Håkan Grahn

Modeling of dispersion, deposition and evaporation from ground deposition in a stochastic particle model
Håkan Grahn

Modeling of dispersion, deposition and evaporation from ground deposition in a stochastic particle model
## Contents

1 Introduction 5

2 The simulation model 7
   2.1 Boundary layer meteorology 7
   2.2 The dispersion model 8
   2.3 Dry deposition 9
      2.3.1 Gas deposition 10
      2.3.2 Particle deposition 12
      2.3.3 Implementation of deposition 14
   2.4 Evaporation from ground deposition 15

3 Software function 17
   3.1 New and improved functions 18
   3.2 Program flowchart and main structure 18

4 Model verification 21

5 Results 25

6 Discussion 27

7 Summary and Conclusions 29

8 Future development 31

References 33

Appendix A
User guide 35
   A.1 Meteorological data 35
   A.2 Evaporation 36
   A.3 Source waypoints 39
   A.4 Weather condition changes 39
   A.5 Input 39

Document information 46

Dokument information 47
1 Introduction

Often when hazardous gases are released in the atmosphere there will be one fraction pure gas and another fraction of aerosols. The gas cloud called the primary cloud, transports a rather long distance with the wind and disperses due to turbulence. On the other hand the aerosols, which also transport with the wind but in a rather short distance, fall down towards the ground where they are deposited. From this ground deposition layer gas evaporates and creates a gas cloud called the secondary cloud. In the current layout of the particle dispersion model used by the Swedish Defence Research Agency (FOI) this secondary cloud has to be created manually from the, possibly huge and irregular, ground deposition layer in a time consuming process. Other aspects the model doesn’t treat are weather changes during the simulation and complex source movements like releases from airplanes flying with different speeds and directions. The main purpose of this work is to implement an integrated handling of evaporation from the ground deposition layer, weather changes during the simulation and complex source movements in the existing particle dispersion model.
2 The simulation model

2.1 Boundary layer meteorology

When gases and aerosols are released in the lower part of the atmosphere they are transported with the wind field and dispersed due to turbulence. Almost all turbulence dispersion of pollutions in the atmosphere occurs in the boundary layer which consists of the lowest 100 to 3000 meters of the troposphere, the layer of the atmosphere that is closest to the ground. Above the boundary layer the dispersion is governed by large-scale weather systems not counted for in this report. This part of the troposphere above the boundary layer is often called the free atmosphere, see figure 1. The height of the boundary layer depends on the degree of turbulence and how it is generated inside the layer.

There are two basic processes that generate turbulence. One is mechanical turbulence generated by the frictional drag on the air flowing over the ground. The other one is the buoyancy. This occurs when an air parcel has different temperature than the surrounding air thus causing the air parcel to move because of density difference until the temperature difference is equalized. The degree of turbulence in the boundary layer is mainly divided into three broad categories.

The first category is called the unstable case which occurs when the temperature is decreasing with height and hence causes thermals of warm air to rise and generate buoyancy turbulence. This effect tends to increase the boundary layer height to well above 1000 meter. The second category is the neutral case. In such situation the temperature is nearly constant in the boundary layer and the turbulence is mainly mechanical and weaker. In the last category called the stable case the air is colder closer to the ground then at the top and the wind speed is rather low. These effects decrease the boundary layer height.
2.2 The dispersion model

There are a number of different methods to simulate dispersion in turbulent flows. There are Eulerian, Gaussian and Lagrangian methods. The one used in this work is a Lagrangian particle methods based on a probability density function (PDF).

In Lagrangian methods the coordinate system isn’t fixed to an inertial frame but instead it is fixed to a fluid particle which moves with the local fluid velocity. The fluid particles move continuously, so that the model simulates species concentration at different locations at different times. This could be compared to an Eulerian approach where the species concentration is maintained in an array of fixed computational cells.

The position of a fluid particle is denoted $X(t, Y)$, where $Y$ is the position of the particle at a reference time $t_0$. The velocity is expressed as

$$\frac{\partial}{\partial t} X(t, Y) = U(X(t, Y), t).$$

which enlightens the fact that fluid particles follow the local velocity field. With this type of coordinate system the Navier-Stokes equations in a constant gravitational field can be rewritten to

$$\rho \left( \frac{\partial}{\partial t} + U \cdot \nabla \right) U = \mu \nabla^2 U - \nabla P,$$

where $\rho$ is fluid density, $\mu$ is viscosity and $P$ is pressure.

In the used Lagrangian particle method each particle represents an air parcel. No momentum exchange takes place between the air parcel and the particles. Each particle carries a mass $m = M/N$, where $M$ is the total mass of the pollution and $N$ is the number of particles simulating the pollution.

Langevin Equation

Stochastic differential equations can be useful when ordinary calculus methods fail. A situation when this happens is for example when the particle velocities undergoes diffusion processes which lead to continuous but nowhere differentiable velocity trajectories, see figure 2. The type of stochastic differential equations used in this report is the Langevin equations which takes the form

$$dU(t) = a(U(t), X(t), t)dt + b(X(t), t)dW(t)$$

$$dX = Udt$$

where $U$ is velocity, $X$ position, $a(U, X, t)$ is the drift coefficient, $b(X, t)^2$ is the diffusion coefficient and $W(t)$ is a Wiener process. In a Wiener process $dW(t) = W(t + dt) - W(t)$ is a normal distributed stochastical variable with mean 0 and variance $dt$ [13]. These types of equations describe a diffusion process and are applicable to particles velocities in turbulent flows. The coefficients $a$ and $b$ depends on the turbulence model via the Fokker-Plank equation. The Fokker-Plank equation describes how the probability density function for the fluid elements varies with speed and time. Details about the derivation of $a$ and $b$ when the boundary layer is dominated by buoyancy generated turbulence, unstable case, can be found in Sehlstedt (2000) [15]. For the stable and neutral case, when the boundary layer is dominated by mechanical turbulence due to ground friction, the details of the derivation of $a$ and $b$ can be found in Schönfeldt (1997) [14].
2.3 Dry deposition

The purpose with this section is to give a complete description on how to calculate deposition velocities for a variety of different particles and gases.

Dry deposition refers to the transfer of airborne material, both gases and aerosols, to the ground including vegetation, soil, leaves etc where it is removed from the airborne stream. The transfer process leading to dry deposition can mainly be divided into three processes. The first process involves transport through the atmosphere to the immediate neighbourhood of the surface via surface-layer turbulence and is referred to as the aerodynamic component of transfer. The second process is the diffusion of material through the quasi laminar sublayer next to the surface and is dominated by molecular mechanisms. The third process is the absorption of the material and its possible removal from the surface through chemical and biological reactions. This process is called the substrate transfer component. In this final state the absorptivity of the matter at the surface determines how much material that is actually removed during the substrate transfer and deposition process. [2] Dry deposition is in the particle dispersion model modeled by a single parameter \( v_d \) (m/s), the deposition velocity.

To calculate the deposition velocity the three exchange processes mentioned are commonly identified as working in series like three resistances in the transport process, known as the resistance analogy. The deposition velocity is then a measure of conductivity between atmosphere and ground, i.e. the reciprocal of these resistances, \( v_d = \frac{1}{R_a + R_b + R_c} \).

Here \( R_a \) is the aerodynamic resistance determined by the ability of the turbulent eddies to bring material close to the surface. \( R_b \) is the sub-layer resistance which is the resistance to transfer material across the quasi-laminar sublayer. \( R_c \) is the surface resistance representing the combined resistance of the surface and
transfer to substrate [4].

Dry deposition for gases and for aerosols are physically different because of difference in the exchange processes. Below is two descriptions of these three resistances, one for gases and the other one for aerosols.

### 2.3.1 Gas deposition

When gases are deposited due to dry deposition in the model, the major resistance terms are generally the sub-layer resistance and the surface resistance while the aerodynamic resistance plays a minor role [4]. Figure 3 gives a short description of the resistance analogy.

![Figure 3. Scheme of resistance analogy.](image)

The modelling of the aerodynamic resistance is based on momentum transfer through the atmospheric surface level. According to Arya (1999) [2]

\[
R_a = \frac{\ln \left( \frac{z_{ref}}{z_0} \right) - \psi \left( \frac{z_{ref}}{L} \right)}{\kappa u_*},
\]

where \(z_{ref}\) is the reference height for aerodynamic resistance, \(z_0\) the roughness height, \(\psi\) the integrated stability function for momentum [4], \(L\) the Monin-Obukhov length defined in Appendix A, \(\kappa\) is von Karman constant (0.4) and \(u_*\) is the friction velocity, also defined in Appendix A. Further more Högström (1988) [5] suggests that the integrated stability function for the momentum can be approximated with the following empirical formulas

\[
\psi = -6 \frac{z_{ref}}{L}, \quad \text{for } \frac{z_{ref}}{L} \geq 0
\]

\[
\psi = \ln \left[ \left( \frac{1 + x^2}{2} \right) \left( \frac{1 + x}{2} \right)^2 \right] - 2 \tan^{-1} x + \frac{x}{2}, \quad \text{for } \frac{z_{ref}}{L} < 0
\]
where \( x = (1 - 19.3z/L)^{1/4} \). Equations (6) can be used for most practical applications in which great precision is not needed. The approximations are only valid for smooth and moderately rough surfaces, i.e. \( z_0/L \) is quite small. Also note that different authors have different estimations of the integrated stability function. E.g. Arya (1988) [1] estimates has a 5 instead of a 6 in the stable case and 15 instead of 19.3 in the unstable case.

The resistance due to molecular diffusion in the thin sublayer near a surface is defined as [7]

\[
R_b = 2.0 \left( \frac{Sc}{} \right)^{2/3},
\]

(7)

where the Schmidt number, \( Sc \), is the ratio of the kinematic viscosity of air, \( \nu \), to the particle Brownian diffusivity, \( D \),

\[
Sc = \frac{\nu}{D},
\]

(8)

where the Brownian diffusivity, \( D \), is calculated by Stokes-Einstein equation,

\[
D = \frac{k_B T C_c}{6 \pi \nu \rho_a \lambda},
\]

(9)

Here \( k_B \) is the Boltzmann constant, \( T \) is temperature, \( \rho_a \) is the air density and \( C_c \) is the Cunningham slip correction factor defined as

\[
C_c = 1 + \frac{2\lambda}{\nu R} \left( 1.257 + 0.5e^{-\frac{1.1\nu}{\lambda R}} \right),
\]

(10)

where \( \lambda \) is the mean free path of gas molecules in air (\( \lambda = 0.065 \times 10^{-6} \) m [4]). The sublayer resistance increases with increasing surface roughness and decreasing diffusivity of the substance. It is noted that different expressions for \( R_b \) are recommended by different authors. For example Hicks et al. [3] uses an expression where they divide the Schmidt number by the Prandtl number, which is assumed to be 0.72, thus implying a roughly 24% larger value of \( R_b \).

As dry deposition of gases only take place as a result of absorption at the surface the last resistance, \( R_c \), reflecting the surface properties place a vital rule in the dry deposition process. In a model presented by Zhang [10] \( R_c \) is in essence described by

\[
\frac{1}{R_c} = \frac{1 - W_{sl}}{R_{sl} + R_m} + \frac{1}{R_{ns}},
\]

(11)

where

\[
\frac{1}{R_{ns}} = \frac{1}{R_{cut}} + \frac{1}{R_{ac} + R_g},
\]

(12)

where \( W_{sl} \) is the fraction of stomatal\(^1\) blocking under wet conditions. \( R_{sl} \) is the stomatal resistance calculated using a sunlit/shade stomatal resistance model [11], where the canopy is subdivided into sunlit leaves and shaded leaves and the resistance is calculated as a function depending on, among other things, temperature, vapour pressure, photosynthesis etc. The mesophyll\(^2\) resistance \( R_m \) depends only on the chemical species and data for it can be found in table 1 [11]. \( R_{ns} \) is non

\(^1\)Small pores in the outer layer of a leaf or stem through which gases and water vapour pass. Also called stomata.

\(^2\)The photosynthetic tissue of a leaf.
stomatal resistance which further is decomposed into resistance due to cuticle uptake, $R_{cut}$, and into resistance to soil uptake including the chemical independent in-canopy aerodynamic resistance, $R_{ac}$, and the soil resistance, $R_g$. Equations (11) and (12) are only for surfaces with canopies. For surfaces without canopies like water, ice and desert $R_{st}$, $R_m$, $R_{ac}$ and $R_{cut}$ could be inapplicable, but for the convenience the same equations are used for these types of surfaces. The only difference when applying the equations to surfaces without canopies are that $R_{ac}$ is put to 0 and $R_{st}$, $R_m$ and $R_{cut}$ are given a very large value (i.e. $10^{25}$ s m$^{-1}$).

According to Zhang (2003) [10] $W_{st}$ is only important during situations when there are sunshine on a wet ground. Typical examples of this are sunny mornings with dew and sunshine immediately after rain. Thus the following formulas are suggested for wet canopies (for dry canopies, $W_{st}$ always equals 0):

$$W_{st} = \begin{cases} 
0, & SR \leq 200 \text{ Wm}^{-2} \\
(SR - 200)/800, & 200 < SR \leq 600 \text{ Wm}^{-2} \\
0.5, & SR > 600 \text{ Wm}^{-2}
\end{cases}$$

(13)

where $SR$ is the solar radiation.

According to [10]

$$R_{ac} = \frac{R_{ac0} \cdot LAI^{1/4}}{u_*^2},$$

(14)

where

$$R_{ac0}(t) = R_{ac0}(\text{min}) + \frac{LAI(t) - LAI(\text{min})}{LAI(\text{max}) - LAI(\text{min})} \cdot [R_{ac0}(\text{max}) - R_{ac0}(\text{min})]$$

(15)

$R_{ac0}$ can be found in table 1 [10]. LAI can be found in table 2 [11].

$R_g$ and $R_{cut}$ are calculated for SO$_2$ and O$_3$ and then scaled for other caseous species based on the formulas [10]

$$\frac{1}{R_{cut}(t)} = \frac{\alpha}{R_{cut}(\text{SO}_2)} + \frac{\beta}{R_{cut}(\text{O}_3)}$$

(16)

and

$$\frac{1}{R_g(t)} = \frac{\alpha}{R_g(\text{SO}_2)} + \frac{\beta}{R_g(\text{O}_3)}$$

(17)

where the scaling parameters $\alpha(i)$ and $\beta(i)$ can be found in table 1 [11] for the different substance, $i$.

### 2.3.2 Particle deposition

In the parameterization of particle dry deposition presented in this report $R_c$ is set to zero. This is based on an assumption that all particles hitting a surface sticks to it [4]. Furthermore the gravitational settling velocity is included in the expression for the dry deposition velocity. This leads to the following expression for particle dry deposition

$$V_d = V_g + \frac{1}{R_a + R_b}.$$  

(18)

1. A layer of waxlike, water-repellent material, cutin, covering the epidermis.
The gravitational settling velocity is calculated numerically according to

\[ V_g = \sqrt{\frac{8 \cdot g \rho_r (\rho_p - \rho_a) \rho_p}{3 \cdot C_D \rho_a}} \]  

(19)

\[ Re = \frac{2V_g r}{\nu} \]  

(20)

\[ C_D = \frac{24}{Re \left(1 + 0.173 \cdot Re^{0.657}\right)} + \frac{0.413}{1 + 16300 \cdot Re^{-1.09}}, \]  

(21)

where \( \rho_p \) and \( \rho_a \) is the density of the particle and the density of air respectively, \( g \) is the acceleration of gravity, \( r \) is the particle radius, \( C_D \) the drag coefficient, \( \nu \) is the kinematic viscosity of air and \( Re \) is the Reynolds number. Notice that the settling velocity, the Reynolds number and the drag coefficient are dependent on each other and several iterations of the equation system are therefore needed to calculate the settling velocity. To get a rough estimation of the settling velocity

\[ V_g = \frac{\rho_p (2r)^2 g C_c}{18 \eta} \]  

(22)

may be used as an estimation. Here \( \eta \) is the viscosity coefficient of air and \( C_c \) is the slip correction factor (Cunningham correction factor) for small particles (equation (10)).

The aerodynamic resistance, \( R_a \), is calculated in the same way as for dry deposition of gases, see equation (5). The surface resistance in equation (22), \( R_b \), depends on several deposition processes, particle size, atmospheric conditions and surface properties [9]. \( R_b \) is parameterized by

\[ R_b = \frac{1}{\epsilon_0 u_s (E_B + E_{IM} + E_{IN}) R_1}, \]  

(23)

where \( E_B, E_{IM}, E_{IN} \) are collection efficiency from Brownian diffusion, impaction and interaction, respectively; \( R_1 \) is a correction factor representing the fraction of particles that stick to the surface. \( \epsilon_0 \) is an empirical constant and is taken as 3. [9]

The collection efficiency due to Brownian diffusion is a function of the Schmidt number given as

\[ E_B = Sc^{-\gamma}. \]  

(24)

The value \( \gamma \) varies with land type categories and have values between 1/2 and 2/3 [16].

The impaction process, \( E_{IM} \), is controlled by the Stokes number, \( St \), which has the form \( St = V_g u_s / g A \) for vegetated surfaces and \( St = V_g u_s^2 / \nu \) for smooth surfaces. \( A \) is a characteristic “radius” of large collectors, e.g. grass blades, stalks, needles etc.

Different expressions for the impaction process are suggested for different types of surfaces. According to Slinn (1982) [16] smooth surface gives

\[ E_{IM} = 10^{-3/St} \]  

(25)

and for vegetative canopies the collection efficiency for impaction is

\[ E_{IM} = \frac{St^2}{1 + St^2}. \]  

(26)
Impaction efficiency over a spruce forest is according to Peters and Eiden (1992) [6]

$$E_{IM} = \left( \frac{St}{\alpha + St} \right)^\beta,$$  \hspace{1cm} (27)

where $\alpha$ and $\beta$ are constants. A good choice for $\beta$ is 2 according to Peters and Eiden (1992) with $\alpha$ depending on land type category. $\alpha$ varies between 0.8 [6] and 2 for surfaces with canopies. For surfaces without canopies the impaction efficiency is negligible and hence $\alpha$ has a rather large value between; 50-100 [9].

When a particle passes an obstacle at a distance shorter than its physical dimension the mechanisms included in the collection efficiency by interception is of importance. This is especially true for large particles over hairy leaves. Interception deposition during particle flow around surfaces as in canopies of coniferous trees is negligible compared to the other deposition mechanisms. Here the collection efficiency by interception is defined as

$$E_{IN} = \frac{1}{2} \left( \frac{2r}{A} \right)^2$$  \hspace{1cm} (28)

where $r$ is the radius of the particles and $A$ is defined as a characteristic radius of large collectors. Typically $A$ is 2 for evergreen-needleleaf trees and 10 for urban terrain.

At last the factor $R_1$, which represent the fraction of particles sticking to the surface in equation (23), has the following form:

$$R_1 = e^{-St^{1/2}}.$$  \hspace{1cm} (29)

2.3.3 Implementation of deposition

The way which the mass flux to the surface is defined is by the dry deposition velocity concept

$$F_m = C v_d,$$  \hspace{1cm} (30)

where $C$ is particle concentration (1/m$^3$) near the surface. This is a cumbersome approach in the Langevin model since each particle is treated individually and the concentration is therefore unknown to each simulation particle. To overcome this problem a deposition probability is derived [8]

$$P = \frac{F_m}{F_p} = (2\pi)^{1/2} \frac{v_d}{\sigma},$$  \hspace{1cm} (31)

where $\sigma$ is the standard deviation of a Gaussian velocity distribution and $F_p$ is mass flux through the surface due to a Gaussian velocity distribution,

$$F_p = \int_{w<0} d^3 V f(V) V \cos(\theta) = \frac{C \sigma}{(2\pi)^{1/2}}.$$  \hspace{1cm} (32)

Where $f(V)$ is the Gaussian velocity distribution with zero mean and standard deviation $\sigma$, $\theta$ is the angle between the particle velocity and the z-axis and the integration boundary, $w < 0$, means integration over all velocities with negative z-component. The probability that an individual particle is deposited when it hits the surface is then proportional to the deposition probability.
2.4 Evaporation from ground deposition

In the implementation of evaporation from the ground deposition the probability for each individual particle to evaporate is calculated during each time step. The probability for evaporation, $P_{evap}$, in the time interval $\Delta t = t_1 - t_2$ is defined as

$$P_{evap} = ppact\_fact \cdot \int_{t_1}^{t_2} \frac{dF_{re}}{dt'} dt'$$  \hspace{1cm} (33)

where $ppact\_fact$ is the factor with unit activity per particle that relates the number of deposited particles to the number of evaporated particles and $\frac{dF_{re}}{dt}$ is the evaporation. $\frac{dF_{re}}{dt}$ is defined as

$$\frac{dF_{re}}{dt} = \frac{\dot{m}(t)}{M(t)}$$ \hspace{1cm} (34)

where $\dot{m}$ is the change in mass and $M$ is the total mass remaining since the particle deposited. However, the evaporation is often given in terms of $\frac{\dot{m}(t)}{M_0}$ and $F_{ev}$, where $M_0$ is the initial mass and $F_{ev}$ the fraction evaporated. This implies that some modifications to equation (33) must be done to use these values as inputs. Since the remaining mass is equal to the initial mass minus the evaporated mass, i.e.

$$\frac{\dot{m}(t)}{M(t)} = \frac{\dot{m}(t)}{M_0 - \int_{t_0}^{t} m(t)dt'}$$

$$= \frac{\dot{m}(t)}{M_0} \cdot \frac{1}{1 - \int_{t_0}^{t} \frac{\dot{m}(t)}{M_0} dt'}$$ \hspace{1cm} (35)

equation (34) can also be written as

$$P_{evap} = ppact\_fact \cdot \int_{t_1}^{t_2} \frac{dF_{re}}{dt'} dt' = ppact\_fact \cdot \int_{t_1}^{t_2} \frac{\dot{m}(t)}{M_0} \cdot \frac{1}{1 - F_{ev}(t)} dt'$$ \hspace{1cm} (36)

The values of $\frac{\dot{m}(t)}{M_0}$ and $F_{ev}$ are based on a model for evaporation from a glass surface used by the model GASSY [12] which may render a rather imprecise estimation of the real values in some situations. To make the model more independent from this approximation $\frac{\dot{m}(t)}{M_0}$ and $F_{ev}$ are read from a table stored in a separate data file which easily could be updated.
3 Software function

In this section an introduction to the main flow in the model and some of the developed algorithms are presented. The purpose is to give a short description of the software in general and to explain the advantages achieved with the new subroutines and discuss some of the decisions made during the development process.

In figure 4 a schematic picture of the existing Fortran90 software is presented together with both new and old routines. Two entirely new routines, evaporate_density and init_new_particle, are constructed where evaporated_density calculates the number of new evaporated particles from each deposited particle and init_new_particle generates the new particles. Some of the modified routines are out_ptp, movepar_ana and movepar_ana_convecive.

The reason for modifying these routines are mainly due to the introduction of a derived-type structure in the loop_ana routine. A derived-type structure is in Fortran a variable that contains other variables called attributes to the derived-type structure. The derived-type in loop_ana controls the particles attributes, e.g. position, velocity, size etc.

The execution of the program starts in the main function ptk_move_ana which reads the input variables from the data file input_ana.nml and thereafter calculates boundary layer height and aligns the x-axis along with the wind direction. The main function thereafter calls the routine ptkstp_ana, which controls the entire simulation. The routine consists of a main loop over the total time during which a particle will be moved and three sub loops inside this main loop, the “distribution loop”, the “evaporation loop” and the “old particle movement loop”.

Before the main loop in ptkstp_ana reaches any of the sub loops a time interval during which particles will be moved is decided and all source movements due to waypoints given by the user are made. A waypoint is a position where a source is supposed to be at a given time. And hence the waypoints define a route for the sources. The time interval is adjusted so that no source-start or -stop time or output time as decided by the user is missed. When a time interval is set and all source movements are done ptksp_ana reaches the “distribution loop” which is a loop over all the evaporable deposited particles. The purpose of this loop is to decide the number of evaporated particles in the given time interval. For each deposited particle an evaporation probability is determined depending on the time since the particle deposited and the evaporation rate which is tabulated in the input file evaporation data. Then the routine init_new_particle generates the new gas particles.

The next loop entered is the “evaporation loop” which is a loop over all the evaporable deposited particles. The purpose of this loop is to decide the number of evaporated particles in the given time interval. For each deposited particle an evaporation probability is determined depending on the time since the particle deposited and the evaporation rate which is tabulated in the input file evaporation data. Then the routine init_new_particle generates the new gas particles.

Thereafter the rest of the particles that haven’t been moved yet are moved in the “old particle movement loop” which moves them with the movepar_ana routines. In the last part of the main loop the routine out_ptp, where the particles state are written to files, is called every time the actual time coincidence with the
output time interval.

The routine that calculates the evaporation probability in the “evaporation loop” mentioned above is \textit{evaporate\_density}. The routine takes a deposited particle and calculates an evaporation probability by integrating \( \frac{dRe}{dt} \) according to equation (33). The integration is performed with the trapezoidal rule which is a fast method with limited accuracy, especially when a small number of data points are used. However the error caused by the trapezoid rule will be a little bit smaller than expected. The reason for this is that the fewer particles left on the ground, the harder it is to maintain a constant mass flow from the deposition layer. Therefore, a temporary overestimation of the integrand, will at first overestimate the mass flow but later on, when the number of deposited particles has decreased sufficiently much the mass flow will be underestimated. The net mass flow is therefore a little bit lower than suggested by the trapezoid rule. The opposite occurs if the integrand is underestimated.

3.1 New and improved functions

The changes in the program from previous versions are summarized by the four points:

- Automatization of the modeling of the evaporation process of deposited particles. A major improvement
- A separate file is created at the end of the simulation so another simulation can continue where the old one stopped by reading all the old particle data from that file. This is useful for changing weather condition during a simulation, which could be done by editing the input data between different simulations.
- The wind direction can change to an arbitrary direction during the simulation.
- The sources can now move in any direction through the entire simulation. This is made possible by specifying discrete trajectories for the sources. The trajectories are defined by points in time and space here called waypoints.

3.2 Program flowchart and main structure

In this section the flowchart of the program is presented, figure 4, and also a short description of the main algorithm in the program.

1. In ptk\_move\_ana
   1.1 Read input variables from input\_ana.nml
   1.2 Calculate boundary layer height.
   1.3 Align x-axis along the wind direction.

2. Call ptkstp\_ana
   2.1 Read the rate of evaporation as a function of time from input table given in input\_ana.nml
Figure 4. A schematic picture over the structure of the program with its subroutines and functions. The bold and italic text marks new routines and the italic text marks improved old routines.

2.2 Start the main-loop

2.2.1 Decide the current time interval so that no source start or stop time is missed and so that each output time given by the user occurs.

2.2.2 Calculate source movement and move them according to the given Waypoints.

2.2.3 Start distribution loop

2.2.3.1 Loop over each source and calculate the number of new particles released from each source during the given time interval.

2.2.3.2 Initiate the new particles with the function init_new_particle

2.2.3.3 Call radius_lognorm and w_settling to calculate particle radius and settling velocity.

2.2.3.4 Call a movepar_ana routine to move the particle.

2.2.4 Start evaporation loop.

2.2.4.1 Loop over each deposited particles and check the rate of evaporation during the time interval with the function evaporate_density.

2.2.4.2 Integrate the rate of evaporation to calculate the number of new gas particles from the deposited particles.

2.2.4.3 Loop over the newly evaporated particles and initiate them with the init_new_particle function.
2.3 Start the old particle movement loop
  2.3.1 Loop over each old particle that haven’t been moved yet.
  2.3.2 Call some of the movepar_ana routines to move the old particles.
2.4 Call outptp to save the particle information to files.

3. Finally create a data file that contains information such as how the simulation was set up, simulation time etc.
4 Model verification

In this section the verification of the model done here is presented and evaluated. The major test is that the evaporation from the ground deposition layer is consistent with the given evaporation in the input files. Another test is that the simulated dispersion is unaffected by the new development since no changes are made in the dispersion model.

As a first verification of the model a number of simulations are done with different substances evaporating from a ground deposition layer during different weather conditions. The evaporation is then compared to the evaporation given by GASSY [12].

The simulated substances are VX and sarin. VX is chosen because of its low volatility and hence long evaporation time and sarin is chosen because if it’s high volatility and short evaporation time. Both impure and pure variants of the substances are tested. Each substance was tested with two different \( ppact \) factors and with different purities in two different weathers, see table 1. \( ppact \) is a factor that specifies the relationship between the source mass and the number of evaporated particles. I.e if \( ppact \) is chosen to 10 and 5 particles are deposited representing 5 kg of some substance, then 50 particles will evaporated from these 5 deposited particles. Each deposited particles will represent a mass of 1 kg and each evaporated particle will represent a mass of 0.1 kg. The \( ppact \) factor is especially useful when the substance has a very low volatility and more particles are needed to improve the statistic.

<table>
<thead>
<tr>
<th>Substance</th>
<th>Vind (m/s)</th>
<th>Temperature (°C)</th>
<th>Purity (%)</th>
<th>Particle diameter (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VX</td>
<td>0.57</td>
<td>20</td>
<td>100</td>
<td>0.3</td>
</tr>
<tr>
<td>VX</td>
<td>1.0</td>
<td>20</td>
<td>50</td>
<td>0.3</td>
</tr>
<tr>
<td>Sarin</td>
<td>0.3</td>
<td>13</td>
<td>100</td>
<td>0.3</td>
</tr>
<tr>
<td>Sarin</td>
<td>4</td>
<td>13</td>
<td>50</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The results of the evaporation test can be seen in figure 5 and table 2. In each pictures there are three different graphs. One with \( ppact=1 \), one with \( ppact=10 \) and the third one is the evaporation given from the GASSY model. Our model tends to slightly overestimate the evaporation compared to GASSY, especially when the substances are impure. For pure substances the error is below 2% in the end of the simulation. In all the four cases a higher \( ppact \) factor improves the model a little bit.

The evaporation overestimation is due to the choice of integration scheme. The trapezoid rule,

\[
\int_{x_1}^{x_2} f(x)dx = \frac{x_2 - x_1}{N} \left[ \frac{1}{2} f(x_1) + \frac{1}{2} f(x_2) \right] + O \left( \frac{(x_2 - x_1)^3}{N^3} f'' \right), \quad (37)
\]

where \( N \) is the number if steps, is indeed fast but inaccurate. This is especially true in this case since the known points are few and in the impure cases unevenly distributed. Further the integrand is a monotonic decreasing or increasing function and hence is the second derivative non zero, see equation (36). The maximal error due to the trapezoid rule when the substance is VX with 50% purity is about 9%. This should be compared to the error shown in figure 5 which is about 7%, see table 2.
Table 2. The relative error in evaporation compared to GASSY for different substance, purity and ppact factor.

<table>
<thead>
<tr>
<th>Substance</th>
<th>Purity (%)</th>
<th>ppact</th>
<th>Mass evaporated (%)</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VX</td>
<td>100</td>
<td>1</td>
<td>90</td>
<td>2.2</td>
</tr>
<tr>
<td>VX</td>
<td>100</td>
<td>10</td>
<td>90</td>
<td>2.0</td>
</tr>
<tr>
<td>VX</td>
<td>50</td>
<td>1</td>
<td>90</td>
<td>6.5</td>
</tr>
<tr>
<td>VX</td>
<td>100</td>
<td>10</td>
<td>90</td>
<td>6.4</td>
</tr>
<tr>
<td>sarin</td>
<td>100</td>
<td>1</td>
<td>90</td>
<td>1.7</td>
</tr>
<tr>
<td>sarin</td>
<td>100</td>
<td>10</td>
<td>90</td>
<td>2.1</td>
</tr>
<tr>
<td>sarin</td>
<td>50</td>
<td>1</td>
<td>90</td>
<td>7.7</td>
</tr>
<tr>
<td>sarin</td>
<td>100</td>
<td>10</td>
<td>90</td>
<td>6.4</td>
</tr>
</tbody>
</table>

Figure 5. Percentage evaporated plotted against time. The green line with squares is evaporation with ppact=1, the red line with x:s is evaporation with ppact=10 and the blue line with diamonds is evaporation due to the GASSY model.

The second verification of the model is the comparison with simulations done before any changes was made to the model. For that purpose four simulations with this new model were compared with previous simulations from two scenarios, one including chemical attack with either VX and one with sarin. The scenarios are defined in table 3 and 4:

The dispersion is supposed to occur over a typical Swedish forest with a roughness factor, $z_0 = 1.0$. 
Table 3. Weather, initial airborne part of 2160 kg sarin. Drop diameter 0.3 mm. Ground contamination area 1 km². L is Monin.Obukhovs length and $u_*$ is friction velocity both defined in appendix A.

<table>
<thead>
<tr>
<th>Weather</th>
<th>Initial airborne part</th>
</tr>
</thead>
<tbody>
<tr>
<td>+10 - +15 °C, wind 1 m/s, stable stratification, $L = 350$ m and $u_* = 0.170$ m/s</td>
<td>1296 kg (60%)</td>
</tr>
<tr>
<td>+10 - +15 °C, wind 4 m/s, neutral stratification, $L = 1 \cdot 10^6$ m and $u_* = 0.695$ m/s</td>
<td>1296 (60%)</td>
</tr>
</tbody>
</table>

Table 4. Weather, initial airborne part of 1520 kg VX. Drop diameter 0.3 mm. Ground contamination area $800 \times 600$ m. L is Monin.Obukhovs length and $u_*$ is friction velocity both defined in appendix A.

<table>
<thead>
<tr>
<th>Weather</th>
<th>Initial airborne part</th>
</tr>
</thead>
<tbody>
<tr>
<td>+10 - +15 °C, wind 1 m/s, stable stratification, $L = 350$ m and $u_* = 0.170$ m/s</td>
<td>152 kg (60%)</td>
</tr>
<tr>
<td>+10 - +15 °C, wind 4 m/s, neutral stratification, $L = 1 \cdot 10^6$ m and $u_* = 0.695$ m/s</td>
<td>152 (10%)</td>
</tr>
</tbody>
</table>

Figure 6. Damage effect.
5 Results

An example of a good scenario which includes most of the models functionality was constructed in the spring of 2004 as an exercise for a NBC-team. In the beginning of the scenario an airplane with a load of 300 kg liquid soman flies over Umeå and unleashes its load. The plane flies from Rådhuset to the railroad station where it turns left and flies along E12 towards the viaduct where the road E4 crosses the road E12. On this viaduct the load is empty and the release stops. From the caused ground contamination, gas evaporates and a hazardous secondary cloud is created. The timescale of the scenario is three hours. A description of the weather during the scenario is given in table 5.

<table>
<thead>
<tr>
<th>Time</th>
<th>Wind Speed</th>
<th>Weather Condition</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00-12:55</td>
<td>Wind from east 3m/s</td>
<td>Cloudy</td>
<td>12 °C</td>
</tr>
<tr>
<td>12:55-13:05</td>
<td>Wind from east 2m/s</td>
<td>Partly cloudy</td>
<td>20 °C</td>
</tr>
<tr>
<td>13:05-14:00</td>
<td>Wind from east 2m/s</td>
<td>Sunny</td>
<td>25 °C</td>
</tr>
<tr>
<td>14:00-15:00</td>
<td>Wind from west 3m/s</td>
<td>Sunny</td>
<td>25 °C</td>
</tr>
</tbody>
</table>

In the simulation two different types of atmospheric stabilities are simulated. Changes in the wind speed and its direction are also included. This together with the source movements in different directions illustrates all the improvements made to the model i.e., the scenario tests the models capability of handling evaporation from the ground deposition layer, weather changes and complex source movements.
Figure 7. V17.
6 Discussion

As can be seen in figure 7 the first hour when the stratification is stable the ground concentration is larger than it is during the two last hours when the stratification is unstable. This is the case despite the temperature increase and hence a faster evaporation in the last two hours. The reason for this is the difference in boundary layer height. When the stratification is stable the boundary layer is lower compared with the boundary layer height during unstable stratification. Therefore the concentration decreases when the stratification turns unstable and the occurrence of downdrafts and updrafts increases the dilution of the substance.

Another effect is the sudden concentration increase when the wind changes. This is the effect of having the same cloud passing the same location twice. A half an hour after the wind direction change, when the concentration from the old secondary cloud has decreased, one can notice that the concentration is lower than during the second hour. This is due to the wind increase during this last hour.
7 Summary and Conclusions
8 Future development

The current implementation of the particle dispersion model has some disadvantage like the horizontally homogeneous wind field or the disability to specify the roughness parameter in more than one direction.
References


Appendix A

User guide

A.1 Meteorological data

The first thing of interest when particle dispersion in the planetary boundary layer is to be simulated is the weather conditions during the simulation which affect the degree of turbulence. Below is a parameter by parameter description of the relevant meteorological parameters that should be specified in the file input_ana.nml.

Roughness length

The roughness length \( z_0 \) is defined to be the height over the displacement plane where the mean wind becomes zero. Typically this value is around 1/10 of the vegetation and obstacles covering the ground.

\( z_0 \) together with the parameter for the lowest allowed \( z \)-value can affect the stability of the differential equations solved in the convective case.

Friction velocity

The friction velocity \( u_* \) is defined as \( u_* = (|\tau/\rho|)^{1/2} \), where \( \tau \) is the Reynolds stress and \( \rho \) is density, is approximately proportional to the mean velocity. The square of the friction velocity, \( u_*^2 \), is called the kinematic stress and is stress per unit density of air.

This parameter could be used to edit the wind speed.

Obukhovs length

(Monin-)Obukhovs length defined as, \( L = -u_*^2 T_v / g \rho Q_{v_0} \), where \( T_v \) is virtual temperature\(^4\), \( k \) is von Karman’s constant which is set to a value of 0.4 [18], \( g \) is gravitational acceleration and \( Q_{v_0} \) is kinematic virtual temperature flux at the surface in units of \( K \cdot m/s \). Obukhovs length is interpreted as the height where the buoyancy generated turbulence becomes dominant over the dynamically generated turbulence.

This parameter tells if it is stable, neutral or unstable stratification where \( L \) is positive for stable conditions, negative for unstable and 0 for neutral conditions. Obukhovs length can also be used to edit the wind speed.

Vertical virtual potential temperature gradient, \( \gamma \)

The constant \( \gamma \) is the gradient of the virtual potential temperature, \( \gamma = \frac{\partial \Theta}{\partial z} \), where \( \Theta \) is the virtual potential temperature and \( z \) is the height above ground. By using \( \Theta \), the degree of stratification is divided into the three categories:

1. Unstable, when \( \frac{\partial \Theta}{\partial z} < 0 \)

\(^4\)Virtual temperature is the temperature that dry air would have if its pressure and density were equal to that of moist air.
2. Neutral, when $\frac{\partial \theta}{\partial z} = 0$

3. Stable, when $\frac{\partial \theta}{\partial z} > 0$

Brunt-Väisälä frequency

Brunt-Väisälä frequency, $N = \left( \frac{g \partial T_v}{\partial z} \right)^{1/2}$ where $\frac{\partial T_v}{\partial z}$ is the vertical gradient of the virtual potential temperature, $g$ is gravitational acceleration and $T_v$ is virtual temperature, is defined to be the frequency at which a sufficiently small volume of air would oscillate when displaced vertically under stable conditions. The oscillation arise when air parcels are hotter (or cooler) than the surrounding air and experience a buoyancy force which will give them an upward (or downward) velocity towards equilibrium. When the air parcels passes the equilibrium condition the buoyancy force changes sign and an oscillation has started.

The frequency is undefined under unstable stratification and equals 0 in neutrals stratification.

Coriolis parameter

The Coriolis parameter is derived from the scalar value of the Coriolis force that is acting on a moving object on earth. The Coriolis force, $F = 2\Omega \sin \Theta v + \sin \Theta v^2/R$, where $\Omega$ is angular frequency of earth given in radians per second and $\Theta$ is latitude, $R$ is the earth radius and $v$ is the speed of the object. The last term is so small, it can be neglected compared to the first term. Thus $F = 2\Omega \sin \Theta \cdot v = f \cdot v$, where $f$ is the Coriolis parameter.

Typical value in Sweden is $1.3 \cdot 10^{-4}$ 1/s.

Large scale vertical velocity

The large scale vertical velocity, $W_{BL}$, is the vertical velocity on the top of the boundary layer. The $W_{BL}$ tends to increase the boundary layer height.

Wind angle

With the wind angle the wind can be rotated clockwise around the z-axis. The wind angle is defined in standard meteorological convention so that a wind angle of $270^\circ$ corresponds to a west wind or wind in the positive x-direction, i.e. the standard direction for the wind in the model, and a wind angle of $0^\circ$ is the same as a wind in the negative y-direction, or north wind, see figure ??.

A.2 Evaporation

When estimating evaporation from the ground deposition layer the interesting factor is the change in the relative fraction remaining with time, defined as $\frac{dF_r}{dt}$. The number of particles evaporated is roughly this factor times $\Delta t$. Theoretically $\frac{dF_r}{dt}$ is constant with time and a function of temperature, particle diameter, substance

---

5Virtual potential temperature is the potential temperature where dry air would have the same density as moist air.
purity and wind speed at a constant height above the ground. However due to impurities and ground absorption $\frac{dF_r}{dt}$ is time dependent.

The parameter $\frac{dF_r}{dt}$ is read from a separate data file containing a table of the form:

1. Number of different temperatures, number of different wind speeds and number of different deposited aerosol drop diameters.

2. For each temperature:
   2.1 For each wind speed:
      2.1.1 For each diameter:
         2.1.1.1 Time in hour to use the specific $\frac{dF_r}{dt}$, the value of the derivative and the fraction evaporated.

See table 6 for a complete description of the file layout.

There is a Python script called gassy.py for generating evaporation tables using the model GASSY.

Bellow is a description of parameters necessary to define for evaporation in the input file.

**Evaporate**

The parameter *Evaporate* is a logical variable set to *true* if deposited non-gas particles should be able to evaporate.

**ppact_fact**

The *ppact_fact* parameter specifies the relationship between the number of evaporated particles and the number of particles released from the source, i.e. if ppact_fact, is equal to 10 when 100 fluid particles are released from the source representing a total mass of 100 kg each evaporated particle will represent 0.1 kg of the substance. In concentration ppact_fact acts as

$$c_m = \frac{m_{tot}}{N_f \cdot \text{ppact_fact}}.$$  \hspace{1cm} (38)
Table 6. Example of an input data table for estimation of $\frac{dF_r}{dt}$.
Comments start with !.

<table>
<thead>
<tr>
<th>! Number of different temperatures in file</th>
<th>Number of different wind speeds in file</th>
<th>Number of different diameters in file</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>10.0</td>
<td>1.0</td>
<td>0.001</td>
</tr>
<tr>
<td>! time [h]</td>
<td>$\frac{dF_r}{dt}$</td>
<td>Mass evaporated (%)</td>
</tr>
<tr>
<td>.339</td>
<td>.0000819</td>
<td>10.0</td>
</tr>
<tr>
<td>.403</td>
<td>.0000239</td>
<td>90.0</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>10.0</td>
<td>1.0</td>
<td>0.002</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>10.0</td>
<td>2.0</td>
<td>0.001</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>20.0</td>
<td>1.0</td>
<td>0.001</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>20.0</td>
<td>1.0</td>
<td>0.002</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>! temp [$^\circ$C]</td>
<td>wind speed [m/s]</td>
<td>diameter [m]</td>
</tr>
<tr>
<td>20.0</td>
<td>2.0</td>
<td>0.001</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

where $c_m$ is mass concentration ($kg/m^3$), $c$ is particle concentration ($1/m^3$), $m_{tot}$ is the total mass ($kg$) released from the source and $N_f$ is the number of particles released from the same source.

resusp_fact

With resusp_fact the particle type from different sources can be specified. A source with resusp_fact defined to 0 is a gas source, 1 is a source with fluid drops and 2 is a solid particle source.
create_evap_file

The parameter create_evap_file is used to specify what type of out file the program should produce, see table 7 for the different values of the parameter and their meanings.

release_fact

Dimension less factor used to modulate evaporated particles release height. Evaporated particles release height is equal to

$$\text{height} = \frac{\nu_a z_{\text{rough}}}{u_s} \frac{10^5}{z_{\text{low}}} \text{release\_fact}, \quad (39)$$

where $\nu_a$ is air viscosity. The first part in equation (39) ($z_{\text{rough}} \cdot \nu_a / u_s$) represent the upper level where the aerodynamic resistance is the dominating deposition resistance.

A.3 Source waypoints

To simulate source movements a number of waypoints can be specified for each source. The conditions are that each source must have the same number of waypoints so if one source is moving and the other is not then the one not moving must have fixed waypoints.

Here is an example with $n$ waypoints and $m$ sources where subscription denotes the waypoint number and superscription denotes the source number:

$$n\_\text{waypoints} = n$$
$$x\_\text{waypoint} = x_1^1, x_2^1, \ldots, x_n^1, x_1^2, x_2^2, \ldots, x_n^2, \ldots, x_1^m, \ldots, x_n^m$$
$$y\_\text{waypoint} = y_1^1, y_2^1, \ldots, y_n^1, y_1^2, y_2^2, \ldots, y_n^2, \ldots, y_1^m, \ldots, y_n^m$$
$$z\_\text{waypoint} = z_1^1, z_2^1, \ldots, z_n^1, z_1^2, z_2^2, \ldots, z_n^2, \ldots, z_1^m, \ldots, z_n^m$$

A.4 Weather condition changes

To change weather conditions during a simulation the model can write to an out file specified by the variable save_out_file that stores the entire final state the particles have at the end of the run. Then another simulation with a different input data can continue from exactly the same position, if the out file from the previous run is renamed to the name of the in file specified be the variable read_in_file.

A.5 Input

The input data to the model are stored in a file called input_ana.nml, see tables 8, 9, 10, 12, 11 and 13 for the full list of input parameters that must be stated in the file.
<table>
<thead>
<tr>
<th>creat_evap_file</th>
<th>Out files description</th>
</tr>
</thead>
</table>
| 1               | Don’t separate between any types of particles. Only create the four files:  
|                 | **NAMN_IN_t.dat**  
|                 | Contains time for output and the number of airborne particles.  
|                 | **NAMN_IN_xyz.dat**  
|                 | Contains x-, y-, z-, age, radius and settling velocity for the airborne particles.  
|                 | **NAMN_IN_t_dep.dat**  
|                 | Contains times for output and number of deposited particles.  
|                 | **NAMN_IN_xyz_dep.dat**  
|                 | Contains x-, y-, z-, age, radius and settling velocity for the deposited particles.  
| 2               | Separate between gas and non-gas particles. Creates the four files mention above, but this time they only contains non-gas particles, and additionally the four files:  
|                 | **NAMN_IN_gas_t.dat**  
|                 | Contains time for output and the number of airborne gas particles.  
|                 | **NAMN_IN_gas_xyz.dat**  
|                 | Contains x-, y-, z-, age, radius and settling velocity for the airborne gas particles.  
|                 | **NAMN_IN_gas_t_dep.dat**  
|                 | Contains times for output and number of deposited gas particles.  
|                 | **NAMN_IN_gas_xyz_dep.dat**  
|                 | Contains x-, y-, z-, age, radius and settling velocity for the deposited gas particles.  
| 3               | Separates between evaporated and non evaporated particles in the same way as above for gas and non-gas particles. In addition to the four files in case one, in this case also the files:  
|                 | **NAMN_IN_evap_t.dat**  
|                 | Specified as in case 2 but for evaporated particles.  
|                 | **NAMN_IN_evap_xyz.dat**  
|                 | Specified as in case 2 but for evaporated particles.  
|                 | **NAMN_IN_evap_t_dep.dat**  
|                 | Specified as in case 2 but for evaporated particles.  
|                 | **NAMN_IN_evap_xyz_dep.dat**  
|                 | Specified as in case 2 but for evaporated particles.  

Table 7. Possible values for the variable creat_evap_file in the file input_ana.dat.
Table 8. Source position variables input_ana.nml.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th># of values</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_source</td>
<td>number of sources</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>XINL1</td>
<td>low x-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>XINL2</td>
<td>high x-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>YINL1</td>
<td>low y-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>YINL2</td>
<td>high y-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>ZINL1</td>
<td>low z-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>ZINL2</td>
<td>high z-coordinates for the sources</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td># of values</td>
<td>Unit</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>n_z0</td>
<td>number of roughness areas. I.e. number of different surfaces in the x-direction. In the unstable case ( n_z0 = 1 )</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>X_Z0</td>
<td>x-coordinates where a new ( z0 ) value becomes valid. The first X_Z0 value must be less than the lowest x-value in the simulation.</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>ZROUGH</td>
<td>Roughness length for the different surfaces. The height above the ground where the mean wind becomes zero.</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>UST</td>
<td>Friction velocity.</td>
<td>n_z0</td>
<td>m/s</td>
</tr>
<tr>
<td>MO_L</td>
<td>(Monin-)Obukhovs length</td>
<td>n_z0</td>
<td>m</td>
</tr>
<tr>
<td>N_FREQ</td>
<td>Brunt-Väisälä frequency</td>
<td>n_z0</td>
<td>1/s</td>
</tr>
<tr>
<td>F_PARAM</td>
<td>Coriolis parameter</td>
<td>n_z0</td>
<td>1/s</td>
</tr>
<tr>
<td>WERT_VEL</td>
<td>Large scale vertical velocity</td>
<td>n_z0</td>
<td>m/s</td>
</tr>
<tr>
<td>ground_temp</td>
<td>temperature at ground</td>
<td>n_z0</td>
<td>K</td>
</tr>
<tr>
<td>h_start</td>
<td>Start value for unstable boundary height calculation</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>t_end</td>
<td>Length of time period during which the unstable boundary height grows</td>
<td>1</td>
<td>h</td>
</tr>
<tr>
<td>gamma</td>
<td>the stratification above the PBL, potential temperature</td>
<td>1</td>
<td>K/m</td>
</tr>
<tr>
<td>XHIGH</td>
<td>the largest x-value in the calculation domain</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>ZLOW</td>
<td>the lowest z-value in the calculation domain</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>VD</td>
<td>the deposition velocity due to transport through the laminar surface layer and surface resistance</td>
<td>1</td>
<td>m/s</td>
</tr>
<tr>
<td>AGE0</td>
<td>age of particles at the release</td>
<td>n_source</td>
<td>s</td>
</tr>
<tr>
<td>WIND_ANGLE</td>
<td>rotate the wind clockwise in the xy-plane</td>
<td>1</td>
<td>°</td>
</tr>
<tr>
<td>RELEASE_FACT</td>
<td>Used to modulate evaporated particles release height</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 10. Namelist method in input_ana.nml.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th># of values</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFCOEF</td>
<td>Switch for LEM/RDM. Not in use. Must be = 1 in this model</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>DTMAX</td>
<td>Maximum time step</td>
<td>1</td>
<td>s</td>
</tr>
<tr>
<td>TOT_TIME</td>
<td>Time limit in the calculations</td>
<td>1</td>
<td>s</td>
</tr>
<tr>
<td>PDENS</td>
<td>release rate for particles</td>
<td>n_source</td>
<td>1/s</td>
</tr>
<tr>
<td>PTK_ST</td>
<td>source release start</td>
<td>n_source</td>
<td>s</td>
</tr>
<tr>
<td>PTK_END</td>
<td>source release stop</td>
<td>n_source</td>
<td>s</td>
</tr>
<tr>
<td>EVAPORATE</td>
<td>if true then deposited fluid particles can evaporate. If false then evaporation from the ground deposition layer is ignored as in previous models.</td>
<td>l</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 11. Namelist moving-source in input_ana.nml.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th># of values</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_waypoints</td>
<td>number of waypoints. Set to 0 if the sources should be fix. The waypoints specifies trajectories for the sources</td>
<td>1</td>
<td>–</td>
</tr>
<tr>
<td>X_WAYPOINT</td>
<td>x position of waypoints</td>
<td>n_waypoints</td>
<td>m</td>
</tr>
<tr>
<td>Y_WAYPOINT</td>
<td>y position of waypoints</td>
<td>n_waypoints</td>
<td>m</td>
</tr>
<tr>
<td>Z_WAYPOINT</td>
<td>z position of waypoints</td>
<td>n_waypoints</td>
<td>m</td>
</tr>
<tr>
<td>T_WAYPOINT</td>
<td>time to reach waypoint</td>
<td>n_waypoints</td>
<td>s</td>
</tr>
</tbody>
</table>

Table 12. Namelist settling in input_ana.nml.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th># of values</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>standard deviation of particle radius</td>
<td>n_source</td>
<td>–</td>
</tr>
<tr>
<td>r0</td>
<td>median particle radii</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>rmax</td>
<td>maximum radius in the distribution</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>rmin</td>
<td>minimum radius in the distribution</td>
<td>n_source</td>
<td>m</td>
</tr>
<tr>
<td>p</td>
<td>air pressure used to calculate settling velocity</td>
<td>n_z0</td>
<td>Pa</td>
</tr>
<tr>
<td>T</td>
<td>air temperature used to calculate settling velocity</td>
<td>n_z0</td>
<td>K</td>
</tr>
<tr>
<td>rho_part</td>
<td>density of the particles</td>
<td>n_source</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>g_earth</td>
<td>acceleration due to gravity</td>
<td>l</td>
<td>m/s$^2$</td>
</tr>
<tr>
<td>ppact_factor</td>
<td>evaporated particles per particle from fluid drop source</td>
<td>n_source</td>
<td>–</td>
</tr>
<tr>
<td>resusp_fact</td>
<td>particle type, gas=0, fluid=1, solid=2. The current version of the model only calculates evaporation from deposited fluid particles. Different particle types are treated in the same way except from evaporation</td>
<td>n_source</td>
<td>–</td>
</tr>
</tbody>
</table>
Table 13. Namelist *utdata* in input_*ana.nml*.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th># of values</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT_UT</td>
<td>time interval for output</td>
<td>1</td>
<td>s</td>
</tr>
<tr>
<td>NAMN_IN</td>
<td>carries basic part of the name of output files</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>name of evaporate data table</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CREATE_EVAP_FILE</td>
<td>see table 7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>READ_IN_FILE</td>
<td>A file containing particle information from previous run. If empty or don’t exist then no file will be read implying that the simulation should start from scratch.</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>SAVE_OUT_FILE</td>
<td>A file for storing the particles status at the end of the run. If empty no file will be created. This file can be read by READ_IN_FILE.</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>
Modeling of dispersion, deposition and evaporation from ground deposition in a stochastic particle model

Abstract
The...
Modellering av spridning, deposition och avdunstning från markbeläggning i en stokastisk partikelmodell