HAL
open science

Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2

A. Stohl, C. Forster, A. Frank, P. Seibert, G. Wotawa

To cite this version:

A. Stohl, C. Forster, A. Frank, P. Seibert, G. Wotawa. Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2. Atmospheric Chemistry and Physics, 2005, 5 (9), pp.2461-2474. hal-00295743

HAL Id: hal-00295743

https://hal.science/hal-00295743v1

Submitted on 18 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Atmos. Chem. Phys., 5, 2461–2474, 2005
www.atmos-chem-phys.org/acp/5/2461/
SRef-ID: 1680-7324/acp/2005-5-2461
European Geosciences Union

Atmospheric Chemistry and Physics

Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2

A. Stohl[1], **C. Forster**[1], **A. Frank**[2], **P. Seibert**[2], **and G. Wotawa**[3]

[1]Norwegian Institute for Air Research, Kjeller, Norway
[2]Institute of Meteorology, University of Natural Resources and Applied Life Sciences, Vienna, Austria
[3]Preparatory Commission for the Comprehensive Nuclear Test Ban Treaty Organization, Vienna, Austria

Received: 21 April 2005 – Published in Atmos. Chem. Phys. Discuss.: 13 July 2005
Revised: 7 September 2005 – Accepted: 9 September 2005 – Published: 21 September 2005

Abstract. The Lagrangian particle dispersion model FLEXPART was originally (about 8 years ago) designed for calculating the long-range and mesoscale dispersion of air pollutants from point sources, such as after an accident in a nuclear power plant. In the meantime FLEXPART has evolved into a comprehensive tool for atmospheric transport modeling and analysis. Its application fields were extended from air pollution studies to other topics where atmospheric transport plays a role (e.g., exchange between the stratosphere and troposphere, or the global water cycle). It has evolved into a true community model that is now being used by at least 25 groups from 14 different countries and is seeing both operational and research applications. A user manual has been kept actual over the years and was distributed over an internet page along with the model's source code. In this note we provide a citeable technical description of FLEXPART's latest version (6.2).

1 Introduction

Lagrangian particle models compute trajectories of a large number of so-called particles (not necessarily representing real particles, but infinitesimally small air parcels) to describe the transport and diffusion of tracers in the atmosphere. The main advantage of Lagrangian models is that, unlike in Eulerian models, there is no numerical diffusion. Furthermore, in Eulerian models a tracer released from a point source is instantaneously mixed within a grid box, whereas Lagrangian models are independent of a computational grid and have, in principle, infinitesimally small resolution.

The basis for current atmospheric particle models was laid by Thomson (1987), who stated the criteria that must be fulfilled in order for a model to be theoretically correct. A monograph on the theory of stochastic Lagrangian models was published by Rodean (1996) and another good review was written by Wilson and Sawford (1996). The theory of modeling dispersion backward in time with Lagrangian particle models was developed by Flesch et al. (1995) and Seibert and Frank (2004). Reviews of the more practical aspects of particle modeling were provided by Zannetti (1992) and Uliasz (1994).

This note describes FLEXPART, a Lagrangian particle dispersion model that simulates the long-range and mesoscale transport, diffusion, dry and wet deposition, and radioactive decay of tracers released from point, line, area or volume sources. It can also be used in a domain-filling mode where the entire atmosphere is represented by particles of equal mass. FLEXPART can be used forward in time to simulate the dispersion of tracers from their sources, or backward in time to determine potential source contributions for given receptors. The management of input data was largely taken from FLEXTRA, a kinematic trajectory model (Stohl et al., 1995). FLEXPART's first version was developed during the first author's military service at the nuclear-biological-chemical school of the Austrian Forces, the deposition code was added soon later (version 2), and this version was validated using data from three large tracer experiments (Stohl et al., 1998). Version 3 saw performance optimizations and the development of a density correction (Stohl and Thomson, 1999). Further updates included the addition of a convection scheme (Seibert et al., 2001) (version 4), better backward calculation capabilities (Seibert and Frank, 2004), and improvements in the input/output handling (version 5). Validation was done during intercontinental air pollution transport studies (Stohl and Trickl, 1999; Forster et al., 2001; Spichtinger et al., 2001; Stohl et al., 2002, 2003), for which also special developments for FLEXPART were made in order to extend the forecasting capabilities (Stohl et al., 2004). The most recent version described here is 6.2, which saw corrections

Correspondence to: A. Stohl
(ast@nilu.no)

© 2005 Author(s). This work is licensed under a Creative Commons License.

to the numerics in the convection scheme, the addition of a domain-filling option, and the possibility to use output nests.

FLEXPART is coded following the Fortran 77 standard and tested with several compilers (gnu, Absoft, Portland Group) under a number of operating systems (Linux, Solaris, etc.). The code is carefully documented and optimized for run-time performance. No attempts have been made to parallelize the code because the model is strictly linear and, therefore, it is most effective to partition problems such that they run on single processors and to combine the results if needed.

FLEXPART's source code and a manual are freely available from the internet page http://zardoz.nilu.no/~andreas/flextra+flexpart.html. According to a recent user survey, at least 25 groups from 14 countries are currently using FLEXPART. The version of FLEXPART described here is based on model level data of the numerical weather prediction model of the European Centre for Medium-Range Weather Forecasts (ECMWF). Other users have developed FLEXPART versions using input data from a suite of different global (e.g., from the National Centers of Environmental Prediction) and meso-scale (e.g., MM5) models, some of which are available from the FLEXPART website but are not described here.

2 Input data and grid definitions

FLEXPART is an off-line model that uses meteorological fields (analyses or forecasts) in Gridded Binary (GRIB) format from the ECMWF numerical weather prediction model (ECMWF, 1995) on a latitude/longitude grid and on native ECMWF model levels as input. The data can be retrieved from the ECMWF archives using a pre-processor that is also available from the FLEXPART website but not described here. The GRIB decoding software is not provided with FLEXPART but is publicly available (see links on the FLEXPART website). The data can be global or only cover a limited area. Furthermore, higher-resolution domains can be nested into a mother domain.

The file `includepar` contains all relevant FLEXPART parameter settings, both physical constants and maximum field dimensions. As the memory required by FLEXPART is determined by the various field dimensions, it is recommended that they are adjusted to actual needs before compilation. The file `includecom` defines all FLEXPART global variables and fields, i.e., those shared between most subroutines.

2.1 Input data organisation

A file `pathnames` must exist in the directory where FLEXPART is started. It must contain at least four lines:
1. line: Directory where all the FLEXPART command files are stored.
2. line: Directory to which the model output is written.
3. line: Directory where the GRIB input fields are located.
4. line: Path name of the `AVAILABLE` file (see below).
If nests with higher-resolution input data shall also be used, lines 3 and 4 must be repeated for every nest, thus also specifying the nesting level order. Any number of nesting levels can be used up to a maximum (parameter `maxnests`).

The meteorological input data must be organised such that all data for a domain and a certain date must be contained in a single GRIB file. The `AVAILABLE` file lists all available dates and the corresponding file names. For each nesting level, the input files must be stored in a different directory and the `AVAILABLE` file must contain the same dates as for the mother domain. Given a certain particle position, the last (i.e., innermost) nest is checked first whether it contains the particle or not. If the particle resides in this nest, the meteorological data from this nest is interpolated linearly to the particle position. If not, the next nest is checked, and so forth until the mother domain is reached. There is no nesting in the vertical direction and the poles must not be contained in any nest.

The maximum dimensions of the meteorological fields are specified by the parameters `nxmax, nymax, nuvzmax, nwzmax, nzmax` in file `includepar`, for x, y, and three z dimensions, respectively. The three z dimensions are for the original ECMWF data (`nuvzmax, nwzmax` for model half levels and model levels, respectively) and transformed data (`nzmax`, see below), respectively. The horizontal dimensions of the nests must be smaller than the parameters `nxmaxn, nymaxn`. Grid dimensions and other basic things are checked in routine `gridcheck.f`, and error messages are issued if necessary.

The longitude/latitude range of the mother grid is also used as the computational domain. All internal FLEXPART coordinates run from the western/southern domain boundary with coordinates (0,0) to the eastern/northern boundary with coordinates (nx-1,ny-1), where (nx,ny) are the mother grid dimensions. For global input data, FLEXPART repeats the westernmost grid cells at the easternmost domain "boundary", in order to facilitate interpolation on all locations of the globe (e.g., if input data run from 0 to 359° with 1° resolution, 0° data are repeated at 360°). A global mother domain can be shifted by `nxshift` (file `includepar`) data columns (subroutines `shift_field.f` and `shift_field_0.f`) if nested input fields would otherwise overlap the "boundaries". For instance, a domain stretching from 320° to 30° can be nested into the mother grid of the above example by shifting the mother grid by 30°.

2.2 Vertical model structure and required data

FLEXPART needs five three-dimensional fields: horizontal and vertical wind components, temperature and specific humidity. Input data must be on ECMWF model (i.e., η) levels which are defined by a hybrid coordinate system.

The conversion from η to pressure coordinates is given by $p_k = A_k + B_k p_s$ and the heights of the η surfaces are defined by $\eta_k = A_k / p_0 + B_k$, where η_k is the value of η at the kth model level, p_s is the surface pressure and $p_0 = 101325$ Pa. A_k and B_k are coefficients, chosen such that the levels closest to the ground follow the topography, while the highest levels coincide with pressure surfaces; intermediate levels transition between the two. The vertical wind in hybrid coordinates is calculated mass-consistently from spectral data by the preprocessor. A surface level is defined in addition to the regular η levels. 2 m temperature, 10 m winds and specific humidity from the first regular model level are assigned to this level, to represent "surface" values.

Parameterized random velocities in the atmospheric boundary layer (ABL, see Sects. 3 and 4) are calculated in units of m s^{-1}, and not in η coordinates. Therefore, in order to avoid time-consuming coordinate transformations every time step, all three-dimensional data are interpolated linearly from the ECMWF model levels to terrain-following Cartesian coordinates $\tilde{z} = z - z_t$, where z_t is the height of the topography (subroutine `verttransform.f`). The conversion of vertical wind speeds from the eta coordinate system into the terrain-following co-ordinate system follows as

$$\tilde{w} = \dot{\tilde{z}} = \dot{\tilde{\eta}} \left(\frac{\partial p}{\partial z} \right)^{-1} + \left. \frac{\partial \tilde{z}}{\partial t} \right|_{\eta} + \boldsymbol{v}_h \cdot \nabla_{\eta} \tilde{z} \tag{1}$$

where $\dot{\tilde{\eta}} = \dot{\eta} \partial p / \partial \eta$. The second term on the right hand side is missing in the FLEXPART transformation because it is much smaller than the others. One colleague has implemented this term in his version of FLEXPART and found virtually no differences (B. Legras, personal communication).

FLEXPART also needs the two-dimensional fields: surface pressure, total cloud cover, 10 m horizontal wind components, 2 m temperature and dew point temperature, large scale and convective precipitation, sensible heat flux, east/west and north/south surface stress, topography, land-sea-mask and subgrid standard deviation of topography. A landuse inventory (for Europe, the data set of Velde et al. (1994) is used) must be provided in an extra file (`landuse.asc`).

3 Physical parameterization of boundary layer parameters

Accumulated surface sensible heat fluxes and surface stresses are available from ECMWF forecasts. The pre-processor selects the shortest forecasts available for that date from the ECMWF archives and deaccumulates the flux data. The total surface stress is computed from

$$\tau = \sqrt{\tau_1^2 + \tau_2^2} \ , \tag{2}$$

where τ_1 and τ_2 are the surface stresses in east/west and north/south direction, respectively. Friction velocity is then calculated in subroutine `scalev.f` as

$$u_* = \sqrt{\tau / \rho} \ , \tag{3}$$

where ρ is the air density (Wotawa et al., 1996). Friction velocities and heat fluxes calculated using this method are most accurate (Wotawa and Stohl, 1997). However, if deaccumulated surface stresses and surface sensible heat fluxes are not available, the profile method after Berkowicz and Prahm (1982) (subroutine `pbl_profile.f`) is applied to wind and temperature data at the second model level and at 10 m (for wind) and 2 m (for temperature) (note that previously the first model level was used; as ECMWF has its first model level now close to 10 m, the second level is used instead). The following three equations are solved iteratively:

$$u_* = \frac{\kappa \Delta u}{\ln \frac{z_l}{10} - \Psi_m(\frac{z_l}{L}) + \Psi_m(\frac{10}{L})} \ , \tag{4}$$

$$\Theta_* = \frac{\kappa \Delta \Theta}{0.74 \left[\ln \frac{z_l}{2} - \Psi_h(\frac{z_l}{L}) + \Psi_h(\frac{2}{L}) \right]} \ , \tag{5}$$

$$L = \frac{\overline{T} u_*^2}{g \kappa \Theta_*} \ , \tag{6}$$

where κ is the von Kármán constant (0.4), z_l is the height of the second model level, Δu is the difference between wind speed at the second model level and at 10 m, $\Delta\Theta$ is the difference between potential temperature at the second model level and at 2 m, Ψ_m and Ψ_h are the stability correction functions for momentum and heat (Businger et al., 1971; Beljaars and Holtslag, 1991), g is the acceleration due to gravity, Θ_* is the temperature scale and $\overline{T}$ is the average surface layer temperature (taken as T at the first model level). The heat flux is then computed by

$$(\overline{w' \Theta_v'})_0 = -\rho c_p u_* \Theta_* \ , \tag{7}$$

where ρc_p is the specific heat capacity of air at constant pressure.

ABL heights are calculated according to Vogelezang and Holtslag (1996) using the critical Richardson number concept (subroutine `richardson.f`). The ABL height h_{mix} is set to the height of the first model level l for which the Richardson number

$$Ri_l = \frac{(g/\Theta_{v1})(\Theta_{vl} - \Theta_{v1})(z_l - z_1)}{(u_l - u_1)^2 + (v_l - v_1)^2 + 100 u_*^2} \ , \tag{8}$$

exceeds the critical value of 0.25. Θ_{v1} and Θ_{vl} are the virtual potential temperatures, z_1 and z_l are the heights of, and (u_1, v_1), and (u_l, v_l) are the wind components at the 1st and l^{th} model level, respectively. The formulation of Eq. 8 can be improved for convective situations by replacing Θ_{v1} with

$$\Theta_{v1}' = \Theta_{v1} + 8.5 \frac{(\overline{w' \Theta_v'})_0}{w_* c_p} \ , \tag{9}$$

where

$$w_* = \left[\frac{(\overline{w'\Theta'_v})_0 g h_{\mathrm{mix}}}{\Theta_{v1} c_p} \right]^{1/3} \tag{10}$$

is the convective velocity scale. The second term on the right hand side of Eq. 9 represents a temperature excess of rising thermals. As w_* is unknown beforehand, h_{mix} and w_* are calculated iteratively.

Spatial and temporal variations of ABL heights on scales not resolved by the ECMWF model play an important role in determining the thickness of the layer over which tracer is effectively mixed. The height of the convective ABL reaches its maximum value (say 1500 m) in the afternoon (say, at 1700 local time (LT)), before a much shallower stable ABL forms. Now, if meteorological data are available only at 12:00 and 18:00 LT and the ABL heights at those times are, say, 1200 m and 200 m, and linear interpolation is used, the ABL height at 17:00 LT is significantly underestimated (370 m instead of 1500 m). If tracer is released at the surface shortly before the breakdown of the convective ABL, this would lead to a serious overestimation of the surface concentrations (a factor of four in the above example). Similar arguments hold for spatial variations of ABL heights due to complex topography and variability in landuse or soil wetness (Hubbe et al., 1997). The thickness of a tracer cloud traveling over such a patchy surface would be determined by the maximum rather than by the average ABL height.

In FLEXPART a somewhat arbitrary parameterization is used to avoid a significant bias in the tracer cloud thickness and the surface tracer concentrations. To account for spatial variations induced by topography, we use an "envelope" ABL height

$$H_{\mathrm{env}} = h_{\mathrm{mix}} + \min\left[\sigma_Z, c\frac{V}{N}\right] . \tag{11}$$

Here, σ_Z is the standard deviation of the ECMWF model subgrid topography, c is a constant (here: 2.0), V is the wind speed at height h_{mix}, and N is the Brunt-Vaisala frequency. Under convective conditions, the envelope ABL height is, thus, the diagnosed ABL height plus the subgrid topography (assuming that the ABL height over the hill tops effectively determines the dilution of a tracer cloud located in a convective ABL). Under stable conditions, air tends to flow around topographic obstacles rather than above it, but some lifting is possible due to the available kinetic energy. $\frac{V}{N}$ is the local Froude number (i.e., the ratio of inertial to buoyant forces) times the length scale of the sub-grid topographic obstacle. The factor $c\frac{V}{N}$, thus, limits the effect of the subgrid topography under stable conditions, with c=2 being a subjective scaling factor. H_{env} rather than h_{mix} is used for all subsequent calculations. In addition, H_{env} is not interpolated to the particle position, but the maximum H_{env} of the grid points surrounding a particle's position in space and time is used.

4 Particle transport and diffusion

4.1 Particle trajectory calculations

FLEXPART generally uses the simple "zero acceleration" scheme

$$\boldsymbol{X}(t+\Delta t) = \boldsymbol{X}(t) + \boldsymbol{v}(\boldsymbol{X}, t)\Delta t , \tag{12}$$

which is accurate to the first order, to integrate the trajectory equation (Stohl, 1998)

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{v}[\boldsymbol{X}(t)] , \tag{13}$$

with t being time, Δt the time increment, $\boldsymbol{X}$ the position vector, and $\boldsymbol{v}=\overline{\boldsymbol{v}}+\boldsymbol{v}_t+\boldsymbol{v}_m$ the wind vector that is composed of the grid scale wind $\overline{\boldsymbol{v}}$, the turbulent wind fluctuations $\boldsymbol{v}_t$ and the mesoscale wind fluctuations $\boldsymbol{v}_m$.

Since FLEXPART version 5.0, numerical accuracy has been improved by making one iteration of the Petterssen (1940) scheme (which is accurate to the second order) whenever this is possible, but only for the grid-scale winds. It is implemented as a correction applied to the position obtained with the "zero acceleration" scheme. In three cases it cannot be applied. First, the Petterssen scheme needs winds at a second time which may be outside the time interval of the two wind fields kept in memory. Second, if a particle crosses the boundaries of nested domains, and third in the ABL if `ctl`>0 (see below).

Particle transport and turbulent dispersion are handled by the subroutine `advance.f` where calls are issued to procedures that interpolate winds and other data to the particle position and the Langevin equations (see below) are solved. The poles are singularities on a latitude/longitude grid. Thus, horizontal winds (variables `uu,vv`) poleward of latitudes (`switchnorth, switchsouth`) are transformed to a polar stereographic projection (variables `uupol,vvpol`) on which particle advection is calculated. As `uupol,vvpol` are also stored on the latitude/longitude grid, no additional interpolation is made.

4.2 The Langevin equation

Turbulent motions $\boldsymbol{v}_t$ for wind components i are parameterized assuming a Markov process based on the Langevin equation (Thomson, 1987)

$$dv_{t_i} = a_i(\boldsymbol{x}, \boldsymbol{v}_t, t)dt + b_{ij}(\boldsymbol{x}, \boldsymbol{v}_t, t)dW_j , \tag{14}$$

where the drift term a and the diffusion term b are functions of the position, the turbulent velocity and time. dW_j are incremental components of a Wiener process with mean zero and variance dt, which are uncorrelated in time (Legg and Raupach, 1982). Cross-correlations between the different wind components are also not taken into account, since they have little effect for long-range dispersion (Uliasz, 1994).

Gaussian turbulence is assumed in FLEXPART, which is strictly valid only for stable and neutral conditions. Under convective conditions, when turbulence is skewed and larger areas are occupied by downdrafts than by updrafts, this assumption is violated, but for transport distances where particles are rather well mixed throughout the ABL, the error is minor.

With the above assumptions, the Langevin equation for the vertical wind component w can be written as

$$dw = -w\frac{dt}{\tau_{L_w}} + \frac{\partial\sigma_w^2}{\partial z}dt + \frac{\sigma_w^2}{\rho}\frac{\partial\rho}{\partial z}dt + \left(\frac{2}{\tau_{L_w}}\right)^{1/2}\sigma_w\, dW\ , \tag{15}$$

where w and σ_w are the turbulent vertical wind component and its standard deviation, τ_{L_w} is the Lagrangian timescale for the vertical velocity autocorrelation and ρ is density. The second and the third term on the right hand side are the drift correction (McNider et al., 1988) and the density correction (Stohl and Thomson, 1999), respectively. This Langevin equation is identical to the one described by Legg and Raupach (1982), except for the term from Stohl and Thomson (1999) which accounts for the decrease of air density with height.

Alternatively, the Langevin equation can be re-expressed in terms of w/σ_w instead of w (Wilson et al., 1983):

$$d\left(\frac{w}{\sigma_w}\right) = -\frac{w}{\sigma_w}\frac{dt}{\tau_{L_w}} + \frac{\partial\sigma_w}{\partial z}dt + \frac{\sigma_w}{\rho}\frac{\partial\rho}{\partial z}dt + \left(\frac{2}{\tau_{L_w}}\right)^{1/2}dW\ , \tag{16}$$

This form was shown by Thomson (1987) to fulfill the well-mixed criterion which states that "if a species of passive marked particles is initially mixed uniformly in position and velocity space in a turbulent flow, it will stay that way" (Rodean, 1996). Although the method proposed by Legg and Raupach (1982) violates this criterion in strongly inhomogeneous turbulence, their formulation was found to be practical, as numerical experiments have shown that it is more robust against an increase in the integration time step. Therefore, Eq. (15) is used with long time steps (see Sect. 4.3); otherwise, Eq. (16) is used. For the horizontal wind components, the Langevin equation is identical to Eq. (15), with no drift and density correction terms.

For the discrete time step implementation of the above Langevin equations (at the example of Eq. 16), two different methods are used. When $\left(\Delta t/\tau_{L_w}\right) \geq 0.5$,

$$\left(\frac{w}{\sigma_w}\right)_{k+1} = r_w\left(\frac{w}{\sigma_w}\right)_k + \frac{\partial\sigma_w}{\partial z}\tau_{L_w}(1-r_w) + \frac{\sigma_w}{\rho}\frac{\partial\rho}{\partial z}\tau_{L_w}(1-r_w) + \left(1-r_w^2\right)^{1/2}\zeta\ , \tag{17}$$

where $r_w = \exp(-\Delta t/\tau_{L_w})$ is the autocorrelation of the vertical wind, and ζ is a normally distributed random number with mean zero and unit standard deviation. The subscripts k and $k+1$ refer to subsequent times separated by Δt.

To save computation time for cases when $\left(\Delta t/\tau_{L_w}\right) < 0.5$, the following first order approximation is used in order to avoid the computation of the exponential function:

$$\left(\frac{w}{\sigma_w}\right)_{k+1} = \left(1-\frac{\Delta t}{\tau_{L_w}}\right)\left(\frac{w}{\sigma_w}\right)_k + \frac{\partial\sigma_w}{\partial z}\Delta t + \frac{\sigma_w}{\rho}\frac{\partial\rho}{\partial z}\Delta t + \left(\frac{2\Delta t}{\tau_{L_w}}\right)^{1/2}\zeta\ . \tag{18}$$

When a particle reaches the surface or the top of the ABL, it is reflected and the sign of the turbulent velocity is changed (Wilson and Flesch, 1993).

4.3 Determination of the time step

FLEXPART can be used in two different modes. The computationally faster one (`ctl`<0 in file `COMMAND`) does not adapt the computation time step to the Lagrangian timescales τ_{L_i} (where i is one of the three wind components) and FLEXPART uses constant time steps of one synchronisation time interval (`lsynctime`, specified in file `COMMAND`, typically 900 seconds). Usually, autocorrelations are very low in this mode and turbulence is not described well. Nevertheless, for large scale applications FLEXPART works very well with this option (Stohl et al., 1998). If turbulence shall be described more accurately, the time steps must be limited by τ_L. Since the vertical wind is most important, only τ_{L_w} is used for this. The user must specify two constants, `ctl` and `ifine` in file `COMMAND`. The first one determines the time step Δt_i according to

$$\Delta t_i = \frac{1}{c_{tl}}\min\left(\tau_{L_w}\ ,\ \frac{h}{2w}\ ,\ \frac{0.5}{\partial\sigma_w/\partial z}\right)\ . \tag{19}$$

The minimum value of Δt_i is 1 second. Δt_i is used for solving the Langevin equations for the horizontal turbulent wind components.

For solving the Langevin equation for the vertical wind component, a shorter time step $\Delta t_w = \Delta t_i/$`ifine` is used. However, note that there is no interaction between horizontal and vertical wind components on timescales less than Δt_i. This strategy (given sufficiently large values for `ctl` and `ifine`) ensures that the particles stay vertically well-mixed also in very inhomogeneous turbulence, while keeping the computational cost at a minimum.

4.4 Parameterization of the wind fluctuations

For σ_{v_i} and τ_{L_i} Hanna (1982) proposed a parameterization scheme based on the boundary layer parameters h, L, w_*, z_0 and u_*, i.e. ABL height, Monin-Obukhov length, convective velocity scale, roughness length and friction velocity, respectively. It is used in subroutines `hanna.f, hanna1.f, hanna_short.f` with a modification taken from Ryall et al. (1997) for σ_w, as Hanna's

scheme does not always yield smooth profiles of σ_w throughout the whole convective ABL. In the following, subscripts u and v refer to the along-wind and the cross-wind components (transformed to grid coordinates in subroutine `windalign.f`), respectively, and w to the vertical component of the turbulent velocities; f is the Coriolis parameter. The minimum τ_{L_u}, τ_{L_v} and τ_{L_w} used are 10 s, 10 s and 30 s, respectively, in order to avoid excessive computation times for particles close to the surface.

Unstable conditions:

$$\frac{\sigma_u}{u_*} = \frac{\sigma_v}{u_*} = \left(12 + \frac{h}{2|L|}\right)^{1/3} \tag{20}$$

$$\tau_{L_u} = \tau_{L_v} = 0.15\frac{h}{\sigma_u} \tag{21}$$

$$\frac{\sigma_w}{w_*} = \left[1.2\left(1 - 0.9\frac{z}{h}\right)\left(\frac{z}{h}\right)^{2/3} + \left(1.8 - 1.4\frac{z}{h}\right)u_*^2\right]^{1/2} \tag{22}$$

For $z/h<0.1$ and $z-z_0>-L$:

$$\tau_{L_w} = 0.1\frac{z}{\sigma_w\left[0.55 - 0.38\left(z - z_0\right)/L\right]} \tag{23}$$

For $z/h<0.1$ and $z-z_0<-L$:

$$\tau_{L_w} = 0.59\frac{z}{\sigma_w} \tag{24}$$

For $z/h>0.1$:

$$\tau_{L_w} = 0.15\frac{h}{\sigma_w}\left[1 - \exp\left(\frac{-5z}{h}\right)\right] \tag{25}$$

Neutral conditions:

$$\frac{\sigma_u}{u_*} = 2.0\exp(-3fz/u_*) \tag{26}$$

$$\frac{\sigma_v}{u_*} = \frac{\sigma_w}{u_*} = 1.3\exp(-2fz/u_*) \tag{27}$$

$$\tau_{L_u} = \tau_{L_v} = \tau_{L_w} = \frac{0.5z/\sigma_w}{1 + 15fz/u_*} \tag{28}$$

Stable conditions:

$$\frac{\sigma_u}{u_*} = 2.0\left(1 - \frac{z}{h}\right) \tag{29}$$

$$\frac{\sigma_v}{u_*} = \frac{\sigma_w}{u_*} = 1.3\left(1 - \frac{z}{h}\right) \tag{30}$$

$$\tau_{L_u} = 0.15\frac{h}{\sigma_u}\left(\frac{z}{h}\right)^{0.5} \tag{31}$$

$$\tau_{L_v} = 0.07\frac{h}{\sigma_v}\left(\frac{z}{h}\right)^{0.5} \tag{32}$$

$$\tau_{L_w} = 0.1\frac{h}{\sigma_w}\left(\frac{z}{h}\right)^{0.5} \tag{33}$$

Lacking suitable turbulence parameterizations above the ABL ($z>h$), a constant vertical diffusivity D_z=0.1 m^2 s^{-1} is used in the stratosphere, following recent work of Legras et al. (2003), whereas a horizontal diffusivity D_h=50 m^2 s^{-1} is used in the free troposphere. Stratosphere and troposphere are distinguished based on a threshold of 2 pvu (potential vorticity units). Diffusivities are converted into velocity scales using $\sigma_{v_i}=\sqrt{D_i/dt}$.

4.5 Mesoscale velocity fluctuations

Mesoscale motions are neither resolved by the ECMWF data nor covered by the turbulence parameterization. This unresolved spectral interval needs to be taken into account at least in an approximate way, since mesoscale motions can significantly accelerate the growth of a dispersing plume (Gupta et al., 1997). For this, we use a similar method as Maryon (1998), namely to solve an independent Langevin equation for the mesoscale wind velocity fluctuations ("meandering" in Maryon's terms). Assuming that the variance of the wind at the grid scale provides some information on its subgrid variance, the wind velocity standard deviation used for the mesoscale Langevin equation is set to `turbmesoscale` (set in file `includepar`) times the standard deviation of the grid points surrounding the particle's position. The corresponding time scale is taken as half the interval at which wind fields are available, assuming that the linear interpolation between the grid points can recover half the subgrid variability, not an unlikely assumption (Stohl et al., 1995). This empirical approach does not describe actual mesoscale phenomena, but it is similar to the ensemble methods used to assess trajectory accuracy (Kahl , 1996; Baumann and Stohl, 1997; Stohl, 1998).

4.6 Moist convection

An important transport mechanism are the updrafts in convective clouds. They occur in conjunction with downdrafts within the clouds and compensating subsidence in the cloud-free surroundings. These convective transports are grid-scale in the vertical, but sub-grid scale in the horizontal, and are not represented by the ECMWF vertical velocity.

To represent convective transport in a particle dispersion model, it is necessary to redistribute particles in the entire vertical column. For FLEXPART we chose the convective parameterization scheme by Emanuel and Živković-Rothman (1999), as it relies on the grid-scale temperature and humidity fields and calculates a displacement matrix providing the necessary mass flux information for the particle redistribution. The convective parameterization is switched on using `lconvection` in file `COMMAND`. It's computation time scales to the square of the number of vertical model levels and may account for up to 70% of FLEXPART's computation time using current 60-level ECMWF data.

The convection is computed within the subroutines `convmix.f`, `calcmatrix.f`, `convect43c.f`, and `redist.f`. It is called every FLEXPART `lsynctime` time step (typically 900 s) with time-interpolated temperature and specific humiditiy profiles from the ECMWF data. Note that the original ECMWF model levels, not the Cartesian coordinates, are used in the convection scheme. For efficiency reasons, particles are sorted according to their horizontal grid positions (`sort2.f`) before calling the convection scheme once per grid column.

In the Emanuel scheme (`convect43c.f`), convection is triggered whenever

$$T_{vp}^{LCL+1} \geq T_v^{LCL+1} + T_{thres} \tag{34}$$

with T_{vp}^{LCL+1} the virtual temperature of a surface air parcel lifted to the level above the lifting condensation level LCL, T_v^{LCL+1} the virtual temperature of the environment there, and T_{thres}=0.9 K a threshold temperature value. Based on the buoyancy sorting principle (Emanuel, 1991; Telford, 1975), a matrix MA of the saturated upward and downward mass fluxes within clouds is calculated by accounting for entrainment and detrainment:

$$MA^{i,j} = \frac{M^i(|\sigma^{i,j+1} - \sigma^{i,j}| + |\sigma^{i,j} - \sigma^{i,j-1}|)}{(1-\sigma^{i,j}) \sum_{j=LCL}^{LNB} [|\sigma^{i,j+1} - \sigma^{i,j}| + |\sigma^{i,j} - \sigma^{i,j-1}|]} \tag{35}$$

Here $MA^{i,j}$ are the mass fractions displaced from level i to level j, M^i the mass fraction displaced from the surface to level i, LNB the level of neutral buoyancy of a surface air parcel and $0<\sigma^{i,j}<1$ the mixing fraction between level i and level j. The fraction $\sigma^{i,j}$ is determined by the environmental potential temperature θ^j, the liquid potential temperature $\theta_l^{i,j}$ of air displaced adiabatically from i to j, and the liquid potential temperature $\theta_{lp}^{i,j}$ of an air parcel first lifted adiabatically to level i and further to level j:

$$\sigma^{i,j} = \frac{\theta^j - \theta_{lp}^{i,j}}{\theta_l^{i,j} - \theta_{lp}^{i,j}} \tag{36}$$

By summing up over all levels j, we then calculate the saturated up- and downdrafts at each level i from Eq. (35) and assume that these fluxes are balanced by a subsidence mass flux in the environment.

The particles in each convectively active box are then redistributed (`redist.f`) according to the matrix MA. If the mass of an ECMWF model layer i is m^i and the mass flux from layer i to layer j accumulated over one time step is $\Delta MA^{i,j}$, then the probability of a particle to be moved from layer i to layer j is $\Delta MA^{i,j}/m^i$. Whether a given particle is displaced or not is determined by drawing a random number between [0,1], which also determines the position of the particle within the destination layer j. After the convective redistribution of the particles, the compensating subsidence mass fluxes are converted to a vertical velocity acting on those particles in the grid box that are not displaced by convective drafts. By calculating a subsidence velocity rather than displacing particles randomly between layers the scheme's numerical diffusion in the cloud-free environment is eliminated. The scheme was tested and fulfills the well-mixed criterion, i.e., if a tracer is well mixed in the whole atmospheric column, it remains so after the convection.

4.7 Particle splitting

During the initial phase of dispersion from a point source in the atmosphere, particles normally form a compact cloud. Relatively few particles suffice to simulate this initial phase correctly. After some time, however, the particle cloud gets distorted and particles spread over a much larger area. More particles are now needed. FLEXPART allows the user to specify a time constant Δt_s (file `COMMAND`). Particles are split into two (each of which receives half of the mass of the original particle) after travel times of Δt_s, $2\Delta t_s$, $4\Delta t_s$, $8\Delta t_s$, and so on (subroutine `timemanager.f`).

5 Forward and backward modeling

Normally, when FLEXPART is run forward in time (`ldirect=1` in file `COMMAND`), particles are released from one or a number of sources and concentrations are determined downwind on a grid. However, FLEXPART can also run backward in time (`ldirect=-1`), which is more efficient than forward modeling for calculating source-receptor relationships if the number of receptors is smaller than the number of (potential) sources. In the backward mode, particles are released from a receptor location (e.g., a measurement site) and a four-dimensional (3 space dimensions plus time) response function (sensitivity) to emission input is calculated.

Since this version (6.2) of FLEXPART, the calculation of the source-receptor relationships is generalized for both forward and backward runs, allowing much greater flexibility regarding the input and output units than before. `ind_source` and `ind_receptor` in file `COMMAND` switch between mass and mass mixing ratio units at the source and at the receptor, respectively. Note that source always stands for the physical source and not the location of the particle release, which is done at the source in forward mode but at the receptor in backward mode. Table 1 gives an overview of the units used in forward and backward modeling for different settings of the above switches. A "normal" forward simulation which specifies the release in mass units (i.e., kg) and also samples the output in mass units (i.e., a concentration in ng m^{-3}) requires both switches to be set to 1.

Table 1. Physical units of the input (in file RELEASES) and output data for forward (files grid_conc_date) and backward (files grid_time_date) runs for the various settings of the unit switches ind_source and ind_receptor (in both switches 1 refers to mass units, 2 to mass mixing ratio units).

Direction	ind_source	ind_receptor	input unit	output unit
Forward	1	1	kg	ng m^{-3}
Forward	1	2	kg	ppt by mass
Forward	2	1	1	ng m^{-3}
Forward	2	2	1	ppt by mass
Backward	1	1	1	s
Backward	1	2	1	s m^3 kg^{-1}
Backward	2	1	1	s kg m^{-3}
Backward	2	2	1	s

In the backward mode, any value not equal zero can be entered as the release "mass" in file `RELEASES` because the output is normalized by this value. The calculated response function is related to the particles' residence time in the output grid cells. The unit of the output response function varies, depending on how the switches are set. If `ind_source=1` and `ind_receptor=1`, the response function has the unit s. If this response function is folded (i.e., multiplied) with a 3-d field of emission mass fluxes into the output grid boxes (in kg m^{-3} s^{-1}), a concentration at the receptor (kg m^{-3}) is obtained. If `ind_source=1` and `ind_receptor=2`, the response function has the unit s m^3 kg^{-1} and if it is folded with the emission mass flux (again in kg m^{-3} s^{-1}), a mass mixing ratio at the receptor is obtained. The units of the response function for `ind_source=2` can be understood in analogy.

In the case of loss processes (dry or wet deposition, decay) the response function is "corrected" for these loss processes. See Seibert (2001) and, particularly, Seibert and Frank (2004) for a description of these generalized in- and output options and the implementation of backward modeling in FLEXPART. Seibert and Frank (2004) also describe the theory of backward modeling and give some examples, and Stohl et al. (2003) presents an application.

There are differences between the dimensions of the output fields needed to run FLEXPART forward and backward in time. For backward simulations, the output for all the receptors (particle release locations) must be kept separate, contrary to the forward runs where releases from several source locations can be combined. Therefore, for the backward runs the output fields must contain a dimension `maxpoint`, the maximum number of release points, that is not needed in forward runs. In order to avoid creating a further dimension for the output fields (thus increasing FLEXPART's memory demands), `maxspec`, the maximum number of chemical species used in forward runs, is replaced with `maxpoint` in the backward runs as a dimension of the output fields. This has the disadvantage that only one species (with certain properties regarding the removal processes, see below) can be calculated in a backward run. To switch between forward and backward runs, the parameter `maxpointspec` is used. It must be set (in `includecom`) to `maxspec` for forward runs and to `maxpoint` for backward runs. FLEXPART must be recompiled upon changing this.

6 Plume trajectories

In a recent paper, Stohl et al. (2002) proposed a method to condense the complex and large FLEXPART output using a cluster analysis (Dorling et al., 1992). The idea behind this is to cluster, at every output time, the positions of all particles originating from a release point, and write out only clustered particle positions, along with additional information (e.g., fraction of particles in the ABL and in the stratosphere). This creates information that is almost as compact as traditional trajectories but accounts for turbulence and convection. This option can be activated by setting `iout` to 4 or 5 in file `COMMAND`. The number of clusters can be set with the parameter `ncluster` in file `includepar`. The clustering is handled and output is produced by subroutine `plumetraj.f`.

7 Removal processes

FLEXPART takes into account radioactive (or other) decay, wet deposition, and dry deposition by reducing a particle's mass. However, as atmospheric transport is the same for all chemical species, a single particle can represent several (up to `maxspec`) chemical species, each affected differently by the removal processes.

7.1 Radioactive decay

Radioactive decay is accounted for by reducing the particle mass according to

$$m(t+\Delta t) = m(t)\exp(-\Delta t/\beta) \ , \tag{37}$$

where m is particle mass, and the time constant $\beta = T_{1/2}/\ln(2)$ is determined from the half life $T_{1/2}$ specified in file `SPECIES`. Deposited pollutant mass decays at the same rate.

7.2 Wet deposition

Wet deposition removes aerosols and gases from the atmosphere. In principle, in-cloud and below-cloud scavenging must be separated (Asman, 1995). However, as data on cloud base height and depth are not available, in-cloud and below-cloud scavenging are treated jointly in FLEXPART. Using

Table 2. Correction factors used for the calculation of the area fraction that experiences precipitation. Precipitation rates are in mm/hour.

	I_l and I_c				
Factor	$I \leq 1$	$1 < I \leq 3$	$3 < I \leq 8$	$8 < I \leq 20$	$20 < I$
fr_l	0.50	0.65	0.80	0.90	0.95
fr_c	0.40	0.55	0.70	0.80	0.90

scavenging coefficients, wet deposition takes the form of an exponential decay process (McMahon, 1979)

$$m(t + \Delta t) = m(t) \exp(-\Lambda \Delta t) \ , \tag{38}$$

where m and Λ are the particle mass and the scavenging coefficient, respectively. The scavenging coefficient Λ increases with precipitation rate according to

$$\Lambda = A I^B \ , \tag{39}$$

where I is the precipitation rate in mm/hour, A [s^{-1}] is the scavenging coefficient at I=1 mm/hour and B gives the dependency on precipitation rate. Both A and B must be specified in file `SPECIES`. FLEXPART uses the same scavenging coefficients for snow and rain.

As wet deposition depends nonlinearly on precipitation rate, subgrid variability of precipitation must be accounted for (Hertel et al., 1995). The area fraction which experiences precipitation given a certain grid-scale precipitation rate is calculated by

$$F = \max\left[0.05, CC \frac{I_l fr_l(I_l) + I_c fr_c(I_c)}{I_l + I_c}\right] \ , \tag{40}$$

where CC is the total cloud cover, I_l and I_c are the large scale and convective precipitation rates, respectively, and fr_l and fr_c are correction factors that depend on I_l and I_c (see Table 2). The subgrid scale precipitation rate is then I_s=$(I_l$+$I_c)/F$.

7.3 Dry deposition

Dry deposition is described in FLEXPART by a deposition velocity

$$v_d(z) = -F_C/C(z) \ , \tag{41}$$

where F_C and C are the flux and the concentration of a species at height z within the constant flux layer. A constant deposition velocity v_d can be set (file `SPECIES`). Alternatively, if the physical and chemical properties of a substance are known (file `SPECIES`), more complex parameterizations for gases and particles are also available.

Table 3. List of the landuse classes and roughness lengths used by FLEXPART. "Charnock" indicates that Charnock's relationship is used to calculate the roughness length.

Grassland	0.10
Arable land	0.15
Permanent crops	0.30
Forest	0.60
Inland water	Charnock
Urban areas	0.70
Other	0.10
Ocean	Charnock

7.3.1 Dry deposition of gases

The deposition velocity of a gas is calculated with the *resistance method* (Wesely and Hicks, 1977) in subroutine `getvdep.f` according to

$$|v_d(z)| = [r_a(z) + r_b + r_c]^{-1} \ , \tag{42}$$

where r_a is the aerodynamic resistance between z and the surface, r_b is the quasilaminar sublayer resistance, and r_c is the bulk surface resistance.

The aerodynamic resistance r_a is calculated in function `raerod.f` using the flux-profile relationship based on Monin-Obukhov similarity theory (Stull, 1988)

$$r_a(z) = \frac{1}{\kappa u_*}[\ln(z/z_0) - \Psi_h(z/L) + \Psi_h(z_0/L)] \ . \tag{43}$$

Following Erisman et al. (1994), the quasilaminar sublayer resistance is

$$r_b = \frac{2}{\kappa u_*}\left(\frac{Sc}{Pr}\right)^{2/3} \ , \tag{44}$$

where Sc and Pr are the Schmidt and Prandtl numbers, respectively. Pr is 0.72 and Sc=υ/D_i, with υ being the kinematic viscosity of air and D_i being the molecular diffusivity of species i in air. The slight dependency of υ on air temperature is formulated in accordance with Pruppacher and Klett (1978). r_b is calculated in function `getrb.f`.

The surface resistance is calculated in function `getrc.f` following Wesely (1989) as

$$r_c = \left[1/(r_s + r_m) + 1/r_{lu} + 1/(r_{dc} + r_{cl}) + 1/(r_{ac} + r_{gs})\right]^{-1} \ , \tag{45}$$

where r_s, r_m and r_{lu} represent the bulk values for leaf stomatal, leaf mesophyll and leaf cuticle surface resistances (alltogether the upper canopy resistance) , r_{dc} represents the gas-phase transfer affected by buoyant convection in canopies, r_{cl} the resistance of leaves, twig, bark and other exposed surfaces in the lower canopy, r_{ac} the resistance for transfer that depends only on canopy height and density, and r_{gs} the resistance for the soil, leaf litter, etc., at the ground.

Each of these resistances is parameterized according to the species' chemical reactivity and solubility, the landuse type, and the meteorological conditions. The landuse inventory (Velde et al., 1994) provides the area fractions of eight landuse classes for which roughness lengths z_0 are estimated, on a grid with 10' resolution (Table 3). Charnock's relationship (Stull, 1988) $z_0{=}0.016u_*^2/g$ is used to calculate z_0 for the classes "Ocean" and "Inland water", because of its dependence on wave height. Deposition velocities are calculated for all landuse classes and weighted with their respective areas. Outside of Europe, the landuse classes are determined only from the ECMWF land-sea-mask, attributing the landuse classes "Ocean" to the sea surfaces and "Grasslands" to the land surfaces.

7.3.2 Dry deposition of particulate matter

The deposition of particulates is calculated in subroutine `partdep.f` according to

$$v_d(z) = \left[r_a(z) + r_b + r_a(z) r_b v_g\right]^{-1} + v_g \ , \tag{46}$$

where v_g is the gravitational settling velocity calculated from (Slinn, 1982)

$$v_g = \frac{g\rho_p d_p^2 C_{\mathrm{cun}}}{18\mu} \ , \tag{47}$$

where ρ_p and d_p are the particle density and diameter, μ the dynamic viscosity of air ($0.000018\,\mathrm{kg\,m^{-1}\,s^{-1}}$) and C_{cun} the Cunningham slip-flow correction. The quasilaminar sublayer resistance is calculated from the same relationship as for gases, with an additional impaction term. For further details see Slinn (1982).

Settling and dry deposition velocities are strongly dependent on particulate size. FLEXPART assumes a logarithmic normal size distribution of the particulate mass. The user must specify the mean particulate diameter $\overline{d_p}$ and a measure of the variation around $\overline{d_p}$, σ_p. Then, the settling and deposition velocities are calculated for several particle diameters and are weighted with their respective particulate mass fractions.

Gravitational settling is important not only for the computation of the dry deposition velocity, but also affects the particle's trajectory. As a FLEXPART particle can normally represent several species, gravitational settling can only be taken into account correctly (i.e., influence particle trajectories) in single-species simulations.

7.3.3 Loss of particle mass due to dry deposition

The depositon velocity is calculated for a reference height (parameter `href` in file `includepar`) of 15 m. For all particles below $2h_{\mathrm{ref}}$, the mass lost by deposition is calculated by

$$\Delta m(t) = m(t)\left[1 - \exp\left(\frac{-v_d(h_{\mathrm{ref}})\Delta t}{2h_{\mathrm{ref}}}\right)\right] \ . \tag{48}$$

8 Calculation of concentrations, uncertainties, age spectra, and mass fluxes

Output quantities C_{T_c} at time T_c (output interval `loutstep` is set in file `COMMAND`) are calculated as time-averages over period $[T_c{-}\Delta T_c/2, T_C{+}\Delta T_c/2]$. ΔT_c must be specified (`loutaver`) in file `COMMAND`. To calculate the time-averages, concentrations C_{T_s} at times T_s within $[T_c{-}\Delta T_c/2, T_C{+}\Delta T_c/2]$ are sampled at shorter intervals ΔT_s (`loutsample` in file `COMMAND`) and are then divided by the number $N{=}\frac{\Delta T_c}{\Delta T_s}$ of samples taken:

$$C_{T_c} = \frac{1}{N}\sum_{i=1}^{N} C_{T_s} \ . \tag{49}$$

Both ΔT_c and ΔT_s must be multiples of the FLEXPART synchronisation interval (`lsynctime` in file `COMMAND`). The shorter the sampling interval ΔT_s, the more samples are taken and the more accurate are thus the time-averaged concentrations.

8.1 Concentrations, mixing ratios, and emission response functions

The user can choose (`iout` in file `COMMAND`, which must be set to 1 for backward runs) whether concentrations, volume mixing ratios or both shall be produced. We shall use the term "concentration" and particle mass here, but note that the actual units are determined by the settings of `ind_source` and `ind_receptor`, according to Table 1. The concentration in a grid cell is calculated in subroutine `conccalc.f` by sampling the tracer mass fractions of all particles within the grid cell and dividing by the grid cell volume

$$C_{T_s} = \frac{1}{V}\sum_{i=1}^{N}(m_i f_i) \ , \tag{50}$$

with V being the grid cell volume, m_i particle mass, N the total number of particles, and f_i the fraction of the mass of particle i attributed to the respective grid cell. This mass fraction is calculated by a uniform kernel with bandwidths $(\Delta x, \Delta y)$, where Δx and Δy are the grid distances on the longitude-latitude output grid. Figure 1 illustrates this: The particle is located at the center of the shaded rectangle with side lengths $(\Delta x, \Delta y)$. Generally, the shaded area stretches over four grid cells, each of which receives a fraction of the particle's mass equal to the fraction of the shaded area falling within this cell. The uniform kernel is not used during the first 3 hours after a particle's release (when the mass is attributed only to the grid cell it resides in), in order to avoid smoothing close to the source.

Wet and dry deposition fields are calculated on the same output grid (subroutines `wetdepokernel.f` and `drydepokernel.f`) and are written to all output grid files. The deposited matter is accumulated over the course of a model run, i.e. it generally increases with model time.

However, radioactive decay is calculated also for the deposited matter.

8.2 Uncertainties

The uncertainty of the output is estimated by carrying `nclassunc` classes of particles in the model simulation, and determining the concentration separately for each class (subroutine `conccalc.f`). The standard deviation, calculated from `nclassunc` concentration estimates and divided by $\sqrt{\texttt{nclassunc}}$, is the standard deviation of the mean concentration (subroutine `concoutput.f`), which is also written to the output files for every grid cell. Note that the memory needed for some auxiliary fields increases with `nclassunc` t and the number of age classes (see below). It may, thus, be necessary to reduce `nclassunc` for runs with large output grids and age spectra calculations or in the backward mode.

8.3 Age spectra

The age spectra option is switched on using `lagespectra` in file `COMMAND`, with the age classes specified in seconds in file `AGECLASSES`. Concentrations are split into contributions from particles of different age, defined as the time passed since their release. Particles are terminated once they are older than the oldest age class and their storage space is made available to new particles. Therefore, the age spectra option can be used also with a single age class for defining a maximum particle age.

8.4 Parabolic kernel

In addition to the simple uniform kernel method, a computationally demanding parabolic kernel as described in (Uliasz, 1994) can be used to calculate surface concentrations for a limited number of receptor points (age spectra are not available in this case):

$$C_{T_s}(x, y, z = 0) = \sum_{i=1}^{N} \left[\frac{2m_i K(r_x, r_y, r_z)}{h_{x_i} h_{y_i} h_{z_i}} \right] , \qquad (51)$$

where h_{x_i}, h_{y_i} and h_{z_i} are the kernel bandwidths which determine the degree of smoothing, $r_x = (X_i - x)/h_{x_i}$, $r_y = (Y_i - y)/h_{y_i}$, $r_z = Z_i/h_{z_i}$ with X_i, Y_i and Z_i being the position of particle i. The kernel bandwidths are a function of the particles' age.

8.5 Mass fluxes

Mass flux calculations can be switched on using `iflux` in file `COMMAND`. Mass fluxes are calculated separately for eastward, westward, northward, southward, upward and downward directions and contain both grid-scale and subgrid-scale motions. Mass fluxes are determined for the centerlines of the output grid cells, e.g. vertical fluxes are calculated for motions across the half level of each output cell.

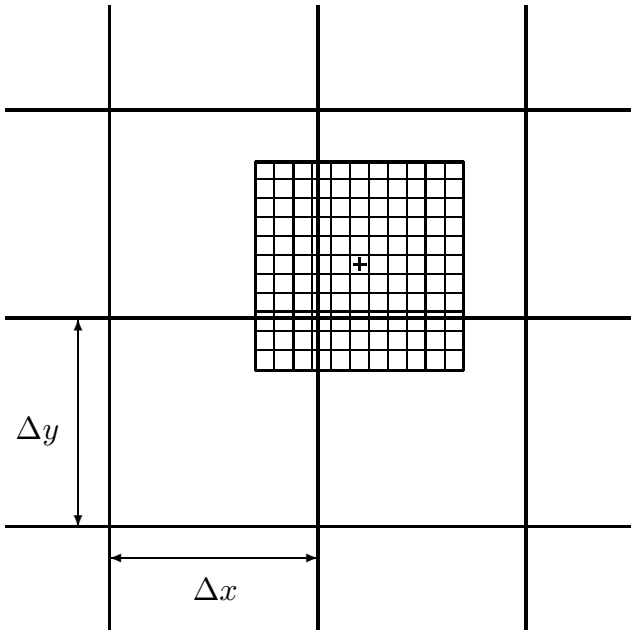

Fig. 1. Illustration of the uniform kernel used to calculate gridded concentration and deposition fields. The particle position is marked by "+".

9 Domain-filling option

9.1 General

If `mdomainfill=1` in file `COMMAND` particles are not released at specific locations. Instead, the longitudes and latitudes specified for the first release in the `RELEASES` file are used to set up a global or limited model domain. The particles (number is also taken from `RELEASES`) are then distributed in the model domain proportionally to air density (subroutine `init_domainfill.f`). Each particle receives the same mass, altogether accounting for the total atmospheric mass. Subsequently, particles move freely in the atmosphere.

If a limited domain is chosen, mass fluxes are determined in small grid boxes at the boundary of this domain (boundaries must be at least one grid box away from the boundaries of the meteorological input data). In the grid cells with air flowing into the model domain, mass fluxes are accumulated over time and whenever the accumulated mass exceeds the mass of a particle, a new particle (or more, if required) is released at a randomly chosen position at the boundary of the box (subroutine `boundcond_domainfill.f`). At the outflowing boundaries particles are terminated. Note that, due to the change of mass of the atmosphere in the model domain and due to numerical effects, the number of particles used is not exactly constant throughout the simulation.

9.2 Stratospheric ozone tracer

If `mdomainfill=2`, the domain-filling option is used to simulate a stratospheric ozone tracer. Upon particle creation, the potential vorticity (PV) at its position is determined by interpolation from the ECMWF data. Particles initially located in the troposphere (PV<`pvcrit` potential vorticity units (pvu), default 2 pvu) are not used. In contrast, stratospheric particles (PV>`pvcrit`) are given a mass according to:

$$M_{O_3} = M_{air}\, P\, C\, 48/29 \tag{52}$$

where M_{air} is the mass of air a particle represents, P is PV in pvu, C=60×10^{-9} pvu^{-1} is the ozone/PV relationship (Stohl et al., 2000) (parameter `ozonescale`), and the factor 48/29 converts from volume to mass mixing ratio. Particles are then allowed to advect through the stratosphere and into the troposphere according to the winds.

10 Model output

Tracer concentrations and/or mixing ratios (for forward runs), or emission sensitivity response functions (for backward runs) are calculated on a three-dimensional longitude-latitude grid, defined in file `OUTGRID`, whose domain and resolution can differ from the grid on which meteorological input data are given. Two-dimensional wet and dry deposition fields are calculated over the same spatial domain, and tracer mass fluxes can also be determined on the 3-d grid. Except for the mass fluxes, output can also be produced on one nested output grid with higher horizontal but the same vertical resolution, defined in file `OUTGRID_NEST`. For certain locations, specified in file `RECEPTORS`, concentrations can also be calculated independently from a grid (see below). The time interval (variable `loutstep`) at which output is produced is read in from file `COMMAND`. For every output time, files are created, whose file name ends with the date and time in the format `yyyymmddhhmmss`. A list of all these output times is written to the formatted file `dates`. The dates indicate the ending time of an output sampling interval (see Sect. 8).

10.1 Gridded output

There are several output options in FLEXPART, which can all be selected in file `COMMAND`. Gridded output fields can be concentrations (files `grid_conc_date`), volume mixing ratios (files `grid_pptv_date`), emission response sensitivity in backward simulations (files `grid_time_date`), or fluxes (files `grid_flux_date`, unit 10^{-12} kg m^{-2} s^{-1} for forward runs). Files `grid_conc_date` are created only in forward runs, whereas files `grid_time_date` are only created in backward runs. Note that the units of the files `grid_conc_date` and `grid_time_date` depend on the settings of the switches `ind_source` and `ind_receptor`, following Table 1. In particular, the units of `grid_conc_date` can also be mass mixing ratios. For forward runs, additional files `grid_pptv_date` can be created, which contain volume mixing ratios for gases. Output files `grid_conc*`, `grid_pptv*`, and `grid_time*` also contain wet and dry deposition fields (unit 10^{-12} kg m^{-2} in forward mode), and all files contain, for each grid cell, corresponding uncertainties. All these file types share a common header, file `header` produced by subroutine `writeheader.f`, where important information on the model run (start of simulation, grid domain, number and position of vertical levels, age classes, release points, etc.) is stored. In all postprocessing programs, the header must be read in before the actual data files. File names for the output nests follow the same nomenclature as described above, but with `_nest` added (e.g., `header_nest`, or `grid_conc_nest_date`). The output files are written with subroutines `concoutput.f` and `fluxoutput.f`.

FLEXPART output files, except for `dates`, are all binary and often contain many grid cells with zero concentrations (or mixing ratios, fluxes, etc.). Writing out only those cells with non-zero values can produce smaller output than a full grid dump. But in this case the grid indices (note that all three are combined into a single integer number) must also be written out and this produces bigger output than a full grid dump if most grid cells contain non-zero concentrations. Therefore, at every output time and for every output field the more efficient method is determined and used.

10.2 Receptor point output

For a list of points at the surface, concentrations or mixing ratios in forward simulations can be determined with a grid-independent method. This information is written to files `receptor_conc` and `receptor_pptv`, respectively, for all dates of a simulation.

10.3 Particle dump and warm start option

Particle information (3-d position, release time, release point, and release masses for all species) can be written out to files (subroutine `partoutput.f`) either continuously (binary files `partposit_date`), or 'only at the end' of a simulation (file `partposit_end`). In both cases output is written every output interval but file `partposit_end` is overwritten upon each new output. If FLEXPART must be terminated, it can be continued later on by reading in files `header` and `partposit_end` produced by the previous run (subroutine `readpartpositions.f`). Such a warm start is done if variable `ipin` is set to 1 in file `COMMAND`.

If option `mquasilag` is chosen in file `COMMAND`, particle dumps every output interval are produced in a very compact format by converting the positions to an `integer*2` format (subroutine `partoutput_short.f`). As some accuracy is lost in the conversion, this output is not used for the

warm start option. Another difference to the normal particle dump is that every particle gets a unique number, thus allowing postprocessing routines to identify continuous particle trajectories.

10.4 Clustered plume trajectories

Condensed particle output using the clustering algorithm described in Sect. 6 is written to the formatted file `trajectories.txt`. Information on the release points (coordinates, release start and end, number of particles) is written by subroutine `openouttraj.f` to the beginning of file `trajectories.txt`. Subsequently, `plumetraj.f` writes out a time sequence of the clustering results for each release point: release point number, time in seconds elapsed since the middle of the release interval, plume centroid position coordinates, various overall statistics (e.g., fraction of particles residing in the ABL and troposphere), and then for each cluster the cluster centroid position, the fraction of particles belonging to the cluster, and the root-mean-square distance of cluster member particles from the cluster centroid.

11 Final remark

In this note, we have described the particle dispersion model FLEXPART version 6.2 with the dual purpose of creating a citeable reference for FLEXPART and providing an actual user manual. In an appendix provided as an electronic supplement (http://www.atmos-chem-phys.org/acp/5/2461/acp-5-2461-sp.pdf), the various FLEXPART input files are briefly explained and examples are given. As FLEXPART develops this text will be kept actual and will be accessible from the internet site http://zardoz.nilu.no/~andreas/flextra+flexpart.html.

Edited by: B. Kärcher

References

Asman, W. A. H.: Parameterization of below-cloud scavenging of highly soluble gases under convective conditions, Atmos. Environ., 29, 1359–1368, 1995.

Baumann, K. and Stohl, A.: Validation of a long-range trajectory model using gas balloon tracks from the Gordon Bennett Cup 95, J. Appl. Meteor., 36, 711–720, 1997.

Beljaars, A. C. M. and Holtslag, A. A. M.: Flux parameterization over land surfaces for atmospheric models, J. Appl. Meteor., 30, 327–341, 1991.

Berkowicz, R. and Prahm, L. P.: Evaluation of the profile method for estimation of surface fluxes of momentum and heat, Atmos. Environ., 16, 2809–2819, 1982.

Businger, J. A., Wyngaard, J. C., Izumi, Y., and Bradley, E. F.: Flux-profile relationships in the atmospheric surface layer, J. Atmos. Sci., 28, 181–189, 1971.

Dorling, S. R., Davies, T. D., and Pierce, C. E.: Cluster analysis: a technique for estimating the synoptic meteorological controls on air and precipitation chemistry – method and applications, Atmos. Environ., 26A, 2575–2581, 1992.

ECMWF: User Guide to ECMWF Products 2.1, Meteorological Bulletin M3.2, Reading, UK, 1995.

Emanuel, K. A.: A scheme for representing cumulus convection in large-scale models, J. Atmos. Sci., 48, 2313–2335, 1991.

Emanuel, K. A., and Živković-Rothman, M.: Development and evaluation of a convection scheme for use in climate models, J. Atmos. Sci., 56, 1766–1782, 1999.

Erisman, J. W., Van Pul, A., and Wyers, P.: Parametrization of surface resistance for the quantification of atmospheric deposition of acidifying pollutants and ozone, Atmos. Environ., 28, 2595–2607, 1994.

Flesch, T. K., Wilson, J. D., and Lee, E.: Backward-time Lagrangian stochastic dispersion models and their application to estimate gaseous emissions, J. Appl. Meteorol., 34, 1320–1333, 1995.

Forster, C., Wandinger, U., Wotawa, G., James, P., Mattis, I., Althausen, D., Simmonds, P., O'Doherty, S., Kleefeld, C., Jennings, S. G., Schneider, J., Trickl, T., Kreipl, S., Jäger, H., and Stohl, A.: Transport of boreal forest fire emissions from Canada to Europe, J. Geophys. Res., 106, 22 887–22 906, 2001.

Gupta, S., McNider, R. T., Trainer, M., Zamora, R. J., Knupp, K., and Singh, M. P.: Nocturnal wind structure and plume growth rates due to inertial oscillations, J. Appl. Meteor., 36, 1050–1063, 1997.

Hanna, S. R.: Applications in air pollution modeling, in: Atmospheric Turbulence and Air Pollution Modelling, edited by: Nieuwstadt, F. T. M. and van Dop, H., D. Reidel Publishing Company, Dordrecht, Holland, 1982.

Hertel, O., Christensen, J., Runge, E. H., Asman, W. A. H., Berkowicz, R., Hovmand, M. F., and Hov, O.: Development and testing of a new variable scale air pollution model – ACDEP, Atmos. Environ., 29, 1267–1290, 1995.

Hubbe, J. M., Doran, J. C., Liljegren, J. C., and Shaw, W. J.: Observations of spatial variations of boundary layer structure over the southern Great Plains cloud and radiation testbed, J. Appl. Meteor., 36, 1221–1231, 1997.

Kahl, J. D. W.: On the prediction of trajectory model error, Atmos. Environ., 30, 2945–2957, 1996.

Legg, B. J. and Raupach, M. R.: Markov-chain simulation of particle dispersion in inhomogeneous flows: the mean drift velocity induced by a gradient in Eulerian velocity variance, Bound.-Layer Met., 24, 3–13, 1982.

Legras, B., Joseph, B., and Lefevre, F.: Vertical diffusivity in the lower stratosphere from Lagrangian back-trajectory reconstructions of ozone profiles, J. Geophys. Res., 108, 4562, doi:10.1029/2002JD003045, 2003.

Maryon, R. H.: Determining cross-wind variance for low frequency wind meander, Atmos. Environ., 32, 115–121, 1998.

McMahon, T. A. and Denison, P. J.: Empirical atmospheric deposition parameters – a survey, Atmos. Environ., 13, 571–585, 1979.

McNider, R. T., Moran, M. D., and Pielke, R. A.: Influence of diurnal and inertial boundary layer oscillations on long-range dispersion, Atmos. Environ., 22, 2445–2462, 1988.

Petterssen, S.: Weather Analysis and Forecasting, McGraw-Hill Book Company, New York., pp. 221–223, 1940.

Pruppacher, H. R. and Klett, J. D.: Microphysics of clouds and precipitation, D. Reidel Publishing Company, Dordrecht, 714p., 1978.

Rodean, H.: Stochastic Lagrangian models of turbulent diffusion, Meteorological Monographs, 26 (48), American Meteorological Society, Boston, USA, 1996.

Ryall, D. B. and Maryon, R. H.: Validation of the UK Met Office's NAME model against the ETEX dataset, in: ETEX Symposium on Long-Range Atmospheric Transport, Model Verification and Emergency Response, edited by: Nodop, K., European Commission, EUR 17 346, 151–154, 1997.

Seibert, P.: Inverse modelling with a Lagrangian particle dispersion model: application to point releases over limited time intervals, in: Air Pollution Modeling and its Application XIV, edited by: Gryning, S. E. and Schiermeier, F. A., Proc. of ITM Boulder, New York, Plenum Press, 381–389, 2001.

Seibert, P. and Frank, A.: Source-receptor matrix calculation with a Lagrangian particle dispersion model in backward mode, Atmos. Chem. Phys., 4, 51–63, 2004,
SRef-ID: 1680-7324/acp/2004-4-51.

Seibert, P., Krüger, B., and Frank, A.: Parametrisation of convective mixing in a Lagrangian particle dispersion model, Proceedings of the 5th GLOREAM Workshop, Wengen, Switzerland, 24–26 September, 2001.

Slinn, W. G. N.: Predictions for particle deposition to vegetative canopies, Atmos. Environ., 16, 1785–1794, 1982.

Spichtinger, N., Wenig, M., James, P., Wagner, T., Platt, U., and Stohl, A.: Satellite detection of a continental-scale plume of nitrogen oxides from boreal forest fires, Geophys. Res. Lett., 28, 4579–4582, 2001.

Stohl, A.: Computation, accuracy and applications of trajectories – a review and bibliography, Atmos. Environ., 32, 947–966, 1998.

Stohl, A., Cooper, O. R., Damoah, R., Fehsenfeld, F. C., Forster, C., Hsie, E.-Y., Hübler, G., Parrish, D. D., and Trainer, M.: Forecasting for a Lagrangian aircraft campaign, Atmos. Chem. Phys., 4, 1113–1124, 2004,
SRef-ID: 1680-7324/acp/2004-4-1113.

Stohl, A., Eckhardt, S., Forster, C., James, P., Spichtinger, N., and Seibert, P.: A replacement for simple back trajectory calculations in the interpretation of atmospheric trace substance measurements, Atmos. Environ., 36, 4635–4648, 2002.

Stohl, A., Forster, C., Eckhardt, S., Spichtinger, N., Huntrieser, H., Heland, J., Schlager, H., Wilhelm, S., Arnold, F., and Cooper, O.: A backward modeling study of intercontinental pollution transport using aircraft measurements, J. Geophys. Res., 108, 4370, doi:10.1029/2002JD002862, 2003.

Stohl, A., Hittenberger, M., and Wotawa, G.: Validation of the Lagrangian particle dispersion model FLEXPART against large scale tracer experiment data, Atmos. Environ., 32, 4245–4264, 1998.

Stohl, A., Spichtinger-Rakowsky, N., Bonasoni, P., Feldmann, H., Memmesheimer, M., Scheel, H. E., Trickl, T., Hübener, S. H., Ringer, W., and Mandl, M.: The influence of stratospheric intrusions on alpine ozone concentrations, Atmos. Environ., 34, 1323–1354, 2000.

Stohl, A. and Thomson, D. J.: A density correction for Lagrangian particle dispersion models, Bound.-Layer Met., 90, 155–167, 1999.

Stohl, A. and Trickl, T.: A textbook example of long-range transport: Simultaneous observation of ozone maxima of stratospheric and North American origin in the free troposphere over Europe, J. Geophys. Res., 104, 30 445–30 462, 1999.

Stohl, A., Wotawa, G., Seibert, P., and Kromp-Kolb, H.: Interpolation errors in wind fields as a function of spatial and temporal resolution and their impact on different types of kinematic trajectories, J. Appl. Meteor., 34, 2149–2165, 1995.

Stull, R. B.: An Introduction to Boundary Layer Meteorology, Kluwer Academic Publishers, Dordrecht, 1988.

Telford, J. W.: Turbulence, entrainment and mixing in cloud dynamics, Pure Appl. Geophys., 113, 1067–1084, 1975.

Thomson D. J.: Criteria for the selection of stochastic models of particle trajectories in turbulent flows, J. Fluid Mech., 180, 529–556, 1987.

Uliasz, M.: Lagrangian particle dispersion modeling in mesoscale applications, in: Environmental Modeling, Vol. II, edited by: Zannetti, P., Computational Mechanics Publications, Southampton, UK, 1994.

Velde van de, R. J., Faber, W. S., Katwijk van, V. F., Kuylenstierna, J. C. I., Scholten., H. J., Thewessen, T. J. M., Verspuij, M., and Zevenbergen, M.: The Preparation of a European Land Use Database, National Institute of Public Health and Environmental Protection, Report No. 712401001, Bilthoven, The Netherlands, 1994.

Vogelezang, D. H. P. and Holtslag, A. A. M.: Evaluation and model impacts of alternative boundary-layer height formulations, Bound.-Layer Met., 81, 245–269, 1996.

Wesely, M. L. and Hicks, B. B.: Some factors that affect the deposition rates of sulfur dioxide and similar gases on vegetation, J. Air Poll. Contr. Assoc., 27, 1110–1116, 1977.

Wesely, M. L.: Parameterization of surface resistances to gaseous dry deposition in regional-scale numerical models, Atmos. Environ., 23, 1293–1304, 1989.

Wilson, D. J. and Flesch, T. K.: Flow boundaries in random-flight dispersion models: enforcing the well-mixed condition, J. Appl. Meteor., 32, 1695–1707, 1993.

Wilson, J. D., Legg, B. J., and Thomson, D. J.: Calculation of particle trajectories in the presence of a gradient in turbulent-velocity scale, Bound.-Layer Met., 27, 163–169, 1983.

Wilson, J. D. and Sawford, B. L.: Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere, Bound.-Layer Met., 78, 191–210, 1996.

Wotawa, G., Stohl, A., and Kromp-Kolb, H.: Parameterization of the planetary boundary layer over Europe – a data comparison between the observation based OML preprocessor and ECMWF model data, Contr. Atmos. Phys., 69, 273–284, 1996.

Wotawa, G. and Stohl, A.: Boundary layer heights and surface fluxes of momentum and heat derived from ECMWF data for use in pollutant dispersion models – problems with data accuracy, in: The Determination of the Mixing Height – Current Progress and Problems, edited by: Gryning, S.-E., Beyrich, F., and E. Batchvarova, EURASAP Workshop Proceedings, 1–3 October 1997, Risø National Laboratory, Denmark, 1997.

Zannetti, P.: Particle Modeling and Its Application for Simulating Air Pollution Phenomena, in: Environmental Modelling, edited by: Melli P. and P. Zannetti, Computational Mechanics Publications, Southampton, UK, 1992.