from field data (assuming $\kappa = 0.4$) (Byun, 1990);

As explained in the Appendix B, Monin-Obukhov length L can be directly furnished by the user (if the option SONIC is used for the wind file) or estimated from air temperature difference following the ‘‘Bulk Richardson number’’ method (if the option CUP is used for the wind file).

Horizontal turbulent diffusivity. Following the large eddy approach (e.g., Smagorinsky, 1963; Pielke et al., 1992; Byun and Schere, 2006), the horizontal eddy diffusivity $K_x = K_y = K_H$ is considered isotropic and estimated as (Byun and Schere, 2006):

$$K_h = Pr_t \alpha^2 \Delta x \Delta y \sqrt{\left(\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right)^2} \quad (12)$$

where $\alpha \simeq 0.28$ is a dimensionless constant empirically determined, and Δx and Δy the grid spacing (in DISGAS-2.6.0 we set $\alpha = 3$). A lower limit of $K_H = 0.075(\sqrt{\Delta x \Delta y})^{4/3}$ m²/s was imposed (Pielke et al., 1992). As we mentioned above, here we neglect the off-diagonal horizontal and vertical turbulent diffusion components (Toon et al., 1988; Jacobson et al., 1996; Park and Kim, 1999). For a more general treatment that includes the off-diagonal diffusion components see (Byun and Chin, 1999).

3 Numerical solution

Considering the assumptions introduced above and a null-divergence wind field, in a terrain-following coordinate system (Douglas and Kessler, 1990) consisting of:

$$X = x, \quad Y = y, \quad Z = z - h(x, y) \quad (13)$$

being $(h(x, y))$ the height of the topography, eq. (1) for the mean scaled concentration C may be re-written in a generalized form as (Toon et al., 1988; Jacobson et al., 1996; Costa, 2006):

$$\begin{aligned} \frac{\partial C}{\partial t} + U \frac{\partial C}{\partial x} + V \frac{\partial C}{\partial y} + W \frac{\partial C}{\partial z_*} = \\ \frac{\partial}{\partial x} \left(K_h \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial z_*} \left(K_z \frac{\partial C}{\partial z_*} \right) + \langle Q_* \rangle \end{aligned} \quad (14)$$

here (U, V, W) denote the scaled wind speeds, K_h and K_z are the diagonal scaled diffusion coefficients and Q_* the source term in the new coordinates. In the case of the simple transformations (13) the Jacobian is equal to the unit, $J = \partial z / \partial z_* = 1$, therefore $c(x, y, z) = C(x, y, z_*)$ and $Q = Q_*$. In the case of particle dispersion, here W represents the sum of the vertical wind component and of the particle settling velocity V_s , *i.e.* $W = V_{Z, Wind} - V_s$.

The advective terms are discretised according to the Lax-Wendroff scheme which is second-order (see *e.g.* Ewing and Wang, 2001). For instance, the advective term in the X -direction yields:

$$-\Delta t \left(U \frac{\partial C}{\partial x} \right)_{i,j,k}^n = -\frac{Cr}{2} (C_{i+1,j,k}^n - C_{i-1,j,k}^n) + \frac{Cr^2}{2} (C_{i+1,j,k}^n - 2C_{i,j,k}^n + C_{i-1,j,k}^n) \quad (15)$$

where $Cr = U\Delta t / \Delta x$ represents the Courant number (Δx is the grid spacing and Δt the computational time step). In order to reduce the numerical over- and under-shooting that commonly affects high-order schemes near discontinuities one can use slope limiter methods. In fact, these methods preserve the monotonicity of the solution while the accuracy remain higher than the first order upwind methods (Sweeby, 1984; Wang and Hutter, 2001; Ewing and Wang, 2001). For instance, applying the minmod slope-limiter method and considering the general case with non-uniform velocity, equation (15) becomes:

$$-\Delta t \left(U \frac{\partial C}{\partial X} \right)_{i,j,k}^n = -\frac{\Delta t}{\Delta x} (U_{i,j,k}^n C_{i,j,k}^n - U_{i-1,j,k}^n C_{i-1,j,k}^n) - \frac{\Delta x}{2} [Cr_{i,j,k}^n (1 - Cr_{i,j,k}^n) \sigma_{i,j,k}^n - Cr_{i-1,j,k}^n (1 - Cr_{i-1,j,k}^n) \sigma_{i-1,j,k}^n] \quad (16)$$

where $Cr_{i,j,k}^n = U_{i,j,k}^n \Delta t / \Delta x$, and:

$$\sigma_{i,j,k}^n = \text{minmod} \left\{ \frac{C_{i+1,j,k}^n - C_{i,j,k}^n}{\Delta x}, \frac{C_{i,j,k}^n - C_{i-1,j,k}^n}{\Delta x} \right\} \quad (17)$$

with $\text{minmod}(a, b) = 1/2(\text{sgn}(a) + \text{sgn}(b)) \min(|a|, |b|)$ (Ewing and Wang, 2001). The introduction of alternative limiter methods is straightforward.

The diffusion terms are evaluated using a central difference scheme for the general case with a not uniform turbulent diffusivity tensor, *e.g.* in the z -direction using:

$$\begin{aligned} \frac{\partial}{\partial x} \left(K_h \frac{\partial C}{\partial x} \right) &= \frac{1}{2\Delta x^2} \left[(K_h^{i,j,k} + K_h^{i-1,j,k}) C_{i-1,j,k} \right. \\ &\left. - (K_h^{i+1,j,k} + 2K_h^{i,j,k} + K_h^{i-1,j,k}) C_{i,j,k} + (K_h^{i+1,j,k} + K_h^{i,j,k}) C_{i+1,j,k} \right] \end{aligned} \quad (18)$$

The stability of the numerical scheme is ensured by using a time step Δt lower than the critical. As established by Hindmarsh et al. (1984), an explicit scheme for the multidimensional advection diffusion equation is numerically stable if the condition:

$$\Delta t \leq \frac{1}{\left[2 \left(\frac{K_h}{\Delta x^2} + \frac{K_h}{\Delta y^2} + \frac{K_z}{\Delta z_*^2} \right) + \frac{U}{\Delta x} + \frac{V}{\Delta y} + \frac{W}{\Delta z_*} \right]} \quad (19)$$

Concerning the z -direction, the numerical scheme was generalized to a non-uniform vertical grid.

4 Application example

DISGAS code was applied to simulate gas dispersion from the Solfatara volcano, Campi Flegrei, Italy (Granieri et al., 2013). Solfatara releases more than 1000 t d⁻¹ of CO₂ through soil diffuse degassing from a relatively small area (0.5-1 km²) (Chiodini et al., 2001; Granieri et al., 2013). Temperature of the gases emitted from Solfatara area is relatively high: fumaroles temperature is between 96 °C and 162 °C (Chiodini et al., 2001) and, flux weighted temperature of diffusing soil is 66 °C (Costa et al., 2005).

This implies a density decrease due to temperature that almost balances the increase due to the greater molecular weight directly at the source ($M_{CO_2}/M_{air} = 44/29$ while $T_{CO_2}/T_{air} \approx 400/300$). Moreover low wind conditions are rarely recorded at Solfatara (daily average wind intensity $U > 1$ m/s is about 75% of the total and the cases with $U < 0.1$ m/s 0.05% of the total). For these reasons, the passive dispersion assumption, described in Section 2, is appropriate (Costa et al., 2005; Granieri et al., 2013).

Simulation results of gas dispersal were first compared with observations, then the model was used to investigate space and time evolution of the volcanic CO₂ plume under different meteorological conditions in order to assess its potential hazard.

Using DISGAS, Granieri et al. (2013) showed that the urban area of Naples is mantled by the volcanic plume of the Solfatara when atmospheric circulatory patterns are dominated by the locally frequent marine winds. In these conditions the air CO₂ content increases above the normal air CO₂ content, with differences varying from more than one thousand ppm (above the normal atmospheric background) in proximity of the Solfatara crater to a few ten ppm at a few kilometres from the source. Although these values indicate that there is no risk to the population, the elevated CO₂ concentration contributes to the deterioration of air quality in Naples, especially on days with atmospheric stagnant conditions.

An emission rate five times higher than the current one may pose a hazard even for the human health. In Figure 1 we reported some simulation results of Granieri et al. (2013). Simulations of typical CO₂ concentrations at 1.5 m of height (typical breathing level for humans) suggest that CO₂ concentration reaches higher values during nocturnal hours (from 2:00 to 7:00), when atmosphere is very stable. The level goes down up to a few hundred ppm at a few km from the source. During daytime, with unstable conditions and more intense wind circulation, gas is diluted and typical concentration values range from a few hundreds ppm near the source up to a few tens ppm (above the normal atmospheric background) at 3-4 km from the source (for further details see Granieri et al., 2013).

5 Fixed bugs

We found a bug introduced in version 2.2.4 (24 December 2021). Actually, for a typo in file `source.f90`, the input fluxes were divided by factor which was 4 for single precision and 8 for double precision. As a consequence, also the concentrations in the atmosphere were divided by the same number.

This bug was corrected in version 2.5.0.

Acknowledgments

This work was supported by the Department of Civil Protection of Italy, under the 2007-2009 INGV-DPC Agreement. We wish to thank Silvia Massaro for her contribution in editing this manual and Matteo Cerminara for his useful suggestions on a earlier version of the document.

Appendix A Program setup

Here we will describe how to install DISGAS model and the code structure.

A.1 Installation

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

1. Uncompress and untar file `disgas-2.6.0.tar.gz` with the commands:

```
tar xzvf disgas-2.6.0.tar.gz
```

This will generate directory `disgas-2.6.0` and the folder structure described in the Figure 2.

2. Configure, compile and install the program `disgas` and the utility (`grd2lw`). Enter the directory `disgas-2.6.0`, then issue the command `configure`, and `make install`.

After the command `make install`, the generated executable files `disgas` and `grd2lw` will be

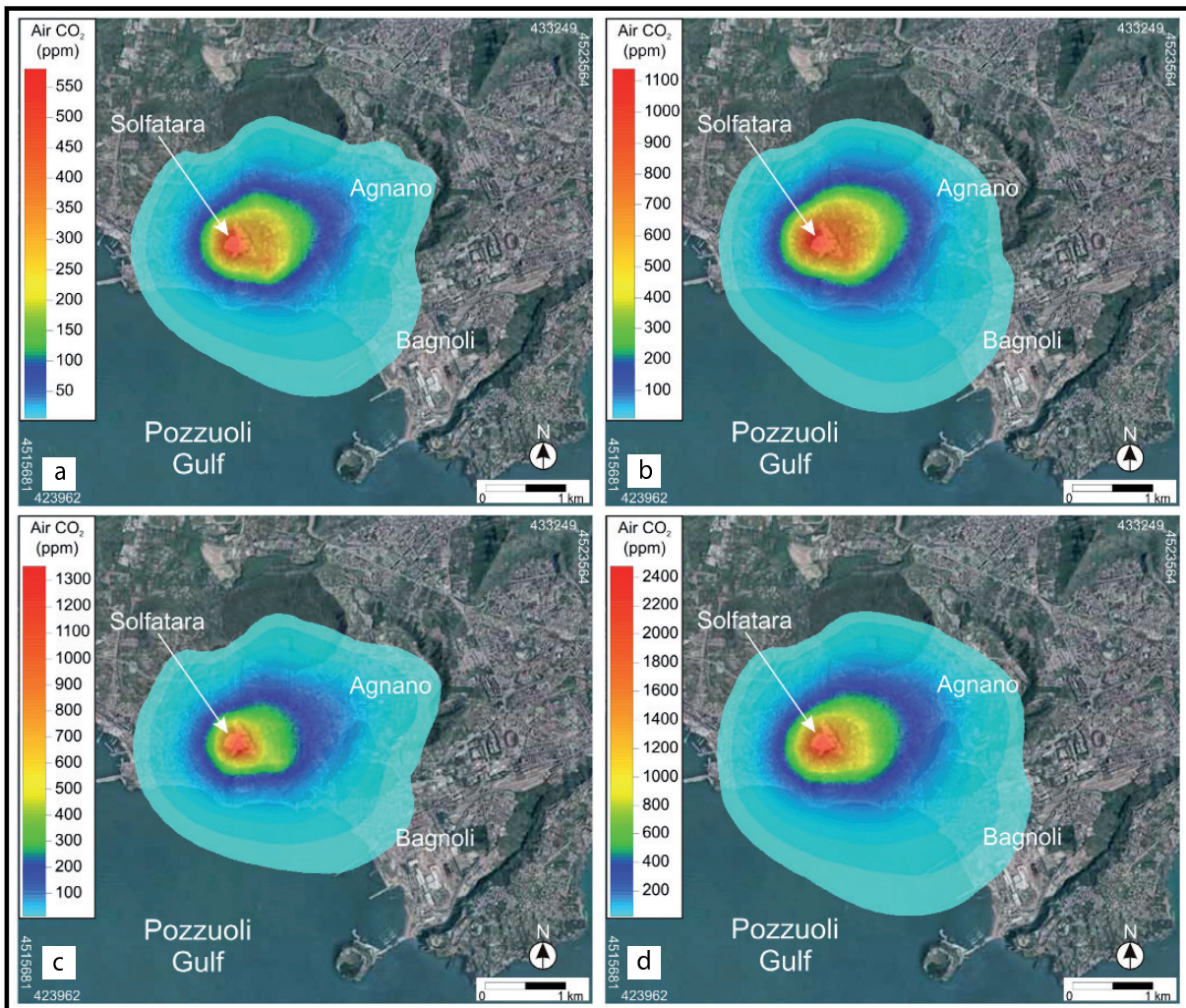


Figure 1: Average CO₂ concentration maps at 1.5 m above ground level assuming the mean CO₂ emission rate of 1067 ton/d (figures a and b on the top) and the maximum CO₂ emission rate measured in the 1998-2008 period of 1800 ton/d (figures c and d on the bottom), over a whole diurnal cycle (figures a and c) and over nocturnal periods (figures b and d). Modified after [Granieri et al. \(2013\)](#).

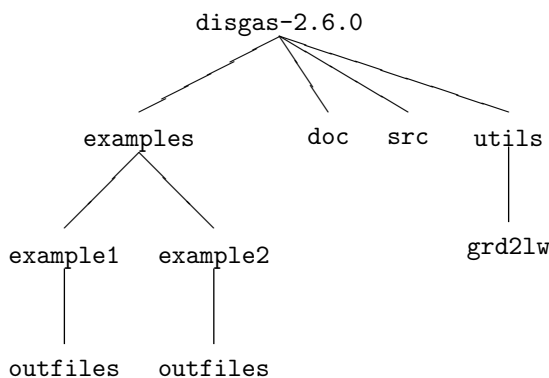
installed in the directory `$HOME/bin` (the directory `bin` under the home directory will be automatically created if needed). It is important that the directory `$HOME/bin` is in the `PATH` of your shell. After compilation you may issue the command `make clean` to remove unneeded files.

Please note that since `disgas-2.1`, the utility `diagno` is distributed separately. This version is compatible with `diagno-1.1.6`.

A.2 Folder structure

Figure 2 shows the folder structure of `DISGAS-2.6.0`. When untarred, `disgas-2.6.0.tar.gz` generates a directory tree as shown in Figure 2. The directory `disgas-2.6.0/src` contains the source code of `DISGAS` in FORTRAN-90 whereas directory `bin` is automatically generated during the installation process and contains the executable binaries. The source codes of the provided utilities are contained under `disgas-2.6.0/utills`.

The directory `disgas-2.6.0/examples` contains two subdirectories `example1` and `example2` with exam-

Figure 2: Directory tree of `disgas-2.6.0`.

ples of applications of DISGAS, which can be used to test the installation.

Example 1 adopts a uniform wind (SIMILARITY flag in `disgas.inp`), whereas example 2 adopts a wind generated by program `diagno`. Please, note that since DISGAS-2.1 `diagno` is distributed separately. Further details on how to run the examples are contained in file `example/README`. The directory `outfiles` where the output files are generated is automatically created by `disgas` (if non existent). The pathname of the output directory is defined in the input file of `disgas`. Finally, the `doc` directory contains the documentation of DISGAS (this file).

A.3 Program run

For running the example of `disgas-2.6.0` you need to previously install `diagno-1.1.6`, which provides the utilities `presfc` and `diagno`.

Then, programs `presfc`, `diagno` and `disgas` can be launched typing, respectively,

- `presfc`
- `diagno`
- `disgas problemname.inp`

where `problemname.inp` is the name of the control input file (see Appendix B). The file `diagno.inp` denotes the DIAGNO input file. Beside `diagno.inp` for running DIAGNO (distributed separately), it is necessary to furnish, in the same directory, a file `surface.dat` containing data from wind stations and a topography file `topography.grd`. The file `surface.dat` can be generated using the utility `PRESFC` that needs the input file `presfc.dat` (for further descriptions see [Douglas and Kessler, 1990](#)). The DIAGNO computational domain has to be equal or larger than the DISGAS computational domain. **NOTE:** To create a new run simply create a new folder in the directory `examples` (with subfolder `outfiles`), copy the input files and the `problemname.inp`. **NOTE:** To create a new run simply create a new folder in the directory `examples` (with subfolder `outfiles`), copy and edit the input files.

Appendix B The DISGAS input and output files

DISGAS 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: `source.dat` - Source term (gas fluxes) file. Mandatory.
- File: `winds.dat` - Meteorological data file. Optional.

- File: `diagno.out` - DIAGNO meteorological data file. 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`.

B.1 The control file `problemname.inp`

The DISGAS 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 1 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.*: $12e7 = 12 \times 10^7$).

B.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 seconds).
- RESTART_RUN: Select whether the simulation loads the restart file (NO or YES).
- RESET_TIME: In case of start from the restart file, select whether the internal variable `time` is reset to zero or continues from last dump. This affects also the reset or the continuation of the numeration of the output files (NO or YES).

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

B.1.2 BLOCK GRID

- NX: Number of grid cells along the *x*-direction.
- NY: Number of grid cells along the *y*-direction.
- NZ: Number of layers along the *z*-direction.
- Z_LAYERS_(M): List of the quotes of the layers in the *z*-direction. They must be exactly NZ.
- DX_(M): Grid spacing along the *x*-direction (in meters).
- DY_(M): Grid spacing along the *y*-direction (in meters).
- X_ORIGIN_(UTM_M): *x*-coordinate of the grid bottom left corner (UTM coordinates in meters).
- Y_ORIGIN_(UTM_M): *y*-coordinate of the grid bottom left corner (UTM coordinates in meters).

B.1.3 BLOCK PROPERTIES

- **DISPERSION_TYPE**: Type of material dispersed. Possibilities are **GAS** (no settling velocity) or **PARTICLES** (settling velocity is calculated in accord to the particle properties specified below).
- **PARTICLE_DIAMETER**: Diameter of particles in metres (only used if **DISPERSION_TYPE=PARTICLES**).
- **PARTICLE_DENSITY**: Density of particles in kg/m^3 (only used if **DISPERSION_TYPE=PARTICLES**).
- **PARTICLE_SHAPE_PARAMETER**: Particle shape parameter to be used to calculate particle settling velocity, *i.e.* particle sphericity or particle aspect ratio. In the case of [Ganser \(1993\)](#) model it corresponds to particle sphericity (only used if **DISPERSION_TYPE=PARTICLES**). **PARTICLE_MODEL_VSET**: Parameterization used for calculating particle settling velocity. Possibilities are: **ARASTOPOUR** for the [Arastoopour et al. \(1982\)](#) model (spherical particles), **GANSER** for [Ganser \(1993\)](#) model, **WILSON_HUANG** for [Wilson and Huang \(1979\)](#) model (only used if **DISPERSION_TYPE=PARTICLES**).

B.1.4 BLOCK TOPOGRAPHY

- **EXTRACT_TOPOGRAPHY_FROM_FILE**: Flag indicating whether the topography of the computational domain is extracted from a ground elevation file or not. Possibilities are **YES/NO**. If **YES**, topography is extracted from the file defined in the **FILES BLOCK**. If **NO**, a flat terrain is assumed with slopes in the x and y directions defined by the parameters **X_SLOPE_(DEG)** and **Y_SLOPE_(DEG)**.
- **Z_ORIGIN_(M)**: Quote of the bottom-left corner of the topography. This is used to generate a flat topography when the flag **EXTRACT_TOPOGRAPHY_FROM_FILE** is set to **NO**.
- **X_SLOPE_(DEG)**: Slope of the flat topography in the x direction (in degrees). This parameter is used to generate a flat topography when the flag **EXTRACT_TOPOGRAPHY_FROM_FILE** is set to **NO**.
- **Y_SLOPE_(DEG)**: Slope of the flat topography in the y direction (in degrees). This parameter is used to generate a flat topography when the flag **EXTRACT_TOPOGRAPHY_FROM_FILE** is set to **NO**.

B.1.5 BLOCK METEO

- **WIND_MODEL**: Flag indicating the wind model. Possibilities are **SIMILARITY** or **DIAGNO**. If **SIMILARITY** is set, the wind (horizontally uniform) is read from a wind file defined in the **BLOCK FILES**, and the vertical profile is evaluated according to the similarity theory. 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 **DISGAS** and parameters read from the file **winds.dat** are used as input for **DIAGNO** (see Sec. 2.1).
- **HORIZONTAL_TURB_MODEL**: Flag indicating the model for the horizontal diffusion coefficients. This may be equal to **CONSTANT** for constant horizontal diffusion coefficients or **SMAGORINSKY** for horizontal diffusion coefficient from the Smagorinsky model (see eq. (12)).
- **VERTICAL_TURB_MODEL**: Flag indicating the model for the vertical diffusion coefficient. This may be set to 0 (zero) for constant vertical diffusion coefficient or 1 (one) for vertical diffusion coefficient from Atmospheric Boundary Level (ABL) model (see eq. (7)).
- **ROUGHNESS_MODEL**: Flag indicating the soil roughness length model. Possibilities are **UNIFORM** (set with the parameter **ROUGHNESS_LENGTH**) or **MATRIX** for non-uniform roughness length, provided with a file (see **BLOCK FILES**).
- **ROUGHNESS_LENGTH**: 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.
- **DIFF_COEFF_HORIZONTAL**: Horizontal diffusion coefficient in m^2/s , used if **HORIZONTAL_TURB_MODEL** is equal to **CONSTANT**.

- `DIFF_COEFF_VERTICAL`: Vertical diffusion coefficient in m^2/s , used if `VERTICAL_TURB_MODEL` is equal to `CONSTANT`.
- `MIN_DIFF_COEFF_HORIZONTAL`: Optional parameter (default $1 \text{ m}^2/\text{s}$). Minimum horizontal diffusion coefficient in m^2/s , used if `HORIZONTAL_TURB_MODEL` is equal to `SMAGORINSKY`.
- `MIN_DIFF_COEFF_VERTICAL`: Optional parameter (default $1 \text{ m}^2/\text{s}$). Minimum vertical diffusion coefficient in m^2/s , used if `VERTICAL_TURB_MODEL` is equal to `SIMILARITY`.

B.1.6 BLOCK FILES

- `TOPOGRAPHY_FILE_PATH`: Name (including relative or absolute path) of the topography file. See Sec. B.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 file containing the roughness. See Sec. B.4 for file format details. This record is read only when `ROUGHNESS_MODEL = MATRIX`.
- `RESTART_FILE_PATH`: Name (including relative or absolute path) of the restart file. This file is automatically generated and overwritten by `disgas` at every print time. It may be later used to restart the simulation from the last dump. See Sec. B.1.1 and B.9.
- `SOURCE_FILE_PATH`: Name (including relative or absolute path) of the source file. See section B.3 for file format details.
- `WIND_FILE_PATH`: Name (including relative or absolute path) of the wind data file. See section B.5 for file format details.
- `DIAGNO_FILE_PATH`: Name (including relative or absolute path) of the `DIAGNO` output file. This is a binary unformatted file created by `DIAGNO`. Only used when `WIND_MODEL = DIAGNO`.
- `OUTPUT_DIRECTORY`: Name (including relative or absolute path) of the folder where `DISGAS` output files are written.

B.1.7 BLOCK OUTPUT

- `LOG_VERBOSITY_LEVEL`: Flag indicating verbosity level of the log file. Possibilities are 0, for low verbosity log file, and 1, for high verbosity log file.
- `OUTPUT_GRD_TYPE`: Format for the GRD file. Possibilities are `ASCII` or `BINARY` (in the current version `ASCII` only).
- `OUTPUT_INTERVAL_(SEC)`: Time interval to output results (in s).
- `OUTPUT_U_VELOCITY`: Flag indicating whether the wind velocity in the x direction (in m/s) has to be printed 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` recorded every `OUTPUT_INTERVAL_(SEC)` seconds.
- `OUTPUT_V_VELOCITY`: Flag indicating whether the wind velocity in the y direction (in m/s) has to be printed 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` recorded every `OUTPUT_INTERVAL_(SEC)` seconds.
- `OUTPUT_CONCENTRATION`: Flag indicating whether the gas concentration (in kg/m^3) has to be printed 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_LAYER`: Flag indicating the number of printed NZ layer. Possibilities are `ALL` or `list of layers`.

- `TRACK_POINTS`: Flag indicating the presence of tracking points. Possibilities are YES or NO.
- `N_POINTS`: Flag indicating the number of tracking points.
- `POINTS_EASTING`: Flag indicating the Easting coordinates (UTM) of tracking points.
- `POINTS_NORTHING`: Flag indicating the Northing coordinates (UTM) of tracking points.
- `POINTS_ELEVATION`: Flag indicating the elevation (m) of tracking points.

B.2 The topography file `topography.grd`

The topography file specifies ground elevation at a regional scale (*i.e.* in a region typically equal or larger than the computational domain). Topography must be specified on a structured grid using arbitrary (but constant) grid spacing (*e.g.*, 5 m, 10 m, 100 m, *etc.*). Discretizations along x - and y -directions (grid spacing) can be different. The only necessary requirement is that the computational domain must lay within the bounds of the region where topography is specified. DISGAS reads the topography file and automatically interpolates elevations onto the nodes of the computational grid. In this version of `disgas`, the topography must be stored in the ASCII GRD format. The structure of a GRD format file is described in Sec. C (see also Table 4).

B.3 The source file `source.dat`

The source file specifies the gas fluxes (mass flow rate expressed in kg/s) from different point sources. DISGAS reads this file and automatically locates the source onto the nearest node of the computational domain. In case multiple records are assigned the same node, the mass fluxes for that node are summed. Each concentrated point source is listed in the source file with the following record structure:

- `X_s` : x -coordinate (UTM in m) of the source.
- `Y_s` : y -coordinate (UTM in m) of the source.
- `PHI_s`: Flux associated to the point source, expressed in kg/s.

There is no limit to the number of point sources; point sources located outside the computational domain are automatically rejected.

B.4 The roughness length file `roughness.grd`

This file specifies the roughness-length in a region typically equal or larger than the computational domain. The roughness length (in m) must be specified on a structured grid using arbitrary (but constant) grid spacing (*e.g.*, 5 m, 10 m, 100 m, *etc.*). Discretizations along x - and y -directions (grid spacing) can be different. The only necessary requirement is that the computational domain must lay within the bounds of the region where roughness length is specified. DISGAS reads this file and automatically interpolates the values onto the nodes of the computational grid. In this version of `disgas`, the roughness length must be stored in the ASCII GRD format. The structure of the roughness length file is the same as the `topography.grd` file, and is described in Sec. C (see also Table 4).

B.5 The wind data file `winds.dat`

The wind data file contains meteorological data at different time slices. If the record `WIND_MODEL`, in the control input file, is `UNIFORM`, DISGAS reads this wind file and estimates the Atmospheric Surface Layer parameters. Otherwise, if the record `WIND_MODEL` is `DIAGNO`, the wind field is instead read by the `DIAGNO` output file (file `diagno.out`). The wind file format is described in Table 2 (if `code=CUP`) and Table 3 (if `code=SONIC`) and the meaning of the used symbols is the following:

- `X_UTM`, `Y_UTM` : UTM coordinates of the wind station (specified in the first record of the file).

- `Z_ref` : Elevation above ground of the wind station (specified in the first record of the file).
- `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), measured at `Z_ref`.
- `p` : Atmospheric pressure (in hPa).
- `ustar` : Friction velocity (in m/s).
- `L` : Monin-Obukhov length (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 coincide in both cases.

B.6 The DIAGNO file `diagno.out`

The 3-D time-dependent wind field produced by the utility DIAGNO is stored in the unformatted file `diagno.out`. The format of this file is defined by subroutine `outfil.f` of `diagno-1.1.6` (distributed separately). This subroutine was modified with respect to the original contained in the utility `diagno` written by Douglas and Kessler (1990). The format of file `diagno.out` is not described here since it cannot be edited (see the user guide of `diagno`).

B.7 The DISGAS output files

At each user specified time (see `OUTPUT_INTERVAL_(SEC)` record in the control file) DISGAS generates 2D matrices written in GRD-format for the gas concentration, each corresponding to a different Z -layer (e.g. `c_00i_00000j.grd` where i denotes the layer and j the time slice). Moreover, `disgas` can produce GRD files containing the x and y components of the wind velocity at each Z -layer (e.g. `u_00i_00000j.grd` where i denotes the layer and j the time slice).

The output files are stored in the directory defined by the `OUTPUT_DIRECTORY` record of the DISGAS control file. Files in GRD format can be read by several plotting programs like the commercial software Surfer.

B.8 The `problemname.log` file

The name of this file is passed with the program call argument. It contains information concerning the run (summary of input data, run time error messages, CPU time, *etc.*).

B.9 The restart file `restart.dat`

The restart file is automatically produced by `disgas` and is overwritten at each print step. This file contains all the information needed to restart the simulation starting from the time of last dump. This may be useful for long runs that need to be interrupted and then restart, or for modifying some parameters (*e.g.* the wind field or the source file) during runtime. The restart file is written in binary and cannot be directly edited. The format of the restart file is defined in the subroutines `wrist.f90` (which generates the restart file) and `setvar.f90` (which reads the restart file), both located in the directory `src`.

Appendix C Format of the GRD files

The GRD files follow the Golden Surfer structure, as described in Table 4. The meaning of the used symbols is the following:

- **ID** : Four-characters identification string. The string DSAA identifies an ASCII grid file, whereas the string DSBB identifies a BINARY file.
- **NX** : Number of grid points along x -direction
- **NY** : Number of grid points along y -direction
- **XMIN** : x -coordinate of the grid left corners
- **XMAX** : x -coordinate of the grid right corners
- **YMIN** : y -coordinate of the grid bottom corners
- **YMAX** : y -coordinate of the grid top corners
- **ZMIN** : Minimum value in the grid
- **ZMAX** : Maximum value in the grid
- **MAT** : Matrix values at each grid point. It consists of an array of $NX \times NY$ values organized in row order. The pixels start from the bottom-left corner and move right along the row. Rows are written from bottom to top. Last value refers to the top-right corner.

In the BINARY files **NX** and **NY** are `INTEGER(2)`, **XMIN**, **XMAX**, **YMIN**, **YMAX**, **ZMIN** and **ZMAX** are `REAL(8)`, and the values of **MAT** are `REAL(4)`.

Table 1: Sample of the input control file `problemname.inp`

```

DISGAS INPUT FILE
TIME
  YEAR   = 2007
  MONTH  = 2
  DAY    = 24
  HOUR   = 12
  MINUTE = 00
  SIMULATION_INTERVAL_(SEC) = 120
  RESTART_RUN = NO
  RESET_TIME  = NO
GRID
  NX     = 40
  NY     = 32
  NZ     = 11
  Z_LAYERS_(M) = 0. 1. 4. 8. 12. 16. 24. 32. 40. 50. 60.
  DX_(M) = 50.
  DY_(M) = 50.
  X_ORIGIN_(UTM_M) = 427000
  Y_ORIGIN_(UTM_M) = 4520000
PROPERTIES
  DISPERSION_TYPE = PARTICLES           (GAS/PARTICLES)
  PARTICLE_DIAMETER = 2.0e-6           (PARTICLE DIAMETER IN M)
  PARTICLE_DENSITY = 2610.             (PARTICLE DENSITY KG/M3)
  PARTICLE_SHAPE_PARAMETER=0.779      (PARTICLE SPHERICITY)
  PARTICLE_MODEL_VSET = GANSER         (ARASTOPOUR/GANSER/WILSON_HUANG)
TOPOGRAPHY
  EXTRACT_TOPOGRAPHY_FROM_FILE = YES   (YES/NO)
  Z_ORIGIN_(M) = 0.0
  X_SLOPE_(DEG) = 0.0
  Y_SLOPE_(DEG) = 0.0
METEO
  WIND_MODEL           = DIAGNO         (SIMILARITY/DIAGNO)
  HORIZONTAL_TURB_MODEL = SMAGORINSKY   (CONSTANT/SMAGORINSKY)
  VERTICAL_TURB_MODEL  = SIMILARITY     (CONSTANT/SIMILARITY)
  ROUGHNESS_MODEL      = UNIFORM        (UNIFORM/MATRIX)
  ROUGHNESS_LENGTH     = 0.18           (Roughness length if UNIFORM)
  DIFF_COEFF_HORIZONTAL = 10.           (Only for turbulence model=0)
  DIFF_COEFF_VERTICAL  = 10.           (Only for turbulence model=0)
FILES
  TOPOGRAPHY_FILE_PATH = topography.grd (Relative or absolute file paths)
  ROUGHNESS_FILE_PATH  = roughness.grd
  RESTART_FILE_PATH    = restart.dat
  SOURCE_FILE_PATH     = source.dat
  WIND_FILE_PATH       = winds.dat
  DIAGNO_FILE_PATH     = diagno.out
  OUTPUT_DIRECTORY     = outfiles
OUTPUT
  LOG_VERBOSITY_LEVEL = 0               (0/1)
  OUTPUT_GRD_TYPE     = ASCII            (ASCII/BINARY)
  OUTPUT_INTERVAL_(SEC) = 21600         (Number of seconds between prints)
  OUTPUT_U_VELOCITY   = NO              (YES/NO)
  OUTPUT_V_VELOCITY   = NO              (YES/NO)
  OUTPUT_W_VELOCITY   = NO              (YES/NO)
  OUTPUT_CONCENTRATION = YES            (YES/NO)
  OUTPUT_LAYERS       = ALL              (ALL/List_of_layers)
  TRACK_POINTS        = NO              (YES/NO)
  N_POINTS            = 2                (Number of tracking points)
  POINTS_EASTING      = 643080 641000   (UTM easting coordinate of each tracking point)
  POINTS_NORTHING     = 1774150 1774250 (UTM northing coordinate of each tracking point)
  POINTS_ELEVATION    = 1 1             (Meters above ground of each tracking point)

```

Table 2: Format of the wind file `winds.dat` with `code=CUP`.

X_UTM	Y_UTM	Zref				
iyр	imo	idy	ihr	imi	code	# 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 3: Format of the wind file `winds.dat` with `code=SONIC`.

X_UTM	Y_UTM	Zref				
iyр	imo	idy	ihr	imi	code	# code=SONIC
t_1	t_2	wx	wy	T_zref	ustar	L
...
t_(nt-1)	t_nt	wx	wy	T_zref	ustar	L

Table 4: Format of a GRD file

ID
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

References

- Arastoopour, H., Lin, S.C. and Weil, S.A. (1982). Analysis of vertical pneumatic conveying of solids using multiphase flow models, *AIChE J.*, 28(3), 467–473, <https://doi.org/10.1002/aic.690280315>.
- Britter, R.E. and McQuaid, J.D. (1988). Workbook on the dispersion of dense gases, Technical report, HSE Contract Research Report No. 17/1988, Trinity Road Bootle, Merseyside L20 3QY, UK.
- Businger, J.A., Wyngaard, J.C., Izumi, Y. and Bradley, E.F. (1971). Flux-profile relationships in the atmospheric surface layer, *J. Atmos. Sci.*, 28(2), 181–189, [https://doi.org/10.1175/1520-0469\(1971\)028%3C0181:FPRITA%3E2.0.CO;2](https://doi.org/10.1175/1520-0469(1971)028%3C0181:FPRITA%3E2.0.CO;2).
- Byun, D. and Schere, K.L. (2006). Review of the governing equations, computational algorithms, and other components of the Models-3 Community Multiscale Air Quality (CMAQ) modeling system, *Applied Mechanics Reviews*, 59(2), 51–77, <https://doi.org/10.1115/1.2128636>.
- Byun, D.W. (1990). On the analytical solutions of flux-profile relationships for the atmospheric surface layer, *J. Appl. Met.*, 29(7), 652–657.
- Byun, D.W. and Chin, J.K.S. (1999). Science Algorithms of the EPA Model-3 Community Multiscale Air Quality (CMAQ) Modeling System, ed. Byun, D. W. and Chin, J. K. S. Volume III, EPA-600/R-99-030, Washington, DC.
- Cerminara, M., Esposti Ongaro, T. and Berselli, L.C. (2016). ASHEE-1.0: a compressible, equilibrium-Eulerian model for volcanic ash plumes, *Geosci. Model Dev.*, 9, 967–730, <https://doi.org/10.5194/gmd-9-697-2016>.
- Chiodini, G., Baldini, A., Barberi, F., Carapezza, M., Cardellini, C., Frondini, F., Granieri, D. and Ranaldi, M. (2007). Carbon dioxide degassing at Latera caldera (Italy): Evidence of geothermal reservoir and evaluation of its potential energy, *J. Geophys. Res.*, 112, B12204, <https://doi.org/10.1029/2006JB004896>.
- Chiodini, G., Frondini, F., Cardellini, C., Granieri, D., Marini, L. and Ventura, G. (2001). CO_2 degassing and energy release at Solfatara volcano, Campi Flegrei, Italy, *J. Geophys. Res.*, 106, 16213–16221, <https://doi.org/10.1029/2001JB000246>.
- Chiodini, G., Granieri, D., Avino, R., Caliro, S., Costa, A., Minopoli, C. and Vilardo, G. (2010). Non-volcanic CO_2 Earth degassing: The case of Mefite di Ansanto (Southern Apennines), Italy, *Geophys. Res. Lett.*, 37(L11303), <https://doi.org/10.1029/2010GL042858>.
- Chiodini, G., Granieri, D., Avino, R., Caliro, S., Costa, A. and Werner, C. (2005). Carbon dioxide diffuse degassing: implications on the energetic state of a volcanic hydrothermal systems, *J. Geophys. Res.*, 110, B08204, <https://doi.org/10.1029/2004JB003542>.
- Cortis, A. and Oldenburg, C.M. (2009). Short-range atmospheric dispersion of carbon dioxide, *Bound-Lay. Meteorol.*, 133(1), 17–34, <https://doi.org/10.1007/s10546-009-9418-y>.
- Costa, A. (2006). Permeability-porosity relationship: a re-examination of the Kozeny-Carman equation based on fractal pore-space geometry, *Geophys. Res. Lett.*, 33(L02318), <https://doi.org/10.1029/2005GL025134>.
- Costa, A., Chiodini, G., Granieri, D., Folch, A., Hankin, R.K.S., Caliro, S., Avino, R. and Cardellini, C. (2008). A shallow layer model for heavy gas dispersion from natural sources: application and hazard assessment at Caldara di Manziiana, Italy, *Geochem. Geophys. Geosyst.*, 9(Q03002), <https://doi.org/10.1029/2007GC001762>.
- Costa, A., Folch, A. and Macedonio, G. (2013). Density-driven transport in the umbrella region of volcanic clouds: Implications for tephra dispersion models, *Geophys. Res. Lett.*, 40, 1–5, <https://doi.org/10.1002/grl.50942>, corrected on 17 June 2019.

- Costa, A. and Macedonio, G. (2016). DISGAS-2.0: A model for passive DISpersion of GAS, Rapporti Tecnici 332, Istituto Nazionale di Geofisica e Vulcanologia, Italy, <https://doi.org/10.13127/rpt/332>.
- Costa, A., Macedonio, G. and Chiodini, G. (2005). Numerical model of gas dispersion emitted from volcanic sources, *Ann. Geophys-Italy*, 48, 805–815.
- De Bruin, H.A.R., Ronda, R.J. and Van de Wiel, B.J.H. (2000). Approximate solutions for the Obukhov length and the surface fluxes in terms of Bulk Richardson Numbers, *Bound-Lay. Meteorol.*, 95, 145–157.
- Douglas, S. and Kessler, R.C. (1990). User's Manual for the Diagnostic Wind Model, ed. Carr, L. Volume III, EPA-450/4-90-007C, San Rafael, CA.
- Dutton, J.A. and Fichtl, G.H. (1969). Approximate equations of motion for gases and liquids, *J. Atm. Sci.*, 26, 241–254.
- Dyer, A.J. (1974). A review of flux-profile relationships, *Bound-Lay. Meteorol.*, 7, 363–372, <https://doi.org/10.1007/BF00240838>.
- Ewing, R.E. and Wang, H. (2001). A summary of numerical methods for time-dependent advection-dominated partial differential equations, *J. Comput. Appl. Math.*, 128(1-2), 423–445, [https://doi.org/10.1016/S0377-0427\(00\)00522-7](https://doi.org/10.1016/S0377-0427(00)00522-7).
- Folch, A., Costa, A. and Hankin, R.K.S. (2009). Twodee-2: A shallow layer model for dense gas dispersion on complex topography, *Comput. Geosci.*, 35(3), 667–674, <https://doi.org/10.1016/j.cageo.2007.12.017>.
- Ganser, G.H. (1993). A rational approach to drag prediction of spherical and nonspherical particles, *Powder Technol.*, 77(2), 143–152, [https://doi.org/10.1016/0032-5910\(93\)80051-B](https://doi.org/10.1016/0032-5910(93)80051-B).
- Golder, D. (1972). Relations among stability parameters in the surface layer, *Bound-Lay. Meteorol.*, 3, 47–58.
- Granieri, D., Costa, A., Macedonio, G., Chiodini, G. and Bisson, M. (2013). Carbon dioxide in the city of Naples: contribution and effects of the volcanic source, *J. Volcanol. Geotherm. Res.*, 260, 52–61, <https://doi.org/10.1016/j.jvolgeores.2013.05.003>.
- Hankin, R.K.S. and Britter, R.E. (1999). 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.
- Hindmarsh, A.C., Gresho, P. and Griffiths, D.F. (1984). The stability of explicit Euler integration for certain finite difference approximations of the multidimensional advection-diffusion equation, *Int. J. Numer. Methods Fluids*, 4, 853–897.
- Jacobson, M.Z. (1999). *Fundamentals of atmospheric modelling*, Cambridge University Press, New York, first edition.
- Jacobson, M.Z., Lu, R., Turco, R.P. and Toon, O.B. (1996). Development and application of a new air pollution modeling system. Part I: gas-phase simulations, *Atmos. Environ.*, 30, 1939–1963.
- Louis, J.F. (1979). A parametric model of vertical eddy fluxes in the atmosphere, *Bound-Lay. Meteorol.*, 17, 187–202, <https://doi.org/10.1007/BF00117978>.
- Lu, W., Zhu, B., Yan, S., Li, J. and Wang, Z. (2023). Parameterized minimum eddy diffusivity in WRF-Chem(v3.9.1.1) for improving PM2.5 simulation in the stable boundary layer over eastern China, *Preprint, Egusphere*, Discussion, <https://doi.org/10.5194/egusphere-2023-1089>.
- Macedonio, G. and Costa, A. (2002). Finite Element modeling of gas dispersion in the atmosphere, in: A. Buccianti, L. Marini, G. Ottonello and O. Vaselli (editors), *Proceedings of the Arezzo Seminar in Fluids Geochemistry*, pages 147–159, Pacini Editore, Ospedaletto (Pisa) - Italy.

- Monin, A.S. and Yaglom, A.M. (1979). *Statistical Fluid Mechanics: Mechanics of Turbulence*, The MIT Press, USA, volumes 1 and 2.
- Park, S.U. and Kim, C.H. (1999). A numerical model for the simulation of so_2 concentrations in the Kyongin region, Korea, *Atmos. Environ.*, 33, 3119–3132.
- Pielke, R.A., Cotton, W.R., Walko, R.L., Tremback, C.J., Nicholls, M.E., Moran, M.D., Wesley, D.A., Lee, T.J. and Copeland, J.H. (1992). A comprehensive meteorological modeling system-RAMS, *Meteor. Atmos. Phys.*, 49, 69–91.
- Prabha, T.V. and Mursch-Radlgruber, E. (1999a). Investigation of air pollution distribution in Linz: case studies to evaluate a K-type diffusion model coupled with a mass-consistent wind model, *Atmos. Environ.*, 33, 4067–4080.
- Prabha, T.V. and Mursch-Radlgruber, E. (1999b). Modeling of diffusion in a wide Alpine Valley, *Theor. Appl. Climatol.*, 64, 93–103.
- Sagaut, P. (2006). *Large Eddy Simulation for incompressible flows: an introduction*, Scientific Computing, Springer, Heidelberg, Germany, 3rd edition.
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations, part I: the basic experiment, *Mon. Weather Rev.*, 91(3), 99–164.
- Sweeby, P.K. (1984). High resolution schemes using flux limiter for hyperbolic conservation laws, *SIAM J. Numer. Anal.*, 21(5), 995–1011.
- Toon, O.B., Turco, R.P., D., W., Malone, R. and Liu, M. (1988). A multidimensional model for aerosols: Description of computer analogs, *J. Atm. Sci.*, 45, 2123–2143.
- Wang, Y. and Hutter, K. (2001). Comparisons of numerical methods with respect to convectively dominated problems, *Int. J. Numer. Methods Fluids*, 37, 721–745, <https://doi.org/10.1002/flid.197>.
- Wilson, L. and Huang, T.C. (1979). The influence of shape on the atmospheric settling velocity of volcanic ash particles, *Earth Planet. Sci. Lett.*, 44(2), 311–324, [https://doi.org/10.1016/0012-821X\(79\)90179-1](https://doi.org/10.1016/0012-821X(79)90179-1).