



D4.1 Explosive risk and fire module

Ludovic Lepers, Laurent Aprin, Sébastien Legrand



ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

DISCLAIMER

The content of this document represents the views of the author only and is his/her sole responsibility; it cannot be considered to reflect the views of the European Commission and/or the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) or any other body of the European Union. The European Commission and the DG-ECHO is not responsible for any use that may be made of the information it contains.





Project Acronym	MANIFESTS
Project Full Title	MANaging risks and Impacts From Evaporating and gaseous Substances To population Safety
Gant Agreement Nr.	101004912
Project Website	https://www.manifests-project.eu/

Deliverable Nr.	D4.1
Status (Final/Draft/Revised)	Final
Work Package	4
Task Number	4.1
Responsible Institute	RBINS
Author/s	Ludovic Lepers, Laurent Aprin, Sébastien Legrand
Recommended Citation	Lepers L., L. Aprin and S. Legrand (2021) 'D4.1 - Fire and explosion risk module', MANIFESTS project, Technical Report, 34pp.
Dissemination Level	Public

Document History			
Version	Date	Modification Introduced	
		Modification Reason	Modified by
1	22/12/2021		

Contents

1. Introduction.....	7
2. Fire module	9
2.1. Parametrization	9
2.1.1. Estimation of the energy flux at the fire	10
2.1.2. Absorption due to the atmosphere.....	12
2.1.3. Form factor	12
2.1.4. Flux received at a distance from the fire.....	15
2.2. Code description	15
2.2.1. Structure.....	15
2.2.2. Inputs.....	17
2.2.1. Outputs.....	17
2.3. Validation	18
2.4. How to run a simulation	19
2.5. Model limitation.....	19
3. Explosion module	20
3.1. Parametrization	20
3.1. Code description	21
3.1.1. Structure.....	21
3.1.2. Inputs.....	23
3.1.3. Outputs.....	23
3.1. Validation	24
3.2. How to run a simulation	25
3.3. Model limitation.....	25
4. Conclusion	26
5. References.....	27
6. Annexes	28

6.1. Fire module input file	28
6.2. Explosion module input file.....	33

1. Introduction

Responding to maritime accidents can be extremely challenging when involving HNS that behave as evaporators. Due to their potential to form toxic or combustible clouds, evidence-based decisions are needed to protect the crew, responders, the coastal population and the environment. However, when an emergency is declared, key information is not always available for all the needs of responders. A case in point is the lack of knowledge and data to assess the risks that responders or rescue teams could take when intervening, or those that could impact coastal communities when allowing a shipping casualty to dock at a place of refuge.

The MANIFESTS project aims to address these uncertainties and improve response and training capacities through the development of an operational decision-support system (DSS) for volatile HNS spills. Besides management and communication, the project includes four other work packages: WP2 on collecting new data on evaporators, WP3 on table top exercises and field training, WP4 on improving modelling tools and WP5 on the development of the DSS. Key expected outcomes include:

- Operational guidance;
- Desktop and field exercises;
- In situ training;
- Experimental data on gas cloud fate;
- A brand-new fire and explosion modelling module;
- Improved HNS database with new experimental data on evaporation/dissolution kinetics.

This report presents the results obtained in the framework of the task 4.1 aiming at developing tools that would help responders to assess risks in case of explosion and of fire of volatile HNS.

The fire module computes the energy flux as a function of the distance to the fire source. It is useful to assess the safety distance at which e.g. a boat can approach a fire while keeping the crew safe. The energy flux can cause burning to people, and start new fire. The burning rate is also estimated.

The explosion module computes the overpressure of the shockwave caused by the combustion of a chemical. This overpressure can be very dangerous for people and structure, causing wounds from minor injury to death and destruction of building. The model could be used to predict what could happen in case of the explosion of a stored explosive for instance.

The two models are simplifications of the reality and do not take everything into account. Their results can be useful to have a rough idea of what could happen in open sea but should always be interpreted keeping the model hypotheses and limitations in mind.

Due to the sensitivity of the topic, the source code of both modules is not made available to public.

2. Fire module

The goal of this module is to model the impact of a fire on people and infrastructures. It computes the energy flux as a function of distance. The fire module enables to establish a zone in which human will be at risk to develop burning in case of a fire ignition. It gives also an estimation of the burning rate that can be useful to assess, the duration of the fire. However, this module does not model all the complex processes occurring inside a fire, and should only be used to have a rough estimation of safety distances and fire duration.

A fire is a phenomenon in which a chemical oxidises another one quickly and releases heat, in many case oxygen oxidizing a fuel. This phenomenon requires energy to start but is then able to maintain itself thanks to the energy released.

The fire module is divided into three parts: the estimation of the energy flux at the fire, the estimation of the atmospheric absorption and the effect due to the geometry. These three factors are then multiplied together to retrieve the flux at a given distance. These points are described in details in the next section.

The parametrization provides a flux as a function of distance. An iterative solver has been implemented to compute the safety distance for predefined threshold values of the flux. To follow local or national legislation, users may specify these threshold values in the input configuration file (cf. section 2.2.2 and annex 6.1).

2.1. Parametrization

The fire module can use multiple parametrization to obtain the result. Some needs more parameters than others, and some values can be provided (for instance the burning rate) and can help to increase the accuracy of the model.

The first step is the computation of the equivalent diameter:

$$D_{eq} = 4 \times \frac{area}{perimeter} \quad (1)$$

D_{eq} : equivalent diameter [m]

$area$: area of the fire [m²]

$perimeter$: perimeter of the fire [m]

The equivalent radius is the equivalent diameter divided by two. If the length of the slick is bigger in one direction than in the other one (more than two times), it is better to take the smallest length.

2.1.1. Estimation of the energy flux at the fire

This section describes the way the energy flux ϕ_0 emitted by the fire at the source is computed as a function of the mass flow of combustion, the flame height and the fire power.

The flow of combustion can be computed as follows (Burgess and Hertzberg, 1974), for a slick with a diameter large enough:

$$m'' = 10^{-3} \times \frac{\Delta H_c}{\Delta H_v + \int_{T_0}^{T_b} C_p dT} \quad (2)$$

m'' : mass flow of combustion [$\text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$]

ΔH_c : combustion enthalpy [$\text{kJ} \cdot \text{kg}^{-1}$]

ΔH_v : vaporization enthalpy at the combustion temperature [$\text{kJ} \cdot \text{kg}^{-1}$]

T_0 : ambient temperature [K]

T_b : ebullition temperature [K]

C_p : specific heat [$\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

If we assume the specific heat is constant between ambient temperature and ebullition temperature, equation (2) becomes:

$$m'' = 10^{-3} \times \frac{\Delta H_c}{\Delta H_v + C_p \times (T_b - T_0)} \quad (3)$$

From that, the power can be computed too:

$$Q = \eta_{comb} \times m'' \times area \times \Delta H_c \quad (4)$$

Q : power [$\text{kJ} \cdot \text{s}^{-1}$]

η_{comb} : yield of the combustion [] (mean value at 0.7(Drysdale, 1985))

The height of the flames can be computed in two ways. The first one (Heskestad, 1984) is:

$$L_f = -1.02D_{eq} + 0.235Q^{\frac{2}{5}} \quad (5)$$

L_f : flame height [m]

The second one (Thomas, 1963) is an empirical correlation from fire of wood. It takes into account the wind:

For a wind which is less than $1 \text{ m}\cdot\text{s}^{-1}$:

$$L_f = 42 \times D_{eq} \times \left(\frac{\dot{m}''}{\rho_{air} \times \sqrt{g \times D_{eq}}} \right)^{0,61} \quad (6)$$

ρ_{air} : air density [$\text{kg}\cdot\text{m}^{-3}$]

g : acceleration due to gravity on earth [$\text{m}\cdot\text{s}^{-2}$]

And if the wind is more than 1 m/s:

$$L_f = 55 \times D_{eq} \times \left(\frac{\dot{m}''}{\rho_{air} \times \sqrt{g \times D_{eq}}} \right)^{0,67} \times (u^*)^{-0,21} \quad (7)$$

with

$$u^* = \frac{u_w}{\sqrt[3]{\frac{g \times \dot{m}'' \times D_{eq}}{\rho_{air}}}} \quad (8)$$

u_w : wind speed at 10 meter [$\text{m}\cdot\text{s}^{-1}$]

With the flame height, it is possible to compute the emission in three ways:

The first one ("Instruction Technique du 9 novembre 1989 - dépôts aériens existants de liquides inflammables," 1989) uses a constant flux:

$$\phi_0 = 29.720 \quad (9)$$

ϕ_0 : flux at the base of the fire [$\text{kW}\cdot\text{m}^{-2}$]

The second one (Mudan and Croce, 1995) is a function of the equivalent diameter:

$$\phi_0 = 20 + 120 e^{-0,12 \cdot D_{eq}} \quad (10)$$

The third one (TNO-Yellow Book-Methods for the calculations of physicals effects CPR14E, 1997) uses more parameters:

$$\phi_{max} = \dot{m}'' \times FR \times \frac{\Delta H_c}{1 + 4 \cdot \frac{L_f}{D_{eq}}} \quad (11)$$

$$\phi_0 = \phi_{max} \times (1 - \xi) + \phi_{soot} \times \xi \quad (12)$$

ϕ_{max} : emission of the flame without black smokes [kW.m⁻²]

FR : radiative fraction (between 0.1 and 0.4)[]

ξ : amount of the flux absorbed by the smokes (0.8) []

ϕ_{soot} : emissive power of the smokes (20 for black smokes at 800K) [kW.m⁻²]

The different parametrizations (2) to (12) have been implemented in the fire module. However, the default module configuration computes the emitted energy flux from TNO Yellow book (equations 11 and 12), using the flame height parametrization from Tomas (equations 6 to 8).

2.1.2. Absorption due to the atmosphere

Once the energy flux emitted at the source is known, one has to compute how much of it will be received at a distance from the fire. The amount of energy absorbed by the atmosphere as a function of distance can be computed (Brzustowski and Sommer, 1973):

$$\tau = 0,79 \times \left(\frac{100}{X}\right)^{\frac{1}{16}} \times \left(\frac{30,5}{RH}\right)^{\frac{1}{16}} \quad (13)$$

X : distance from the fire [m]

RH : relative humidity of the air (10-100)[]

To avoid unrealistic values, the empirical formula (13) is applied only for distance larger than 2 m and relative humidity larger than 10 %.

2.1.3. Form factor

The form factor computes how the energy flux decreases as function of the distance from the fire due to geometrical considerations (Raj, 1977).

$$\int_S \frac{\cos\theta_1 \times \cos\theta_2}{\pi \times r^2} dS \quad (14)$$

This is the general formula for computing the form factor, with S as the area, r the distance between the fire and the element of interest (where the flux needs to be known), θ_1 and θ_2 are the angle between the normal of the element of interest and the fire and the angle between the normal of the fire and the element of interest. This is shown in the Figure 1.

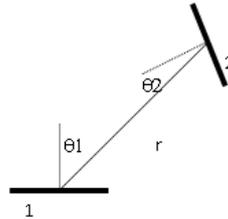


Figure 1: How the form factor is computed

In the fire module, we use a cylindrical view factor, “inclined” due to the wind, which depends on the parameters shown in the Figure 2.

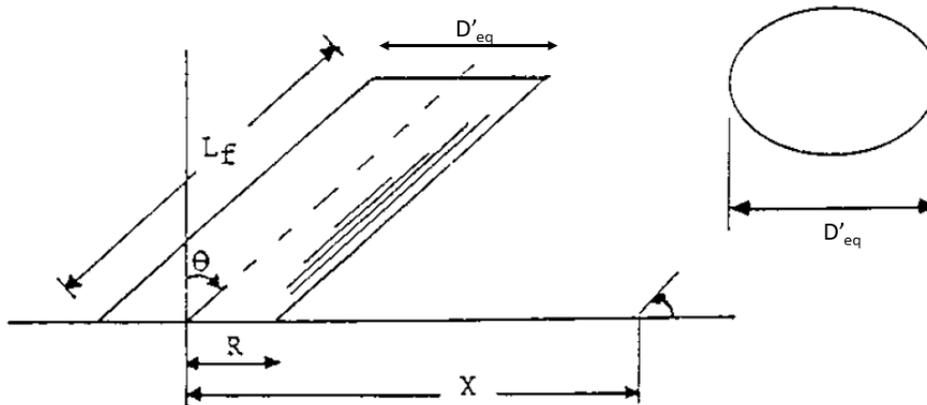


Figure 2: Form factor parameters

The equations used for computing the angle of the flame are:

$$Fr_{10} = \frac{u_w^2}{g \cdot D_{eq}} \tag{15}$$

$$Re = \frac{u_w \cdot D_{eq}}{\nu_{air}} \tag{16}$$

ν_{air} : air kinematic viscosity [$m^2 \cdot s^{-1}$]

$$c = 0,666 \cdot (Fr_{10})^{0,333} \cdot (Re)^{0,117} \tag{17}$$

$$\theta = \arcsin\left(\frac{\sqrt{4 \cdot c^2 + 1} - 1}{2 \cdot c}\right) \quad (18)$$

θ : angle of inclination of the fire due to the wind [rad]

When the angle is known, the form factor can be computed as follow:

$$D'_{eq} = 1,5 \cdot (Fr_{10})^{0,069} \cdot D_{eq} \quad (19)$$

$$R'_{eq} = \frac{D'_{eq}}{2} \quad (20)$$

$$a = \frac{L_f}{R'_{eq}} \quad (21)$$

$$b = \frac{X}{R'_{eq}} \quad (22)$$

$$A = \sqrt{(a^2 + (b + 1)^2 - 2 \cdot a \cdot (b + 1) \cdot \sin\theta)} \quad (23)$$

$$B = \sqrt{(a^2 + (b - 1)^2 - 2 \cdot a \cdot (b - 1) \cdot \sin\theta)} \quad (24)$$

$$C = \sqrt{(1 + (b^2 - 1) \cdot \cos^2\theta)} \quad (25)$$

$$D = \sqrt{\left(\frac{(b - 1)}{(b + 1)}\right)} \quad (26)$$

$$E = \frac{(a \cdot \cos\theta)}{(b - a \cdot \sin\theta)} \quad (27)$$

$$F = \sqrt{(b^2 - 1)} \quad (28)$$

$$F_v = \frac{1}{\pi} \left(-E \cdot \tan^{-1}(D) + E \left[\frac{a^2 + (b+1)^2 - 2 \cdot b \cdot (1 + a \cdot \sin(\theta))}{A \cdot B} \right] \cdot \tan^{-1} \left(\frac{A \cdot D}{B} \right) + \frac{\cos(\theta)}{C} \cdot \left[\tan^{-1} \left(\frac{a \cdot b - F^2 \cdot \sin(\theta)}{F \cdot C} \right) + \tan^{-1} \left(\frac{F \cdot \sin(\theta)}{C} \right) \right] \right) \quad (29)$$

$$F_h = \frac{1}{\pi} \left(\tan^{-1} \left(\frac{1}{D} \right) + \frac{\sin(\theta)}{C} \left[\tan^{-1} \left(\frac{a \cdot b - F^2 \cdot \sin(\theta)}{F \cdot C} \right) + \tan^{-1} \left(\frac{F \cdot \sin(\theta)}{C} \right) \right] - \left[\frac{a^2 + (b+1)^2 - 2 \cdot (b+1 + a \cdot b \cdot \sin(\theta))}{A \cdot B} \right] \cdot \tan^{-1} \left(\frac{A \cdot D}{B} \right) \right) \quad (30)$$

$$F_{form} = \sqrt{F_v^2 + F_h^2} \quad (31)$$

F_{form} : form factor []

2.1.4. Flux received at a distance from the fire

The energy flux received at given distance from the fire can be computed by the simple formula (TNO-Yellow Book-Methods for the calculations of physical effects CPR14E, 1997):

$$\phi = F_{form} \times \tau \times \phi_0 \quad (32)$$

ϕ : flux received [kW.m⁻²]

2.2. Code description

2.2.1. Structure

The code is written in python 3 with numpy and matplotlib as external modules. It is divided in two files: one named fire_utils.py and the other one run_fire.py. The “run” file contains all the

instructions for the execution of a simulation, from the reading of the input file to the creation of the output files. The “utils” file contains all the equations of the parametrizations implemented.

In the same directory as these files, there are three directories: “log”, “request” and “results”. The “log” directory contains one log file for each month (yyyy_mm.log) in which all the information about the simulations errors are written. The “request” directory contains all the json files ([...]_{sim_data}.json, with [...] the simulation ID) for launching a simulation. The model adds in the “results” directory a new folder each time a simulation is launched. It is named with the simulation ID and contains all the results files. For starting a simulation, it is needed to provide the simulation ID as argument. The folder structure is shown in Figure 6.

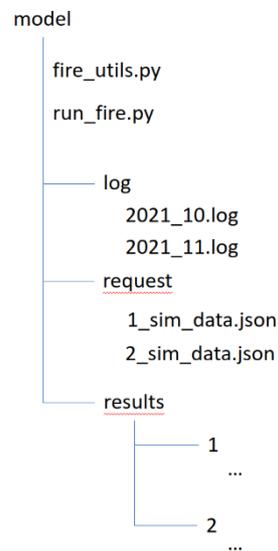


Figure 3 : Folder structure for the model, the “...” are the outputs files

When a simulation is launched, it follows the steps described here:

1. Initialization of the logging (if an error occurs here, it is not reported)
2. Reading the arguments (simulation ID)
3. Loading the json input file in the “request” folder
4. Reading the values of interest in the json, and verifying the units provided
5. Testing the range of some of the value provided (for instance, percent must be between 0 and 100)
6. Compute some values with the selected parametrization(s)
7. Generating plots and data array, and saving them

The parametrizations are used to compute a flux as a function of the distance from the fire. The code will compute a flux at a reference distance, and then it will try a new distance depending on the threshold chosen, if it is smaller or higher than the computed value. When the flux is close enough from the threshold, the code will stop to iterate and start again for the next value of the threshold.

If no error has been raised, the log file will now contains the line “INFO Simulation X is finished”.

2.2.2. Inputs

The model needs the simulation ID as argument in order to run. It will then use it to open a json file “simulation ID”_sim_data.json in the “request” directory. Annex 6.1 details the structure of this json file. It is organised around 4 data categories: environmental_data, release_data, pollutant_properties and advanced parameters. A description of the parameters is given in the Table 5, Annex 6.1. Each time a data is provided, its value should be given after “value:” and its units after “units:” (there are some exception in the advanced category). All data must be given to allow the model to run. Note that the values for the advanced parameters “mass”, “Deq”, “comb_flow”, “energy_flux”, “fl_height” and “fI0” should usually be set to “-1”, meaning the fire model will evaluate these parameters. This functionality has been implemented in order to constrain the model by some external values.

2.2.1. Outputs

The fire module produces three files, two plots shown in Figure 4 and Figure 5 and one csv file shown in Table 1 (simulation from a reference case).

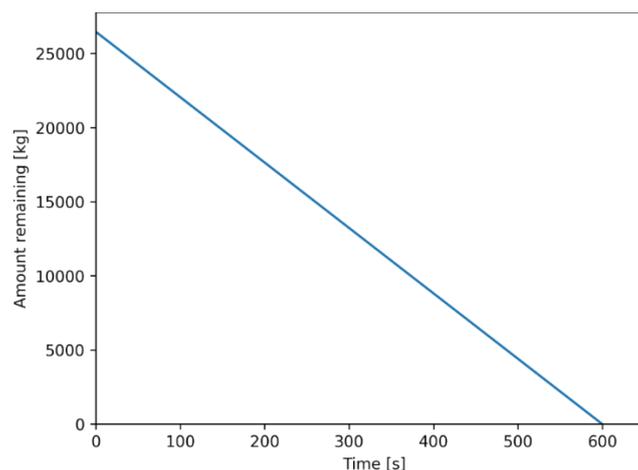


Figure 4 : Amount of combustible remaining as a function of time

Figure 4 gives information on the burning rate, considered as constant. The flux is calculated only at the start of the fire.

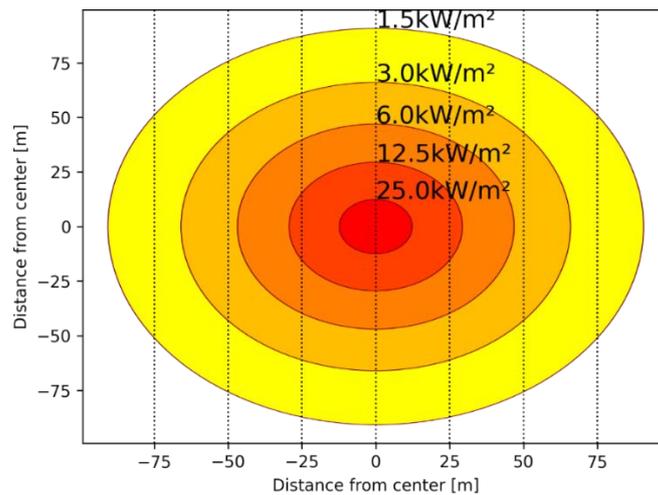


Figure 5 : Radiative flux as a function of the distance to the source

All the information presented graphically in Figure 5 is written in the csv provided in Table 1.

Table 1 : Distance for the threshold values of the radiative flux given in the input file

Flux [kW/m ²]	Distance [m]
1.5	90.2
3	65.5
6	46.4
12.5	28.9
25	11.9
53.1	0

It is possible to choose a flux and the model will give the distance at the source for receiving this radiative flux.

2.3. Validation

Some experimental data are needed here, CEDRE will perform experiments in 2022 and their results will be used to validate the model.

2.4. How to run a simulation

In order to start a simulation, the python files (`run_fire.py` and `fire_utils.py`) must be placed in the same folder, all the python dependencies (`numpy` and `matplotlib`) must be installed, and three folders “log”, “request”, “results” must have been created. A json file “X_sim_data.json” with the same structure as the one described in the Annexes must be present in the folder “request”. “X” is the ID of the simulation and must be an integer. To start the simulation, type “python `run_fire.py` X”. A folder “results/X” will be created in which all the results will be stored, and a new logfile will be created if there is not already one for the month.

2.5. Model limitation

As the model relies on several assumptions this necessarily implies limitations that are short-listed below:

- The burning rate of the fire is supposed to be constant, which could not be the case, for instance, when a certain amount of pollutant has already burnt. The flux will also change during the burning of the pollutant.
- The wind changes the shape of the fire, causing the downwind side to receive more energy, and the upwind side to receive less at the same distance from the fire. The model considers only one flux for one distance given, and so the flux on all the side of the fire is shown as if it was downwind, causing overestimation everywhere but downwind.
- The form factor considers a flat surface and that there is no obstacle between the fire and the target, if there are some obstacle, it can change the flux received.
- The specific heat is supposed to be constant, so it is assumed to be independent of the temperature.
- The parametrization of the absorption coefficient τ (eq 13) is not valid for short distances (<2m) and dry air condition (RH<10%).

3. Explosion module

Explosions are extremely fast combustion (see the introduction of the fire module for more information on the combustion process) inducing a new hazard: a wave of overpressure, moving in the air. This overpressure can be very dangerous for people and building, causing injury, death and destruction of structures. In order to release the energy fast enough to create this wave, the oxidizer and the fuel must be very well mixed (for instance a cloud of gaseous methane in air at a certain concentration). Some molecules are able to decompose themselves and do not need any other chemical to create an explosion if enough energy is provided to start the reaction (for instance TNT). Usually 3 types of explosion are considered:

- Flash fire: the flames front propagates with a subsonic velocity (less than $1\text{m}\cdot\text{s}^{-1}$). Overpressure is usually limited to a few millibars.
- Deflagration: the flames front propagates at the sonic speed, leading to moderate overpressure (several hundreds of millibars).
- Detonation: the flames front propagates at a supersonic speed, what generates a shock wave with overpressure superior to 1 bar.

The goal of the explosion module is not to simulate all the complex processes that occur during an explosion but to assess the magnitude of the overpressure as a function of the distance from an explosion. To this purpose, the model estimates the amount of energy released by the combustion of the given chemical. Then it estimates the TNT equivalent of the explosion, i.e. the amount of TNT needed to release the same amount of energy. Finally, thanks to an empirical parametrization, the model provides an estimation of the overpressure at a certain distance.

The parametrizations have been built to estimate the overpressure from a distance; they have been reversed for computing the distance at which the overpressure will reach a threshold. These thresholds are specified in the input json file, to follow the local legislation for instance.

3.1. Parametrization

The equivalent mass of TNT is the first parameter computed by the module (Frank P Lees, 2003; Marshall, 1976):

$$M_{TNT} = a \times \frac{m_{gas} \times \Delta H_c}{E_{TNT}} \quad (33)$$

M_{TNT} : TNT equivalent [kg]

a : yield of the combustion

m_{gas} : mass of explosive gas [kg]

ΔH_c : combustion enthalpy of the gas [kJ.kg⁻¹]

E_{TNT} : energy density released by TNT in an explosion (4690 kJ.kg⁻¹(Frank P Lees, 2003))

This relation are then used to define a “reduce distance” from the equivalent mass of TNT (the following relations have been realised from (Marshall, 1976)).

$$\lambda = \frac{R}{\sqrt[3]{M_{TNT}}} \quad (34)$$

R : distance from the explosion source [m]

λ : reduce distance [m.kg^{-1/3}]

The overpressure at a given distance is then computed in one of these three empirical equations depending on the reduce distance.

$$1.6 \leq \lambda < 4 : \Delta P = 14.89\lambda^{-2.28} \quad (35)$$

$$4 \leq \lambda < 10 : \Delta P = 7.05\lambda^{-1.72} \quad (36)$$

$$10 \leq \lambda < 63 : \Delta P = 1.55\lambda^{-1.06} \quad (37)$$

ΔP : overpressure [bar]

3.1. Code description

3.1.1. Structure

The explosion module is written in python 3 with numpy and matplotlib as external libraries. It has the same structure as the fire module. It is divided in two files: one named explosion_utils.py and the other one run_explosion.py. The “run” file contains all the instructions for the execution of a simulation, from the reading of the input file to the creation of the output files. The “utils” file contains all the equations of the parametrizations implemented.

In the same directory as these files, there are three directories: “log”, “request” and “results”. The “log” directory contains one log file for each month (yyyy_mm.log), in which all the information about the simulation errors is written. The “request” directory contains all the json files ([...]_sim_data.json, with [...] the simulation ID) for launching a simulation. The model adds in the “results” directory a new folder each time a simulation is launched. It is named with the simulation ID and contains all the results files. For starting a simulation, it is needed to provide the simulation ID as argument. The folder structure is shown in Figure 6.

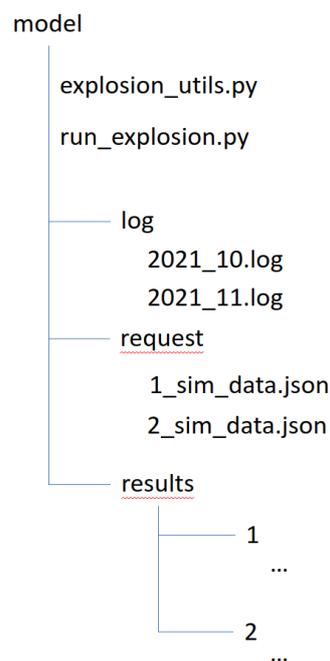


Figure 6 : folder structure for the model, the “...” are the outputs files

When a simulation is launched, it follows the steps described here:

1. Initialization of the logging (if an error occurs here, it is not reported)
2. Reading the arguments (simulation ID)
3. Loading the json input file in the “request” folder
4. Reading the values of interest in the json, and verifying the units provided
5. Testing the range of some of the value provided (for instance, percent must be between 0 and 100)
6. Compute some values with the parametrization chosen
7. Generating plots and data array, and saving them.

If no error has been raised, the line “INFO Simulation X is finished” is added to the log file.

3.1.2. Inputs

The model needs the simulation ID as argument in order to run. This ID refers to a json input file “simulation ID”_sim_data.json in the “request” directory. This json file contains 4 data categories: environmental_data, release_data, pollutant_properties and advanced. A description of the parameters is given in the Table 6, Annex 6.2. Each time a data is provided, its value should be given after “value:” keyword and its units after “units:” (there are some exception in the advanced category). All the data must be given for the model to run. Note that the values for the advanced parameters “mass” and “equ_TNT” should usually be set to “-1”, meaning the explosion model must evaluate these parameters. This functionality has been implemented in order to constrain the model by some external values.

3.1.3. Outputs

The model generates two files, a plot (Figure 7) and a csv (Table 2) which contains the same information as the plot.

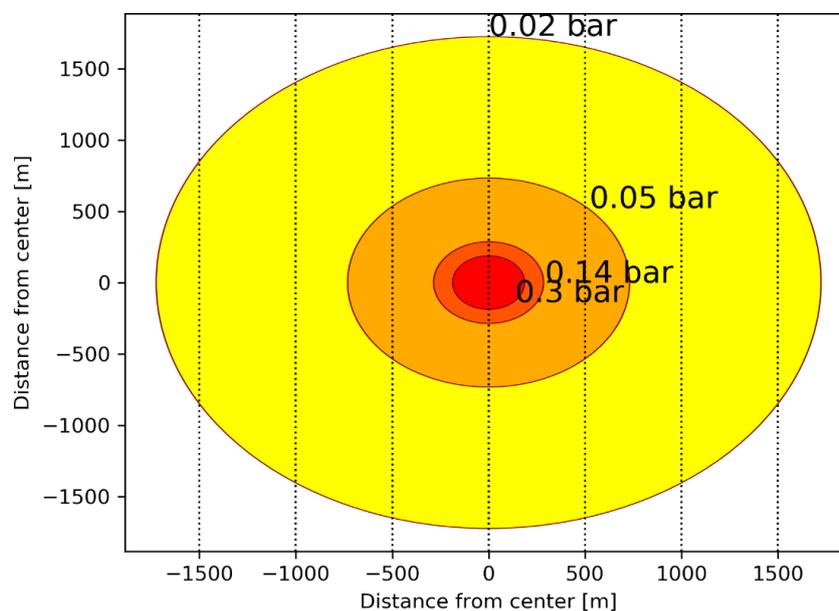


Figure 7 : Overpressure as a function of the distance from the explosion center

Table 2 : Distance from the explosion center for an overpressure given

Overpressure[bar]	Distance[m]
0.02	1714.9
0.05	722.5
0.14	276.3
0.3	177.4

3.1. Validation

The explosion module has been validated against a real experiment performed by ARMINES (L. Aprin, personal communication). In this experiment, 0.17 kg of methane mixed in a stoichiometric ratio with oxygen (0.59kg) has been ignited with an explosive charge. Four sensors have been installed at 15 meters from the charge and four others at 20 meters of the charge. The experiment results are shown in the Table 3.

Table 3: Results of the explosion test

Distance [m]	Overpressure [mbar]				Average [mbar]	Confidence interval 95 % [mbar]
15	0.0532	0.0493	0.0575	0.0472	0.0518	±0.0072 (±13,9%)
20	0.0345	0.0315	0.0399	0.0418	0.03693	±0.0076 (±20,5%)

As the explosion module has been designed to compute a distance for given threshold values, the validation exercise consisted in modelling the “theoretical” distance of the sensors for the average overpressure values shown in Table 3. The relative differences between these modelled distances and the sensors distances in the experimental set-up is used as validation metric. In this validation exercise, the combustion enthalpy of the methane was of 50345 kJ/kg (“NIST Methane,” n.d.). The results are shown in the Table 4.

Table 4 Comparison between model and experiment

Average overpressure [mbar]	Distance [m]	Estimated distance [m]	Relative difference
0.0518	15	14.00	6.7 %
0.03693	20	19.27	3.65 %

In this case the model seems to work well, the relative difference between the estimation and the average of data is closer to 0 than the size of the confidence interval (in relative value), but it is important to not forget the non-linearity of the model.

3.2. How to run a simulation

In order to start a simulation, the python files (run_explosion.py and explosion_utils.py) must be placed in the same folder, all the modules needed must be installed (numpy and matplotlib), and the three folders “log”, “request”, “results” must have been created. Then, a json file “X_sim_data.json” with the same structure as the one described in the Annex 6.2 must be present in the request folder. “X” is the id of the simulation and must be an integer. To start the simulation, type “python run_fire.py X”. A folder results/X will be created in which all the results will be stored, and a new logfile will be created if there is not already one for the month.

3.3. Model limitation

The model uses assumptions, which implies limitations:

- The model assumes a flat area with no obstacle.
- The model is designed for explosion due to combustion. The model **may not be used** for other explosion types such as explosion caused by another type of chemical reaction, rapid gas expansion, nuclear reaction or any other phenomenon. However, if the user can estimate a TNT equivalent for these explosions, the user can set this value in the input json file (advanced parameter equ_TNT) and run a simulation.
- The model has been validated against an explosion of small size.

4. Conclusion

Two models have been developed in python, and are available for all the MANIFESTS members. The energy flux from a fire as a function of distance and the overpressure caused by an explosion as a function of distance are computed. The explosion model has been validated with experimental data.

The models generate a figure and a table and in case of an error in the log. They are very lightweight in computation time and they will be used in the MANIFESTS interface for decision support tools.

Despite using the worst case as much as possible in the model hypothesis, it is still advised to look at the model outputs not as an accurate depiction of the reality but as a tool to help in decision making.

5. References

- Brzustowski, T.A., Sommer, E.C., 1973. , in: Predicting Radiant Heating from Flares. Washington DC, pp. 865–893.
- Burgess, D., Hertzberg, M., 1974. Radiation from pool flames, in: Heat Transfer in Flames, John Wiley and Sons. New York, p. 413.
- Drysdale, D., 1985. An introduction to fire dynamics. John Wiley & sons, New York.
- Frank P Lees, 2003. Loss prevention in the process industry.
- Heskestad, G., 1984. Engineering relations for fire plumes. Fire Safety Journal 7, 25–32.
[https://doi.org/10.1016/0379-7112\(84\)90005-5](https://doi.org/10.1016/0379-7112(84)90005-5)
- Instruction Technique du 9 novembre 1989 - dépôts aériens existants de liquides inflammables, 1989.
- Marshall, V.C., 1976. The siting and construction of control buildings – a strategic approach. I. Chem. E. Symp. Series No 47.
- Mudan, K.S., Croce, P.A., 1995. Fire hazard calculations for large open hydrocarbon pool fires, in: SFPE Handbook of Fire Protection Engineering. National Fire Protection association, NIST Methane [WWW Document], n.d. URL <https://webbook.nist.gov/cgi/cbook.cgi?ID=74-82-8> (accessed 12.6.21).
- Raj, P.K., 1977. Calculation of thermal radiation hazards from LNG fires, a review of the state of the art,. Presented at the American Gas Association. Transmission Conference, pp. 135–148.
- Thomas, P.H., 1963. The size of flames from natural fires. Symposium (International) on Combustion 9, 844–859. [https://doi.org/10.1016/S0082-0784\(63\)80091-0](https://doi.org/10.1016/S0082-0784(63)80091-0)
- TNO-Yellow Book-Methods for the calculations of physicals effects CPR14E, C.J.H. van den Bosch, R.A.P.M. Weterings. ed, 1997.

6. Annexes

6.1. Fire module input file

The inputs in bold underlined needs to be given by the user, the ones which are only bold are not necessary but are the most useful to help increase the accuracy (some example value are used), because they are computed from the others parameters or have a value by default. If the user provides the mass of the chemical, the volume and density will not be taken into account and can be put at any value. The flux threshold must be provided in increasing order. The advanced category defines parameters linked to the implementation of the model and to the parametrizations.

```
{
  "simulation_id": 1,
  "description": null,
  "environmental_data": {
    "wind_speed": {
      "value": 2.0,
      "units": "m/s"
    },
    "air_density": {
      "value": 1.2243,
      "units": "kg/m\u00b3"
    },
    "ambient temperature": {
      "value": 288.15,
      "units": "K"
    },
    "relative humidity": {
      "value": 70.0,
      "units": "%"
    }
  },
  "release_data": {
    "volume": {
      "value": 30.0,
      "units": "m\u00b3"
    },
    "surface": {
      "value": 600.0,
      "units": "m\u00b2"
    },
    "perimeter": {
      "value": 86.8,
```

```

    "units": "m"
  }
},
"pollutant_properties": {
  "density": {
    "value": 882.0,
    "units": "kg/m\u00b3"
  },
  "ebullition temperature": {
    "value": 353.25,
    "units": "K"
  },
  "vaporization enthalpy": {
    "value": 433.107,
    "units": "kJ/kg"
  },
  "Cp": {
    "value": 1.745,
    "units": "kJ/(kg\u00b7K)"
  },
  "combustion enthalpy": {
    "value": 40200.0,
    "units": "kJ/kg"
  }
},
"advanced": {
  "flux threshold": {
    "value": [
      1.5,
      3.0,
      6.0,
      12.5,
      25.0
    ],
    "units": "kW/m\u00b2"
  },
  "dev_tolerated": {
    "value": 0.01,
    "units": "%"
  },
  "ratio_add": 5,
  "cmb_yield": {
    "value": 70.0,
    "units": "%"
  },
  "fl_abs_smokes": {

```

```
"value": 80.0,
"units": "%"
},
"radiative_fraction": {
"value": 40.0,
"units": "%"
},
"fl_smokes": {
"value": 20.0,
"units": "kW/m\u00b2"
},
"min_dist": {
"value": 2.0,
"units": "m"
},
"start_dist": {
"value": 50.0,
"units": "m"
},
"visc_cin_air": {
"value": 7.51e-06,
"units": "m\u00b2/s"
},
"g": {
"value": 9.81,
"units": "m/s\u00b2"
},
"flame_param": {
"value": 1.0,
"keys": [
"Heskestad",
"Thomas"
]
},
"flux_param": {
"value": 2.0,
"keys": [
"IT89",
"Mudan_Croce",
"YBook"
]
},
"mass": {
"value": -1.0,
"units": "kg"
},
}
```

```

"Deq": {
  "value": -1.0,
  "units": "m"
},
"comb_flow": {
  "value": -1.0,
  "units": "kg/(m\u00b2\u00b7s)"
},
"energy_flux": {
  "value": -1.0,
  "units": "kW"
},
"fl_height": {
  "value": -1.0,
  "units": "m"
},
"fl0": {
  "value": -1.0,
  "units": "kW/m\u00b2"
}
}
}
}

```

Table 5: Description of the parameters for the fire module

Category	Name	Units	Description
/	<u>simulation_id</u>	/	ID of the simulation
	description	/	Description of the simulation
environmental_data	<u>wind_speed</u>	m.s ⁻¹	Speed of the wind
	<u>ambient temperature</u>	K	Temperature of the water, the air, and the pollutant
	<u>air_density</u>	kg.m ⁻³	Density of the air around the fire
	<u>relative_humidity</u>	%	Relative humidity of the air
release_data	<u>volume</u>	m ³	Volume of pollutant
	<u>surface</u>	m ²	Surface of the slick of pollutant
	<u>perimeter</u>	m	Perimeter of the slick of pollutant
pollutant_properties	<u>density</u>	kg.m ⁻³	Density of the pollutant
	<u>ebullition temperature</u>	K	Ebullition temperature of the pollutant

	<u>vaporization enthalpy</u>	kJ.kg^{-1}	Enthalpy of vaporization of the pollutant at combustion temperature
	<u>Cp</u>	$\text{kJ.kg}^{-1}.\text{K}^{-1}$	Specific heat of the pollutant
	<u>combustion enthalpy</u>	kJ.kg^{-1}	Combustion enthalpy of the pollutant
advanced	<u>flux threshold</u>	kW.m^{-2}	Thresholds for the distance estimation, in increasing order
	dev_tolerated	%	Maximal relative difference to stop the iteration
	ratio_add	/	Proportional to the increase in distance when doing the iterations
	cmb_yield	%	Amount of the pollutant which is fully burned
	fl_abs_smokes	%	Amount of the flux absorbed by the smokes
	radiative_fraction	%	Amount of energy radiated
	fl_smokes	kW.m^{-2}	Energy flux radiated by the smokes
	min_dist	m	The minimal value during an iteration
	start_dist	m	The starting distance for the iterations
	visc_cin_air	$\text{m}^2.\text{s}^{-1}$	Kinematic viscosity of air around the fire
	g	m.s^{-2}	Acceleration due to gravity
	flux_param	/	Parametrization chosen for computing the flux
	mass	kg	Mass of pollutant (-1 by default)
	Deq	m	Equivalent diameter of the slick (-1 by default)
	comb_flow	$\text{kg.m}^{-2}.\text{s}^{-1}$	Combustion flow of the pollutant (-1 by default)
	energy_flux	kW	Power emitted by the fire (-1 by default)
fl_height	m	Height of the flame (-1 by default)	

	f10	kW.m ⁻²	Flux at the fire (-1 by default)
--	-----	--------------------	----------------------------------

6.2. Explosion module input file

The inputs in bold underlined needs to be given by the user, the ones which are only bold are not necessary but are the most useful to help increase the accuracy (some example value are used), because they are computed from the others parameters or have a value by default. The overpressure thresholds must be provided in increasing order. If the user provide the mass of explosive, the volume and density will not be taken into account and can be put at any value, which can be useful for a gas where the user only know the mass released. The advanced category defines parameters linked to the implementation of the model and to the parametrizations.

```
{
  "simulation_id": 1,
  "description": null,
  "release_data": {
    "volume": {
      "value": 30.0,
      "units": "m\u00b3"
    }
  },
  "pollutant_properties": {
    "density": {
      "value": 882.0,
      "units": "kg/m\u00b3"
    },
    "combustion enthalpy": {
      "value": 40200.0,
      "units": "kJ/kg"
    }
  },
  "advanced": {
    "overpressure_threshold": {
      "value": [
        0.02,
        0.05,
        0.14,
        0.3
      ],
      "units": "bar"
    },
    "ex_cmb_yield": {
```

```

    "value": 10.0,
    "units": "%"
  },
  "energy_TNT": {
    "value": 4690.0,
    "units": "kJ/kg"
  },
  "mass": {
    "value": -1.0,
    "units": "kg"
  },
  "equ_TNT": {
    "value": -1.0,
    "units": "kg"
  }
}
}
}

```

Table 6 : Description of the parameters of the explosion module

Category	Name	Units	Description
/	<u>simulation_id</u>	/	ID of the simulation
	description	/	Description of the simulation
release_data	<u>volume</u>	m ³	Volume of explosive
pollutant_properties	<u>density</u>	kg.m ⁻³	Density of the explosive
	<u>combustion enthalpy</u>	kJ.kg ⁻¹	Combustion enthalpy of the explosive
advanced	<u>overpressure_threshold</u>	bar	Thresholds used to compute the distances
	<u>ex_cmb_yield</u>	%	Combustion yield of the pollutant when in an explosion
	energy_TNT	kJ.kg ⁻¹	Energy released by the explosion of TNT
	mass	kg	Mass of explosive (-1 by default)
	equ_TNT	kg	Mass of TNT required to release the same energy as the explosive (-1 by default)