

# **The Lagrangian particle dispersion model FLEXPART version 8.2**

A. Stohl<sup>1</sup>, H. Sodemann<sup>1</sup>, S. Eckhardt<sup>1</sup>, A. Frank<sup>2</sup>, P. Seibert<sup>2</sup>, and G. Wotawa<sup>3</sup>

<sup>1</sup>Norwegian Institute of Air Research, Kjeller, Norway

<sup>2</sup>Institute of Meteorology, University of Natural Resources and Applied Life Sciences, Vienna,  
Austria

<sup>3</sup>Preparatory Commission for the Comprehensive Nuclear Test Ban Treaty Organization,  
Vienna, Austria

# The Lagrangian particle dispersion model FLEXPART version 8.2

A. Stohl<sup>1</sup>, H. Sodemann<sup>1</sup>, S. Eckhardt<sup>1</sup>, A. Frank<sup>2</sup>, P. Seibert<sup>2</sup>, and G. Wotawa<sup>3</sup>

<sup>1</sup>Norwegian Institute of Air Research, Kjeller, Norway

<sup>2</sup>Institute of Meteorology, University of Natural Resources and Applied Life Sciences, Vienna, Austria

<sup>3</sup>Preparatory Commission for the Comprehensive Nuclear Test Ban Treaty Organization, Vienna, Austria

**Abstract.** The Lagrangian particle dispersion model FLEXPART was originally (in its first release in 1998) 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 35 groups from 14 different countries and is seeing both operational and research applications. The last citable manuscript for FLEXPART is: (Stohl et al., 2005)

- The code was updated to work with the ECMWF grib\_api V1.6.1 and above (tested up to 1.9.5)
- An important bug the concentration output routines was fixed. A numerical problem lead in some circumstances to garbled data. The header version identifier has been changed to version 8.2. New routines for loading data available for download on the FLEXPART homepage.
- Several bugs in the wet deposition parameterization were fixed.
- A bug which lead to ignoring landuses that was introduced in version 8.0 has been fixed.
- Several bugs concerning nested grids have been fixed.

---

## 1 Updates since FLEXPART version 8.0

In this version 8.2 of FLEXPART the representation of physical processes was improved as well as a number of technical changes and bugfixes implemented. In addition, the program is now released under a GNU GPL license. For the first time, detailed installation instructions are provided to help users getting started with running FLEXPART. A short section on the new Python routines pflexpart for reading FLEXPART output data has been included as well.

### Technical Changes:

- grib2 compatibility for ECMWF data which will be introduced soon (new data retrieval routines available)
- AVAILABLE files can now contain up to 256 characters, and include the path directly with the input file name. This is used to gather data files stored in different directories. The third line in the `pathnames` file should be left empty if long data file names are used.

### Algorithm Changes:

- A new, more detailed settling parameterisation for aerosols was implemented. The dynamic viscosity of air is now calculated as a function of temperature, and the calculation of settling velocities is now more accurate for higher Reynolds numbers.
- It is now possible to produce output files for backward runs that can be used to interface FLEXPART with model output from another model or from a FLEXPART forward run. A gridded output file containing the exact sensitivities to these initial conditions can produced. To this end, a new switch has been added in the COMMAND file.

## 2 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.

---

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

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). Version 6.2 saw corrections to the numerics in the convection scheme, the addition of a domain-filling option, and the possibility to use output nests. Version 6.4 runs with NCEP GFS model data. Version 7.0 was a transition version that was not publicly released. Version 8.0 was a major release. It unifies the ECMWF and GFS model versions in one source package. GFS data in GRIB2 format can be read, using ECMWF's `grib_api` library. Each species got its own definition file. Output can be written individually for multiple species in backward runs. The output format was changed to a compressed sparse matrix format. Memory is partly allocated dynamically. Furthermore, dry and wet deposition algorithms were updated and new, global landuse inventory introduced. OH reaction based on a monthly averaged 3 dimensional OH-

field is available as an option.

FLEXPART is coded following the Fortran 95 standard and tested with several compilers (gfortran, Absoft, Portland Group) under a number of operating systems (Linux, Solaris, Mac OS X, 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://transport.nilu.no/flexpart>. According to a recent user survey, at least 34 groups from 17 countries are currently using FLEXPART. The user community maintains discussion by a mailing list to which one can subscribe on the flexpart home page. 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). The standard source code distribution contains also the source files for a version of FLEXPART using the global National Centers of Environmental Prediction (NCEP) model data on pressure levels. Other users have developed FLEXPART versions using input data from a suite of different and meso-scale (e.g. MM5, WRF, COSMO) models, some of which are available from the FLEXPART website but are not described here.

### 3 License

FLEXPART has been free software ever since it first was released. The status as a free software is now more formally established by releasing the code under the GNU General Public License (GPL) Version 3. The text of the license is included in the file `COPYING` in the source code archive.

### 4 Installation

Getting FLEXPART up and running on their systems has been a major hurdle to many new users of the software. Since the inclusion of the `grib_api` library has further complicated the compilation of FLEXPART, we include a step-by-step installation instruction of FLEXPART. The steps below are described for an Ubuntu 10.4 LT UNIX release.

#### 4.1 Installing required libraries

In order to process input files in GRIB2 format, FLEXPART V8.2 needs to be compiled with ECMWF's `grib_api` library version 1.6.1 and above. Since the API of the `grib_api` library may change in the future, upward compatibility cannot be guaranteed. The input files in GRIB2 format can be compressed to save bandwidth and storage space. In order to make use of this feature, the `jasper` library needs to be installed on the system. Optionally, the `emos` library can be used to run FLEXPART with input files in GRIB1 format. The following steps should be executed in sequence:

#### 4.1.1 Install jasper library (Version 1.900.1)

Download the jasper library from the jasper project page<sup>1</sup>

```
unzip jasper-1.900.1.zip
cd jasper-1.900.1
./configure [--prefix=<installation path>]
make
make check
make install
```

#### 4.1.2 Install grib\_api library (Version 1.6.1 or later)

Download the grib\_api library from the ECMWF website<sup>2</sup>

```
tar -xvf grib_api-1.9.5.tar.gz
./configure [--with-jasper=<jasper path>]
make
make check
make install
```

#### 4.1.3 Optional: Install emos library (Version 000372)

Download the emos library from the ECMWF website<sup>3</sup>

```
tar -xvf emos_000372.tar.gz
./build_library
./install
```

### 4.2 Compiling FLEXPART V8.2

Download the FLEXPART source code archive from the FLEXPART homepage<sup>4</sup>

```
tar -xvf flexpart82.tar.gz
```

optionally edit the file `includepar` to set parameters for the data center, grid dimension, and particle number edit the `LIBRARY` path variable in the makefiles according to the position of `libgrib_api` and `libjasper`

```
make -f makefile.<center>_<compiler>_<system>
```

In the above statement, `center` can be one of: `gfs`, `ecmwf`, `ecmwf_emos`, `gfs_emos` compiler can be one of `absoft` or `gfortran` (`emos` library only with `absoft`) system can be one of 32, 64 (`emos` only 32). The system parameter must match that of the compiled libraries. See also Table 1 for all available makefiles.

When recompiling after making changes, all object and module files can be removed safely by using

```
make -f makefile.<xxx> clean
```

<sup>1</sup><http://www.ece.uvic.ca/mdadams/jasper/>

<sup>2</sup>[http://www.ecmwf.int/products/data/software/download/grib\\_api.html](http://www.ecmwf.int/products/data/software/download/grib_api.html)

<sup>3</sup><http://www.ecmwf.int/products/data/software/interpolation.html>

<sup>4</sup><http://transport.nilu.no/flexpart/flexpart/view>

## 5 Input data and grid definitions

FLEXPART is an off-line model that uses meteorological fields (analyses or forecasts) in Gridded Binary (GRIB) format in version 1 or 2 from the ECMWF numerical weather prediction model (ECMWF, 1995) on a latitude/longitude grid and on native ECMWF model levels as input. Optionally, GRIB data from NCEP's GFS model, available on pressure levels, can be used. The ECMWF 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 libraries is *not* provided with the FLEXPART source codes but is publicly available (see Sec. 4). 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.

### 5.1 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

**Table 1.** List of available makefiles

Makefile GFS	Makefile ECMWF	GRIB version
makefile.gfs_gfortran_32	makefile.ecmwf_gfortran_32	1/2
makefile.gfs_gfortran_64	makefile.ecmwf_gfortran_64	1/2
makefile.gfs_absoft_32	makefile.ecmwf_absoft_32	1/2
makefile.gfs_absoft_64	makefile.ecmwf_absoft_64	1/2
makefile.gfs_emos_absoft_32	makefile.ecmwf_emos_absoft_32	1

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` (`gridcheck_gfs.f` for the GFS version), 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°. The default setting for global ECMWF fields is `nxshift=359`, while GFS fields the default value is `nxshift=0`.

## 5.2 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.  $\eta$ ) levels which are defined by a hybrid coordinate system. The conversion from  $\eta$  to pressure coordinates is given by  $p_k = A_k + B_k p_s$  and the heights of the  $\eta$  surfaces are defined by  $\eta_k = A_k/p_0 + B_k$ , where  $\eta_k$  is the value of  $\eta$  at the  $k^{\text{th}}$  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 pre-processor. A surface level is defined in addition to the regular  $\eta$  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 sections 6 and 7) are calculated in units of  $\text{m s}^{-1}$ , and not in  $\eta$  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

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

where  $\dot{\eta} = \eta \partial p / \partial \eta$ . 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. The landuse inventory of Belward et al. (1999) is provided in an extra file in the options directory (`IGBP_intl.dat`).

Note that GFS data is provided on pressure levels (currently 26) which requires that the FLEXPART code is compiled with a makefile for the GFS model. GFS data are freely available from the NCEP data archives. An example for a retrieval of GFS data files via ftp is given below:

```
cd $DIR_GFS_05TEMPORARY
ftp ftpprd.ncep.noaa.gov << EOF11
binary
cd /pub/data/nccf/com/gfs/prod/gfs.2009111006
get gfs.t06z.pgrb2f00 GF09111006
quit
EOF11
```

## 5.3 Meteorological input data formats

While NCEP has already switched to the more flexible and compact GRIB2 format, the ECMWF is only gradually transitioning from GRIB1 to the new GRIB2 format. Transition to GRIB2 becomes necessary, at least for 3D fields, when the ECMWF increases the vertical resolution beyond 100 model levels in 2012 (which cannot be handled by the GRIB1 format). Currently, GRIB2 codes have been defined for all

3D-fields required by FLEXPART, while definitions are still missing for a part of the 2D-fields.

According to the ECMWF it may take another year to define the GRIB2 codes for all 2D-fields used by FLEXPART. Therefore, in version 8.2, it is now possible to read in GRIB files that contain only GRIB1 coded fields, files that contain only GRIB2 coded fields, and files that contain mixed GRIB1/2 fields. Such mixed GRIB1/2 files are produced by the current version 4 of the FLEXPART data retrieval routines (available from the FLEXPART homepage).

## 6 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

$$\tau = \sqrt{\tau_1^2 + \tau_2^2}, \quad (2)$$

where  $\tau_1$  and  $\tau_2$  are the surface stresses in east/west and north/south direction, respectively. Friction velocity is then calculated in subroutine `scalev.f` as

$$u_* = \sqrt{\tau/\rho}, \quad (3)$$

where  $\rho$  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:

$$u_* = \frac{\kappa \Delta u}{\ln \frac{z_l}{z_2} - \Psi_m(\frac{z_l}{L}) + \Psi_m(\frac{z_2}{L})}, \quad (4)$$

$$\Theta_* = \frac{\kappa \Delta \Theta}{0.74 \left[ \ln \frac{z_l}{z_2} - \Psi_h(\frac{z_l}{L}) + \Psi_h(\frac{z_2}{L}) \right]}, \quad (5)$$

$$L = \frac{\bar{T} u_*^2}{g \kappa \Theta_*}, \quad (6)$$

where  $\kappa$  is the von Kármán constant (0.4),  $z_l$  is the height of the second model level,  $\Delta u$  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,  $\Psi_m$  and  $\Psi_h$  are the stability correction functions for momentum and heat (Businger et al., 1971; Beljaars and Holtslag, 1991),  $g$  is the acceleration due to gravity,  $\Theta_*$  is the temperature scale and  $\bar{T}$  is the average surface layer temperature (taken as  $T$  at the first model level). The heat flux is then computed by

$$\overline{(w'\Theta'_v)_0} = -\rho c_p u_* \Theta_*, \quad (7)$$

where  $\rho c_p$  is the specific heat capacity of air at constant pressure.

ABL heights are calculated according to Vogelesang 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

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

exceeds the critical value of 0.25.  $\Theta_{v1}$  and  $\Theta_{vl}$  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 1<sup>st</sup> and  $l^{\text{th}}$  model level, respectively. The formulation of Eq. 8 can be improved for convective situations by replacing  $\Theta_{v1}$  with

$$\Theta'_{v1} = \Theta_{v1} + 8.5 \frac{\overline{(w'\Theta'_v)_0}}{w_* c_p}, \quad (9)$$

where

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

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 1200 and 1800 LT and the ABL heights at those times are, say, 1200 m and 200 m, and linear interpolation is used, the ABL height at 1700 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

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

Here,  $\sigma_Z$  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.

## 7 Particle transport and diffusion

### 7.1 Particle trajectory calculations

FLEXPART generally uses the simple “zero acceleration” scheme

$$\mathbf{X}(t + \Delta t) = \mathbf{X}(t) + \mathbf{v}(\mathbf{X}, t)\Delta t, \quad (12)$$

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

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

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

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  $ct1 > 0$  (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.

### 7.2 The Langevin equation

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

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

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

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

where  $w$  and  $\sigma_w$  are the turbulent vertical wind component and its standard deviation,  $\tau_{L_w}$  is the Lagrangian timescale for the vertical velocity autocorrelation and  $\rho$  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  $w/\sigma_w$  instead of  $w$  (Wilson et al., 1983):

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

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 section 7.3); 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  $(\Delta t/\tau_{L_w}) \geq 0.5$ ,

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

where  $r_w = \exp(-\Delta t/\tau_{L_w})$  is the autocorrelation of the vertical wind, and  $\zeta$  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  $\Delta t$ .

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

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

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).

### 7.3 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  $\tau_{L_i}$  (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  $\tau_L$ . Since the vertical wind is most important, only  $\tau_{L_w}$  is used for this. The user must specify two constants, `ctl` and `ifine` in file `COMMAND`. The first one determines the time step  $\Delta t_i$  according to

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

The minimum value of  $\Delta t_i$  is 1 second.  $\Delta t_i$  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  $\Delta t_w = \Delta t_i / \text{ifine}$  is used. However, note that there is no interaction between horizontal and vertical wind components on timescales less than  $\Delta t_i$ . 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.

### 7.4 Parameterization of the wind fluctuations

For  $\sigma_{v_i}$  and  $\tau_{L_i}$  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  $\sigma_w$ , as Hanna’s scheme does not always yield smooth profiles of  $\sigma_w$  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  $\tau_{L_u}$ ,  $\tau_{L_v}$  and  $\tau_{L_w}$  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:

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

$$\tau_{L_u} = \tau_{L_v} = 0.15 \frac{h}{\sigma_u} \quad (21)$$

$\sigma_w =$

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

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

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

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

$$\tau_{L_w} = 0.59 \frac{z}{\sigma_w} \quad (24)$$

For  $z/h > 0.1$ :

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



**Neutral conditions:**

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

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

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

**Stable conditions:**

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

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

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

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

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

Lacking suitable turbulence parameterizations above the ABL ( $z > h$ ), a constant vertical diffusivity  $D_z=0.1 \text{ m}^2\text{s}^{-1}$  is used in the stratosphere, following recent work of Legras et al. (2003), whereas a horizontal diffusivity  $D_h=50 \text{ m}^2\text{s}^{-1}$  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  $\sigma_{v_i} = \sqrt{D_i/dt}$ .

**7.5 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).

**7.6 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 humidity 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

$$T_{vp}^{LCL+1} \geq T_v^{LCL+1} + T_{thres} \quad (34)$$

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  $T_{thres} = 0.9 \text{ K}$  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:

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

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  $\theta^j$ , 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  $j$ :

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

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.

### 7.7 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  $\Delta t_s$  (file `COMMAND`). Particles are split into two (each of which receives half of the mass of the original particle) after travel times of  $\Delta t_s$ ,  $2\Delta t_s$ ,  $4\Delta t_s$ ,  $8\Delta t_s$ , and so on (subroutine `timemanager.f`).

## 8 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.

**Table 2.** 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 settings 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 <sup>-3</sup>
Forward	1	2	kg	ppt by mass
Forward	2	1	1	ng m <sup>-3</sup>
Forward	2	2	1	ppt by mass
Backward	1	1	1	s
Backward	1	2	1	s m <sup>3</sup> kg <sup>-1</sup>
Backward	2	1	1	s kg m <sup>-3</sup>
Backward	2	2	1	s

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 2 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<sup>-3</sup>) requires both switches to be set to 1.

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<sup>-3</sup>s<sup>-1</sup>), a concentration at the receptor (kg m<sup>-3</sup>) is obtained. If `ind_source=1` and `ind_receptor=2`, the response function has the unit s m<sup>3</sup> kg<sup>-1</sup> and if it is folded with the emission mass flux (again in kg m<sup>-3</sup>s<sup>-1</sup>), 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.

## 9 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`.

## 10 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.

### 10.1 Radioactive decay

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

$$m(t + \Delta t) = m(t) \exp(-\Delta t/\beta), \quad (37)$$

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_nnn`. Deposited pollutant mass decays at the same rate.

### 10.2 OH Reaction

If a positive value for the OH reaction rate is given in the file `SPECIES_nnn`, OH reaction is performed in the model simulation and tracer mass is lost by this reaction. A monthly averaged  $3^\circ \times 5^\circ$  resolution OH field averaged to 7 atmospheric levels is used. The fields have been obtained from the GEOS-CHEM model (Bey et al., 2001). The reaction rate is temperature corrected and an activation rate of  $1000 \text{ J mol}^{-1}$  is assumed.

### 10.3 Wet deposition

Since version 8.0, in-cloud and below-cloud scavenging are treated differently. Based on the humidity and temperature from the meteorological input data, the occurrence of clouds is calculated. If relative humidity exceeds 80% the occurrence of a cloud is assumed.

**In cloud scavenging** is treated differently for gases and particles. The implementation follows the scheme of Hertel et al. (1995).

In general, the scavenging coefficient  $\Lambda$  ( $\text{s}^{-1}$ ) depends on the precipitation rate  $I$  ( $\text{mm h}^{-1}$ ) and the height  $H_i$  over which scavenging takes place:

$$\Lambda = \frac{S_i I}{H_i} \quad (38)$$

$S_i$  is different for gases and particles. For particles,

$$S_i = 0.9/cl \quad (39)$$

where  $cl$  is the cloud liquid water content

$$cl = 2 \times 10^{-7} \cdot I^{0.36}. \quad (40)$$

For gases,

$$S_i = 1/cl_{\text{eff}}, \quad (41)$$

where  $cl_{\text{eff}}$  is an effective cloud liquid water content:

$$cl_{\text{eff}} = \frac{(1-cl)}{H_{\text{eff}}RT} + cl. \quad (42)$$

**Below cloud scavenging** takes the form of an exponential decay process (McMahon, 1979). With the scavenging coefficient  $\Lambda$ , wet deposition is described as

$$m(t + \Delta t) = m(t) \exp(-\Lambda \Delta t), \quad (43)$$

where  $m$  is the particle mass.  $\Lambda$  increases with the precipitation rate  $I$  according to

$$\Lambda = AI^B, \quad (44)$$

where  $A$  [ $\text{s}^{-1}$ ] 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_nnn`. 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

$$F = \max \left[ 0.05, CC \frac{I_l fr_l(I_l) + I_c fr_c(I_c)}{I_l + I_c} \right], \quad (45)$$

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 3). The subgrid scale precipitation rate is then  $I_s = (I_l + I_c)/F$ .

**Table 3.** Correction factors used for the calculation of the area fraction that experiences precipitation. Precipitation rates are in mm/hour.

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

#### 10.4 Dry deposition

Dry deposition is described in FLEXPART by a deposition velocity

$$v_d(z) = -F_C/C(z), \quad (46)$$

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_nnn`). Alternatively, if the physical and chemical properties of a substance are known (file `SPECIES_nnn`), more complex parameterizations for gases and particles are also available.

##### 10.4.1 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

$$|v_d(z)| = [r_a(z) + r_b + r_c]^{-1}, \quad (47)$$

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)

$$r_a(z) = \frac{1}{\kappa u_*} [\ln(z/z_0) - \Psi_h(z/L) + \Psi_h(z_0/L)]. \quad (48)$$

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

$$r_b = \frac{2}{\kappa u_*} \left( \frac{Sc}{Pr} \right)^{2/3}, \quad (49)$$

where  $Sc$  and  $Pr$  are the Schmidt and Prandtl numbers, respectively.  $Pr$  is 0.72 and  $Sc = v/D_i$ , with  $v$  being the kinematic viscosity of air and  $D_i$  being the molecular diffusivity of species  $i$  in air. The slight dependency of  $v$  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

$$\frac{1}{r_c} = \frac{1}{r_s + r_m} + \frac{1}{r_{lu}} + \frac{1}{r_{dc} + r_{cl}} + \frac{1}{r_{ac} + r_{gs}}, \quad (50)$$

where  $r_s$ ,  $r_m$  and  $r_{lu}$  represent the bulk values for leaf stomatal, leaf mesophyll and leaf cuticle surface resistances (altogether 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 IGBP landuse inventory (Belward et al., 1999) provides the area fractions of 13 landuse classes for which roughness lengths  $z_0$  are estimated, on a grid with 0.3 resolution (Table 4). 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. The original resolution of the IGBP land cover classification is 1 km x 1 km. To save storage space, this was regrided to a 0.3 x 0.3 degree resolution. Table 5 shows how the initial 17 classes were transferred in the 13 classes used in the deposition scheme of Wesely (1989). According to Helmig et al. (2007) a resistance of snow to ozone deposition of 10000 s m<sup>-1</sup> was used. For SO<sub>2</sub> a value of 100 according to Zhang et al. (2002) was used in the model. As the Wesely scheme was initially developed for North America, the rain forest was not represented well. Therefore a new category was introduced and resistance values according to Jacob and Wofsy (1990) were used. The resistance values are dependent on 5 different seasons. As on the southern hemisphere they are asynchronous to the northern hemisphere half a year was added when choosing the appropriate seasonal category. The snow depth is included in the original Wesely (1989) parameterization to some extent, as at a certain time of the year snow cover is assumed. In order to have more accurate information we use the snow cover based on the snow depth in the ECMWF fields. If the snow cover gets over 1 mm of water equivalent, the grid cell is considered as snow covered and category 12 (snow and ice) is applied.

##### 10.4.2 Dry deposition of particulate matter

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

$$v_d(z) = [r_a(z) + r_b + r_a(z)r_b v_g]^{-1} + v_g, \quad (51)$$

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

$$v_g = \frac{g \rho_p d_p^2 C_{cun}}{18\mu}, \quad (52)$$

where  $\rho_p$  and  $d_p$  are the particle density and diameter,  $\mu$  the dynamic viscosity of air (0.00018 kg m<sup>-1</sup>s<sup>-1</sup>) and  $C_{cun}$  the Cunningham slip-flow correction. The quasilaminar sublayer resistance is calculated from the same relationship as

**Table 4.** List of the landuse classes and roughness lengths used by FLEXPART. “Charnock” indicates that Charnock’s relationship is used to calculate the roughness length.

Urban land	0.7
Agricultural land	0.1
Range land	0.1
Deciduous forest	1.0
Coniferous forest	1.0
Mixed forest including wetland	0.7
Water, both salt and fresh	Charnock
Barren land mostly desert	0.01
Nonforested wetland	0.1
Mixed agricultural and range land	0.1
Rocky open areas with low growing shrubs	0.05
Snow and ice	0.001
Rainforest	1.0

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  $\bar{d}_p$  and a measure of the variation around  $\bar{d}_p$ ,  $\sigma_p$ . 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. Pay attention that gravitational settling is simply switched off in multi-species simulations, without warning.

With FLEXPART version 8.2, the temperature dependence of the dynamic viscosity is taken into account. Furthermore, we have extended the calculation of settling velocities to higher Reynolds numbers. For this, an iterative procedure has been introduced in subroutine `get_settling.f`, where the iteration is started with Stokes law settling (Naeslund and Thaning, 1991), and then Reynolds number and settling velocity are calculated until convergence is achieved.

#### 10.4.3 Loss of particle mass due to dry deposition

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

$$\Delta m(t) = m(t) \left[ 1 - \exp\left(\frac{-v_d(h_{ref})\Delta t}{2h_{ref}}\right) \right]. \quad (53)$$

## 11 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  $[T_c - \Delta T_c/2, T_c + \Delta T_c/2]$ .  $\Delta T_c$  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  $\Delta T_s$  (`loutsample` in file `COMMAND`) and are then divided by the number  $N = \frac{\Delta T_c}{\Delta T_s}$  of samples taken:

$$C_{T_c} = \frac{1}{N} \sum_{i=1}^N C_{T_s}. \quad (54)$$

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

### 11.1 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 2. 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

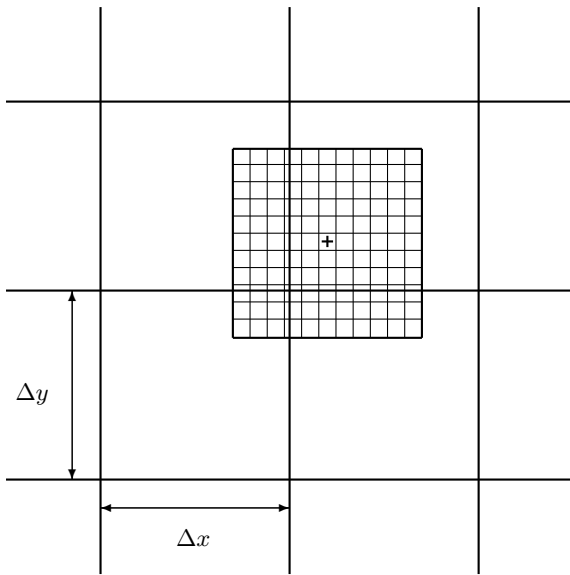
$$C_{T_s} = \frac{1}{V} \sum_{i=1}^N (m_i f_i), \quad (55)$$

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  $\Delta x$  and  $\Delta y$  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.

**Table 5.** Conversion from the IGBP land cover legend to the landuse categories used by Wesely

IGBP Land Cover Legend		Wesely	
Value	Description	Value	Description
1	Evergreen Needleleaf Forest	5	Coniferous forest
2	Evergreen Broadleaf Forest	13	Rainforest
3	Deciduous Needleleaf Forest	4	Deciduous forest
4	Deciduous Broadleaf Forest	4	Deciduous forest
5	Mixed Forest	6	Mixed forest including wetland
6	Closed Shrublands	11	rocky open areas with low growing shrubs
7	Open Shrublands	11	rocky open areas with low growing shrubs
8	Woody Savannas	11	rocky open areas with low growing shrubs
9	Savannas	11	rocky open areas with low growing shrubs
10	Grasslands	3	Range land
11	Permanent Wetlands	9	nonforested wetland
12	Croplands	2	Agricultural land
13	Urban and Built-Up	1	Urban land
14	Cropland/Natural Vegetation Mosaic	10	mixed agricultural and range land
15	Snow and Ice	12	snow and ice
16	Barren or Sparsely Vegetated	8	barren land mostly desert
17	Water Bodies	7	water, both salt and fresh

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

## 11.2 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  $\sqrt{\text{nclassunc}}$ , 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` 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.

## 11.3 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.

## 11.4 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):

$$C_{T_z}(x, y, z = 0) = \sum_{i=1}^N \left[ \frac{2m_i K(r_x, r_y, r_z)}{h_{x_i} h_{y_i} h_{z_i}} \right], \quad (56)$$

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.

## 11.5 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.

## 12 Domain-filling option

### 12.1 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.

### 12.2 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:

$$M_{O_3} = M_{air} P C 48/29 \quad (57)$$

where  $M_{air}$  is the mass of air a particle represents,  $P$  is PV in pvu,  $C = 60 \times 10^{-9} \text{ pvu}^{-1}$  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.

## 13 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 and for every species (`nnn`), files are created, with file names ending with date, time and species number in the format `yyyymmddhhmmss_nnn`. 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 section 11).

### 13.1 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_nnn`), volume mixing ratios (files `grid_pptv_date_nnn`), emission response sensitivity in backward simulations (files `grid_time_date_nnn`), or fluxes (files `grid_flux_date`, unit  $10^{-12} \text{ kg m}^{-2} \text{ s}^{-1}$  for forward runs), or, in backward mode, sensitivities to initial conditions taken from another model (`grid\_initial\_nnn`).

The species number identifier `nnn` starts at one (with leading zeros) and increases to the maximum number of species used in the simulation. Files `grid_conc_date_nnn` are created only in forward runs, whereas files `grid_time_date_nnn` are only created in backward runs. Note that the units of the files `grid_conc_date_nnn` and `grid_time_date_nnn` depend on the settings of the switches `ind_source` and `ind_receptor`, following Table 2. In particular, the units of `grid_conc_date` can also be mass mixing ratios. For forward runs, additional files `grid_pptv_date_nnn` 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  $10^{-12} \text{ kg m}^{-2}$  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_nnn`). The output files are written with subroutines `concoutput.f` and `fluxoutput.f`.

FLEXPART output typically contains many grid cells with zero values. It would be inefficient to write out all these zeroes. Therefore, a special format has been designed that compresses the information to the relevant information. In previous versions (up to version 7), the output consisted of a mixture of a full grid dump (including zeroes) and a sparse matrix format - output was switched between these two formats based on which one was smaller.

In version 8.0, the output has been redesigned completely for yet more efficiency such that output file sizes are only about 40% of what they used to be. The output grid is searched for consecutive sequences of non-zero values. The variable `sp_count_i` gives the number of such sequences ( $n$ ), and the integer field `sparse_dump_i(n)` contains the field positions of the first non-zero element for every sequence. The variable `sp_count_r` gives the total number ( $k$ ) of non-zero values written out. They are contained in the real field `sparse_dump_r(n)`. Since all physical output quantities of FLEXPART are non-zero, sequences are written out alternately as positive or negative values. Every switch between positive and negative values indicates that a new sequence with non-zero values starts. The field position for that start is contained in `sparse_dump_i(k)` for the  $k$ -th switch between positive and negative. Zero values in between sequences are ignored and not written out. Field positions within the 3-d output field are coded such that a single integer value is sufficient. It can later be converted back to give positions in all three coordinates.

The possibility to output the sensitivities to initial conditions for backward runs has been introduced with FLEXPART version 8.2. This option can be used to calculate the exact sensitivities of the concentrations (or mixing ratios, depending on what is simulated) to initial conditions either in concentration or mixing ratio units taken from a gridded data set, which can be produced either by another model or by a FLEXPART forward simulation. This option has been introduced to allow interfacing FLEXPART with other models. Multiplying the sensitivities in files `grid_initial_nnn` with the corresponding concentrations (or mixing ratios, depending on option chosen for switch `LINIT_COND` in file `COMMAND`) from the forward simulation at the interfacing time gives the concentration (or mixing ratio) response at the receptor (taking into account possible loss processes during the FLEXPART simulation time).

### 13.2 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.

### 13.3 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.

### 13.4 Clustered plume trajectories

Condensed particle output using the clustering algorithm described in section 9 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.

## 14 **pflexpart: a Python routine for the analysis of FLEXPART output**

To assist in the usage and analysis of FLEXPART data we have created a Python module that is available with this new release. The Python module ‘`pflexpart`’ enables the user to easily read and access the header and grid output data of the FLEXPART model runs. Furthermore, we provide some basic classes that assist in conducted standard analysis of backward and forward model runs.

The module is released under the same GPL license as FLEXPART. As open source code it is constantly undergoing revision and updates from the community. Thus, the core functionality of the module is described online. See: <http://transport.nilu.no/pflexpart>

The module is largely based on the well known Numpy and matplotlib Python modules as well as the “basemap”



tool box of matplotlib. Additionally, for efficiency in reading the data, there are several custom build modules that use the `readgrid.f` FORTRAN routines described above. Note, that some of these modules will require the user to compile them for their system to achieve maximum benefit.

The central framework to the module is the `pflexpart Header` class. This class will read a FLEXPART header file and provide some functionality toward data analysis. For instance, the `Header` class has a `fill_backward` method for backward runs that will calculate the Total Column and Footprint sensitivities from the `N` ageclasses used in the run. From this method, plotting of residence times and emission sensitivities is relatively simple.

Other features include wrapper functions around the matplotlib toolboxes that allow for plotting of the data easily. The functions `plot_totalcolumn` and `plot_footprint` for example are customized wrappers of the `plot_sensitivity` function. These functions take data objects that are created by the `fill_backward` method of the `Header` class and allow the user to create plots quickly. For forward runs, the `Header.readgrid` method can be used, again with the `plot_sensitivity` function of the `pflexpart` module.

For more details, the reader is referred again to the more frequently updated module home page.

## 15 Final remark

In this note, we have described the Lagrangian particle dispersion model FLEXPART in version 8.2. As FLEXPART is developed further this text will be kept up to date and will be accessible from the FLEXPART home page at <http://transport.nilu.no/flexpart>.

## 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 Holtlag, A. A. M.: Flux parameterization over land surfaces for atmospheric models. *J. Appl. Meteor.*, 30, 327–341, 1991.
- Belward, A.S., Estes, J.E., and Kline, K.D.: The IGBP-DIS 1-Km Land-Cover Data Set DISCover: A Project Overview. *Photogrammetric Engineering and Remote Sensing*, v. 65, no. 9, p. 1013–1020, 1999.
- 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.
- Bey I, Jacob DJ, Yantosca RM, et al.: Asian chemical outflow to the Pacific in spring: Origins, pathways, and budgets. *J. Geophys. Res.*, 106, 19, 23073–23095, 2001.
- 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., 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: Nieuwstadt F.T.M. and H. van Dop (ed.): *Atmospheric Turbulence and Air Pollution Modelling*. D. Reidel Publishing Company, Dordrecht, Holland, 1982.
- Helmig D., Ganzeveld, L., Butler, T. and Oltmans S.J.: The role of ozone atmosphere-snow gas exchange on polar, boundary-layer tropospheric ozone - a review and sensitivity analysis *Atmos. Chem. and Physics*, 7, 15–30, 2007.
- 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.
- Jacob D. J., Wofsy W. C.: Budgets of reactive nitrogen, hydrocarbons, and ozone over the amazon-forest during the wet season. *J. Geophys. Res.*, 95, 16737–16754, 1990.
- 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.
- Naeslund, E., and Thaning, L.: On the settling velocity in a nonstationary atmosphere. *Aerosol Sci. Technol.*, 14, 247–256, 1991.
- 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: Nodop, K. (editor): *ETEX Symposium on Long-Range Atmospheric Transport, Model Verification and Emergency Response*, European Commission EUR 17346, 151–154, 1997.
- Seibert, P.: Inverse modelling with a Lagrangian particle dispersion model: application to point releases over limited time intervals. In: Gryning, S. E., Schiermeier, F.A. (eds.): *Air Pollution Modeling and its Application XIV*. 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.
- 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, September 24–26, 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., Forster, C., Frank, A., Seibert, P., Wotawa, G.: Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2., *Atmos. Chem. Phys.*, 5, 2461–2474, 2005
- Stohl, A., Cooper, O. R., Damoah, R., Fehsenfeld, F. C., Forster, C., Hsie, E.-Y., Hbler, G., Parrish, D. D. and Trainer, M.: Forecasting for a Lagrangian aircraft campaign. *Atmos. Chem. Phys.*, 4, 1113–1124, 2004.
- 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.
- Walmsley J.L., Wesely M.L.: Modification of coded parametrizations of surface resistances to gaseous dry deposition. *Atmos. Environ.*, 30, 1181–1188, 1996.
- 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: Zannetti, P. (ed.): *Environmental Modeling, Vol. II*. 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 nr 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: Gryning, S.-E., Beyrich, F. and E. Batchvarova (editors): *The Determination of the Mixing Height – Current Progress and Problems*. 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: Melli P. and P. Zannetti (ed.): Environmental Modelling. Computational Mechanics Publications, Southampton, UK, 1992.

Zhang L.M., Moran M.D., Makar P.A., Brook J.R. and S. Gong Modelling gaseous dry deposition in AURAMS: a unified regional air-quality modelling system Atmos. Environ., 36, 537-560, 2002.

## Appendix A FLEXPART sample input files

### A1 The pathnames file

A file `pathnames` must exist in the directory where FLEXPART is started. It states the pathnames (absolute or relative) of input and output files:

```
/home/as/FLEXPART50/options/
/volc/as/contrace/modelresults/forward/
/volc/windcontrace/
/volc/windcontrace/AVAILABLE
/volc/nested/
/volc/nested/AVAILABLE
=====
```

Line 1: path where control files "COMMAND" and "RELEASES" are available  
 Line 2: name of directory where output files are generated  
 Line 3: path where meteorological fields are available (mother grid)  
 Line 4: full filename of "AVAILABLE"-file (mother grid)

Subsequent lines:

Line  $2n+3$ : path where meteorological fields are available (nested grid  $n$ )  
 Line  $2n+4$ : full filename of "AVAILABLE"-file (nested grid  $n$ )

Line below last pathname must be:

```
=====
```

The grids must be arranged such as that the coarse-scale nests come before the fine-scale nests. Multiple nests of the same nesting level are allowed. In that case, the order is arbitrary.

## A2 Files in directory windfields

The directory where the meteorological input data are stored, here called `windfields` (`/volc/windcontrace/` in the above example `pathnames` file), contains grib-code files containing the ECMWF data. All meteorological fields must have the same structure, i.e. the same computational domain and the same resolution. An example listing of this directory is given below.

AVAILABLE	EN01102806	EN01102815
EN01102800	EN01102809	EN01102818
EN01102803	EN01102812	EN01102821

The file names of the grib-code files and their validation dates and times (in UTC) must be listed in the file `AVAILABLE`. While it is practical to have this file reside in the same directory as the wind fields, this is no necessity and it can also be located elsewhere, as its file name is also given in the `pathnames` file.

DATE	TIME	FILENAME	SPECIFICATIONS
YYYYMMDD	HHMISS		
20011028	000000	EN01102800	ON DISC
20011028	030000	EN01102803	ON DISC
20011028	060000	EN01102806	ON DISC
20011028	090000	EN01102809	ON DISC
20011028	120000	EN01102812	ON DISC
20011028	150000	EN01102815	ON DISC
20011028	180000	EN01102818	ON DISC
20011028	210000	EN01102821	ON DISC

Since version 7.0, `FILENAME` can be up to 256 characters long. Thus, the path to a file can be directly placed in the `AVAILABLE` file. If this feature is used, line 3 in the `pathnames` file should be left empty.

DATE	TIME	FILENAME	SPECIFICATIONS
YYYYMMDD	HHMISS		
20011028	000000	/work_disk/data/EN01102800	ON DISC
20011028	030000	/work_disk/data/EN01102803	ON DISC
20011028	060000	/work_disk/data/EN01102806	ON DISC
20011028	090000	/work_disk/data/EN01102809	ON DISC
20011028	120000	/work_disk/data/EN01102812	ON DISC
20011028	150000	/work_disk/data/EN01102815	ON DISC
20011028	180000	/work_disk/data/EN01102818	ON DISC
20011028	210000	/work_disk/data/EN01102821	ON DISC

Nested wind fields must be stored in one or more different directory/ies, as specified in the `pathnames` file.

### A3 Files in directory options

The files in directory `options` are used to specify the model run. An example listing of `options` is given below.

```
AGECLASSES          IGBP_int1.dat          RECEPTORS          SPECIES
COMMAND             OH_7lev_agl.dat       RELEASES             surfdata.t
COMMAND.alternative OUTGRID               RELEASES.alternative surfdepo.t
COMMAND.reference   OUTGRID_NEST          RELEASES.reference
```

Here, `SPECIES` is a subdirectory containing a number of files (this feature was introduced in version 8.0):

```
SPECIES:
SPECIES_001 SPECIES_004 SPECIES_007
SPECIES_002 SPECIES_005 SPECIES_008
SPECIES_003 SPECIES_006 SPECIES_009
```

#### A3.1 File COMMAND

The most important configuration file for setting up a FLEXPART simulation is the `COMMAND` file which specifies (1) the simulation direction (either forward or backward), (2) the start and (3) the end time of the simulation, (4) the frequency  $T_c$  of the model output, (5) the averaging time  $\Delta T_c$  of model output, and (6) the intervals  $\Delta T_s$  at which concentrations are sampled, (7) the time constant for particle splitting  $\Delta t_s$ , (8) the synchronisation interval of FLEXPART, (9) the factor  $c_{tl}$  by which the time steps must be smaller than the Lagrangian time scale, and (10) the refinement factor for the time step used for solving the Langevin equation of the vertical component of the turbulent wind. If (9) ( $c_{tl}$ ) is negative, the Langevin equations are solved with constant time steps according to the synchronisation interval. In that case, the value of (10) is arbitrary. The synchronisation interval is the minimum time interval used by the model for all activities (such as concentration calculations, wet deposition calculations, interpolation of data, mesoscale wind fluctuations or output of data) other than the simulation of turbulent transport and dry deposition (if  $c_{tl} > 0$ ). Further switches determine (11) whether concentrations, mixing ratios, residence times or plume trajectories (or combinations thereof) are to be calculated, (12) the option of particle position dump either at the end of or continuously during the simulation, (13) on/off of subgrid terrain effect parameterization, (14) on/off of deep convection parameterization, (15) on/off calculation of age spectra, (16) continuation of simulation from previous particle dump, (17) write output for each `RELEASE` location on/off, (18) on/off for mass flux calculations and output, (19) on/off for the domain-filling option of FLEXPART, (20) an indicator that determines whether mass or mass mixing ratio units are to be used at the source, (21) an indicator that determines whether mass or mass mixing ratio units are to be used at the receptor, (22) on/off of additional compact dump of the positions of numbered particles, (23) on/off for the use of nested output fields, (24) write sensitivity to initial conditions in backward mode off/mass units/mass mixing ratio units.

Two versions of `COMMAND` may be used, which both can be read in by FLEXPART: the first contains formatted input (i.e., a mask to be filled for the various input options that must be filled in), the second contains largely unformatted input and is recommended for the more experienced FLEXPART user. The following example is for formatted input.

```
*****
*
*      Input file for the Lagrangian particle dispersion model FLEXPART
*      Please select your options
*
*****

1.  ___          3X, I2
    1
    LDIRECT      1 FOR FORWARD SIMULATION, -1 FOR BACKWARD SIMULATION

2.  _____ 3X, I8, 1X, I6
    20040626 000000
    YYYYMMDD HHMISS BEGINNING DATE OF SIMULATION

3.  _____ 3X, I8, 1X, I6
    20040816 120000
    YYYYMMDD HHMISS ENDING DATE OF SIMULATION

4.  ___          3X, I5
    7200
    SSSSS       OUTPUT EVERY SSSSS SECONDS

5.  ___          3X, I5
    7200
    SSSSS       TIME AVERAGE OF OUTPUT (IN SSSSS SECONDS)

6.  ___          3X, I5
    900
```

SSSSS	SAMPLING RATE OF OUTPUT (IN SSSSS SECONDS)
7. _____ 999999999 SSSSSSSS	3X, I9 TIME CONSTANT FOR PARTICLE SPLITTING (IN SECONDS)
8. _____ 900 SSSSS	3X, I5 SYNCHRONISATION INTERVAL OF FLEXPART (IN SECONDS)
9. ---.-- -5.0 CTL	4X, F6.4 FACTOR, BY WHICH TIME STEP MUST BE SMALLER THAN TL
10. --- 4 IFINE	4X, I3 DECREASE OF TIME STEP FOR VERTICAL MOTION BY FACTOR IFINE
11. - 3 IOUT	4X, I1 1 CONCENTRATION (RESIDENCE TIME FOR BACKWARD RUNS) OUTPUT, 2 MIXING RATIO OUTPUT, 3 BOTH, 4 PLUME TRAJECT., 5=1
12. - 2 IPOINT	4X, I1 PARTICLE DUMP: 0 NO, 1 EVERY OUTPUT INTERVAL, 2 ONLY AT END
13. _ 1 LSUBGRID	4X, I1 SUBGRID TERRAIN EFFECT PARAMETERIZATION: 1 YES, 0 NO
14. _ 1 LCONVECTION	4X, I1 CONVECTION: 1 YES, 0 NO
15. _ 0 LAGESPECTRA	4X, I1 AGE SPECTRA: 1 YES, 0 NO
16. _ 0 IPIN	4X, I1 CONTINUE SIMULATION WITH DUMPED PARTICLE DATA: 1 YES, 0 NO
17. _ 0 IOFR	4X, I1 IOUTPUTFOREACHREL CREATE AN OUTPUT FILE FOR EACH RELEASE LOCATION: 1 YES, 0 NO
18. _ 0 IFLUX	4X, I1 CALCULATE FLUXES: 1 YES, 0 NO
19. _ 0 MDOMAINFILL	4X, I1 DOMAIN-FILLING TRAJECTORY OPTION: 1 YES, 0 NO, 2 STRAT. O3 TRACER
20. _ 1 IND_SOURCE	4X, I1 1=MASS UNIT , 2=MASS MIXING RATIO UNIT
21. _ 1 IND_RECEPTOR	4X, I1 1=MASS UNIT , 2=MASS MIXING RATIO UNIT
22. _ 0 MQASILAG	4X, I1 QUASILAGRANGIAN MODE TO TRACK INDIVIDUAL PARTICLES: 1 YES, 0 NO
23. _ 0 NESTED_OUTPUT	4X, I1 SHALL NESTED OUTPUT BE USED? 1 YES, 0 NO
24. _ 2 LINIT_COND	4X, I1 INITIAL COND. FOR BW RUNS: 0=NO,1=MASS UNIT,2=MASS MIXING RATIO UNIT

1. Simulation direction, 1 for forward, -1 for backward in time  
(consult Seibert and Frank, 2004 for backward runs)
2. Beginning date and time of simulation. Must be given in format  
YYYYMMDD HHMISS, where YYYY is YEAR, MM is MONTH, DD is DAY, HH is HOUR,  
MI is MINUTE and SS is SECOND. Current version utilizes UTC.
3. Ending date and time of simulation. Same format as 3.
4. Average concentrations are calculated every SSSSS seconds.

5. The average concentrations are time averages of SSSSS seconds duration. If SSSSS is 0, instantaneous concentrations are outputted.
6. The concentrations are sampled every SSSSS seconds to calculate the time average concentration. This period must be shorter than the averaging time.
7. Time constant for particle splitting. Particles are split into two after SSSSS seconds, 2xSSSSS seconds, 4xSSSSS seconds, and so on.
8. All processes are synchronized with this time interval (lsynctime). Therefore, all other time constants must be multiples of this value. Output interval and time average of output must be at least twice lsynctime.
9. CTL must be >1 for time steps shorter than the Lagrangian time scale. If CTL<0, a purely random walk simulation is done.
10. IFINE=Reduction factor for time step used for vertical wind
11. IOUT determines how the output shall be made: concentration (ng/m<sup>3</sup>, Bq/m<sup>3</sup>), mixing ratio (pptv), or both, or plume trajectory mode, or concentration + plume trajectory mode. In plume trajectory mode, output is in the form of average trajectories.
12. IPOUT determines whether particle positions are outputted (in addition to the gridded concentrations or mixing ratios) or not. 0=no output, 1 output every output interval, 2 only at end of the simulation
13. Switch on/off subgridscale terrain parameterization (increase of mixing heights due to subgridscale orographic variations)
14. Switch on/off the convection parameterization
15. Switch on/off the calculation of age spectra: if yes, the file AGECLASSES must be available
16. If IPIN=1, a file "partposit\_end" from a previous run must be available in the output directory. Particle positions are read in and previous simulation is continued. If IPIN=0, no particles from a previous run are used
17. If IOUTPUTFOREACHRELEASE is set to 1, one output field for each location in the RELEASES file is created. For backward calculation this should be set to 1. For forward calculation both possibilities are applicable.
18. If IFLUX is set to 1, fluxes of each species through each of the output boxes are calculated. Six fluxes, corresponding to northward, southward, eastward, westward, upward and downward are calculated for each grid cell of the output grid. The control surfaces are placed in the middle of each output grid cell. If IFLUX is set to 0, no fluxes are determined.
19. If MDOMAINFILL is set to 1, the first box specified in file RELEASES is used as the domain where domain-filling trajectory calculations are to be done. Particles are initialized uniformly distributed (according to the air mass distribution) in that domain at the beginning of the simulation, and are created at the boundaries throughout the simulation period.
20. IND\_SOURCE switches between different units for concentrations at the source. NOTE that in backward simulations the release of computational particles takes place at the "receptor" and the sampling of particles at the "source".  
1=mass units (for bwd-runs = concentration)  
2=mass mixing ratio units
21. IND\_RECEPTOR switches between different units for concentrations at the receptor  
1=mass units (concentrations)  
2=mass mixing ratio units
22. MQUASILAG indicates whether particles shall be numbered consecutively (1) or with their release location number (0). The first option allows tracking of individual particles using the partposit output files
23. NESTED\_OUTPUT decides whether model output shall be made also for a nested output field (normally with higher resolution)
24. LINIT\_COND determines whether, for backward runs only, the sensitivity to initial conditions shall be calculated and written to output files  
0=no output, 1 or 2 determines in which units the initial conditions are provided.



### A3.2 File OUTGRID

The file OUTGRID specifies the output grid.

```
*****
*
*   Input file for the Lagrangian particle dispersion model FLEXPART
*   Please specify your output grid
*
*****

1.  -----
    -10.0000      4X,F11.4
    OUTLONLEFT   GEOGRAFICAL LONGITUDE OF LOWER LEFT CORNER OF OUTPUT GRID
                  (left boundary of the first grid cell - not its centre)

2.  -----
     40.0000      4X,F11.4
    OUTLATLOWER  GEOGRAFICAL LATITUDE OF LOWER LEFT CORNER OF OUTPUT GRID
                  (lower boundary of the first grid cell - not its centre)

3.  -----
     101          4X,I5
    NUMXGRID     NUMBER OF GRID POINTS IN X DIRECTION (= No. of cells + 1)

4.  -----
     47           4X,I5
    NUMYGRID     NUMBER OF GRID POINTS IN Y DIRECTION (= No. of cells + 1)

5.  -----
     0.500        4X,F10.3
    DXOUTLON    GRID DISTANCE IN X DIRECTION

6.  -----
     0.500        4X,F10.3
    DYOUTLAT    GRID DISTANCE IN Y DIRECTION

7.  -----
     100.0        4X, F7.1
    LEVEL 1     HEIGHT OF LEVEL (UPPER BOUNDARY)

8.  -----
     300.0        4X, F7.1
    LEVEL 2     HEIGHT OF LEVEL (UPPER BOUNDARY)

9.  -----
     600.0        4X, F7.1
    LEVEL 3     HEIGHT OF LEVEL (UPPER BOUNDARY)

10. -----
    1000.0        4X, F7.1
    LEVEL 4     HEIGHT OF LEVEL (UPPER BOUNDARY)

11. -----
    2000.0        4X, F7.1
    LEVEL 5     HEIGHT OF LEVEL (UPPER BOUNDARY)

12. -----
    3000.0        4X, F7.1
    LEVEL 6     HEIGHT OF LEVEL (UPPER BOUNDARY)
```

In order to define the grid for a nested output field, the file OUTGRID\_NEST must exist. It has the same format as file OUTGRID, but does not contain the vertical level information:

```
*****
*
*   Input file for the Lagrangian particle dispersion model FLEXPART
*   Please specify your output grid
*
*****

1.  -----
    -125.0000     4X,F11.4
    OUTLONLEFT   GEOGRAFICAL LONGITUDE OF LOWER LEFT CORNER OF OUTPUT GRID
                  (left boundary of the first grid cell - not its centre)

2.  -----
     25.0000     4X,F11.4
    OUTLATLOWER  GEOGRAFICAL LATITUDE OF LOWER LEFT CORNER OF OUTPUT GRID
                  (lower boundary of the first grid cell - not its centre)

3.  -----
     1           4X,I5
    NUMXGRID     NUMBER OF GRID POINTS IN X DIRECTION (= No. of cells + 1)

4.  -----
                4X,I5
```

```
      1          NUMBER OF GRID POINTS IN Y DIRECTION (= No. of cells + 1)
NUMYGRID

5.  -----
    0.33333      4X,F12.5
DXOUTLON      GRID DISTANCE IN X DIRECTION

6.  -----
    0.25000      4X,F12.5
DYOUTLAT      GRID DISTANCE IN Y DIRECTION
```

### A3.3 File RECEPTORS

RECEPTORS specifies the receptor locations for which the parabolic kernel method shall be applied to calculate air concentrations. The maximum number of receptor sites is set by parameter `maxreceptor` in file `includepar`.

```
*****
*
*      Input file for the Lagrangian particle dispersion model FLEXPART
*      Please specify your receptor points
*      For the receptor points, ground level concentrations are calculated
*
*****
1. ----- 4X,A16
   F15      NAME OF RECEPTOR POINT
   RECEPTORNAME

2. ----- 4X,F11.4
   6.1333   GEOGRAFICAL LONGITUDE
   XRECEPTOR

3. ----- 4X,F11.4
   49.0833  GEOGRAFICAL LATITUDE
   YRECEPTOR
=====
1. ----- 4X,A16
   NL01     NAME OF RECEPTOR POINT
   RECEPTORNAME

2. ----- 4X,F11.4
   5.7833   GEOGRAFICAL LONGITUDE
   XRECEPTOR

3. ----- 4X,F11.4
   50.9167  GEOGRAFICAL LATITUDE
   YRECEPTOR
=====
```

### A3.4 File RELEASES

RELEASES defines the release specifications. In the first input line, the number  $N$  of emitted species is defined (1 in the example below). At all locations, the same species must be released. The next  $N$  input lines give a cross-reference to the respective file SPECIES\_nnn, where the physical and chemical properties of the released species are given (also the temporal variations of emissions is defined for each species). Then follows a list of release sites, for each of which the release characteristics must be entered: the beginning and the ending time of the release, geographical coordinates of the lower left and upper right corners of the release location, type of vertical coordinate (above ground level, or above sea level), lower level and upper level of source box, the number of particles to be used, and the total mass emitted. Note that the mass entry must be repeated  $N$  times, one mass per species released. Finally, a name is assigned to each release point.

The particles are released from random locations within a four-dimensional box extending from the lower to the upper level above a rectangle (on a lat/lon grid) defined by the geographical coordinates, and between the release's start and end. With some of the coordinates set identically, line or point sources can be specified.

As for COMMAND, the RELEASES file can be provided formatted or unformatted. The example below shows the formatted version.

```
*****
*
*
*
*   Input file for the Lagrangian particle dispersion model FLEXPART
*   Please select your options
*
*
*
*****
+++++
 1
-----
          i3   Total number of species emitted

 24
-----
          i3   Index of species in file SPECIES

=====
20011028  150007
-----
          i8,1x,i6 Beginning date and time of release

20011028  150046
-----
          i8,1x,i6 Ending date and time of release

 9.4048
-----
          f9.4  Longitude [DEG] of lower left corner

48.5060
-----
          f9.4  Latitude [DEG] of lower left corner

 9.5067
-----
          f9.4  Longitude [DEG] of upper right corner

48.5158
-----
          f9.4  Latitude [DEG] of upper right corner

 2
-----
          i9    1 for m above ground, 2 for m above sea level, 3 for pressure in hPa

6933.60
-----
          f10.3 Lower z-level (in m agl or m asl)

6950.40
-----
          f10.3 Upper z-level (in m agl or m asl)

20000
-----
          i9    Total number of particles to be released

1.0000E00
-----
          e9.4  Total mass emitted

FLIGHT_11242
-----
          character*40 comment
+++++
20011028  150047
-----
          i8,1x,i6 Beginning date and time of release

20011028  150107
-----
          i8,1x,i6 Ending date and time of release

 9.3038
-----
          f9.4  Longitude [DEG] of lower left corner
```

```

48.5158
_____.____
f9.4 Latitude [DEG] of lower left corner

9.4048
_____.____
f9.4 Longitude [DEG] of upper right corner

48.5906
_____.____
f9.4 Latitude [DEG] of upper right corner

2
_____
i9 1 for m above ground, 2 for m above sea level, 3 for pressure in hPa

6833.50
_____.____
f10.3 Lower z-level (in m agl or m asl)

6950.40
_____.____
f10.3 Upper z-level (in m agl or m asl)

20000
_____
i9 Total number of particles to be released

1.0000E00
_._.____E____
e9.4 Total mass emitted

FLIGHT_11185
_____ character*40 comment
+++++
```

### A3.5 File AGECLASSES

AGECLASSES provides the times for the age class calculation. In the first data line, the number  $n$  of age classes is set, and ages are listed in the following  $n$  lines. The entries specify the end times (in seconds) of the respective intervals to be used, the first one starting at zero seconds. Particles are dropped from the simulation once they exceed the maximum age. Even if no age classes are needed, this option (with the number of age classes set to 1) can be useful to determine the age at which particles are removed from the simulation.

```
*****
*
*Lagrangian particle dispersion model FLEXPART *
*   Please select your options               *
*
*This file determines the ageclasses to be used*
*
*Ages are given in seconds. The first class   *
*starts at age zero and goes up to the first  *
*age specified. The last age gives the maximum *
*time a particle is carried in the simulation. *
*
*****
 6          Integer      Number of age classes
43200      Integer      Age class 1
86400      Integer      Age class 2
129600
172800
259200
345600
```

## A3.6 Files SPECIES\_nnn and surfdata.t

SPECIES\_nnn (where nnn is a zero-padded identifier of the species number) specifies all physico-chemical properties for the given species. Entries are the half life (due to radioactive or chemical decay), wet deposition information (A and B are the factors defined by Eq. 44), dry deposition information for gases ( $D = D_{H_2O}/D_i$ ,  $D_{H_2O}$  is the diffusivity of water vapor and  $D_i$  is the diffusivity of the species, H is the effective Henry's constant, and f0 varies between 0 and 1 and gives the reactivity of a species relative to that of ozone. For nonreactive species f0 is 0, for slightly reactive it is 0.1 and for highly reactive it is 1.), dry deposition information for particulates (rho specifies the density of the substance, dquer its mean diameter  $\overline{d_p}$ , and dsig the measure of variation  $\sigma_p$ ). Radioactive decay is switched off by specifying a negative half life, wet deposition is switched off by specifying negative A, dry deposition of gases is switched off by negative D, dry deposition of particles is switched off by negative rho. If no detailed information for deposition velocity calculation is available, a constant deposition velocity vd (cm s<sup>-1</sup>) can be used. molweight gives the molecular weight of the species, which is needed for mixing ratio output. For degradation in an monthly averaged OH field (Bey et al., 2001) the OH Reaction ratio at 25 °C for the species can be given, units are (cm<sup>3</sup>/s).

Optionally an emission variation information can be added at the end of the file, defined as following: Since FLEXPART version 6.0, emission factors can be defined that change the temporal variation of particle releases. This is useful, for instance, to simulate the typical daily and weekly cycle of anthropogenic emissions. The emission factors are given in the file of the corresponding species SPECIES\_nnn, where nnn is the species number defined in file RELEASES. If no emission variation information is given, emission rates for species nnn are taken as constant. Release rates can vary with the hour of the day and with the day of the week, according to the local time at the release location. Emission factors must be 1 on average. 24 hourly as well as 7 daily values must be specified. Furthermore, different disaggregation factors must be given for area sources and for point sources. FLEXPART distinguishes between the two using the lower altitude of the release box: area sources are assumed to start below 0.5 m above the ground, whereas point sources are assumed to be higher. Please note that when this option is used, it is not so easy to determine the maximum number of particles present at a particular time of the model run. It might then be necessary to increase the parameter maxpart to a higher value than what would otherwise be needed. The following is an example for an SPECIES\_nnn file including emission variation.

```
*****
*
*      Input file for the Lagrangian particle dispersion model FLEXPART      *
*      Definition file of chemical species/radionuclides                    *
*
*****
EU-CO          Tracer name
-999.9         Species half life
-9.9E-09       Wet deposition - A
              Wet deposition - B
-9.9          Dry deposition (gases) - D
              Dry deposition (gases) - Henrys const.
              Dry deposition (gases) - f0 (reactivity)
-9.9E09       Dry deposition (particles) - rho
              Dry deposition (particles) - dquer
              Dry deposition (particles) - dsig
-9.99        Alternative: dry deposition velocity
28.00         molweight
-9.9E-09      OH Reaction rate at 25 deg, [cm^3/s]
-9           number of associated specias (neg. none) - not implemented yet
-99.99       KOA - organic matter air partitioning
hr_start co_area co_point
0          0.535  0.932    0-1 local time
1          0.405  0.931    1-2 local time
2          0.317  0.927
3          0.265  0.926
4          0.259  0.928
5          0.367  0.936
6          0.668  0.952
7          1.039  0.975
8          1.015  1.046
9          0.965  1.055
10         1.016  1.061
11         1.133  1.064
12         1.269  1.067
13         1.368  1.068
14         1.516  1.069
15         1.681  1.068
16         1.777  1.024
17         1.827  1.017
18         1.538  1.008
19         1.282  1.007
20         1.136  1.004
21         1.020  0.996
22         0.879  0.981
```

23	0.723	0.958	23-24 local time
week_day	co_area	co_point	
1	1.060	1.000	Monday
2	1.060	1.000	Tuesday
3	1.060	1.000	Wednesday
4	1.060	1.000	Thursday
5	1.060	1.000	Friday
6	0.900	1.000	Saturday
7	0.800	1.000	Sunday

IGBP\_intl.dat contains the landuse inventory in binary format, and surfdata.t, shown below, gives the roughness lengths for each landuse class:

13 landuse categories are related roughness length

landuse	comment	z0
1	Urban land	0.7
2	Agricultural land	0.1
3	Range land	0.1
4	Deciduous forest	1.
5	Coniferous forest	1.
6	Mixed forest including wetland	0.7
7	Water, both salt and fresh	0.001
8	Barren land mostly desert	0.01
9	Nonforested wetland	0.1
10	Mixed agricultural and range land	0.1
11	Rocky open areas with low growing shrubs	0.05
12	Snow and ice	0.001
13	Rainforest	1.



A3.7 File surfdepo.t

surfdepo.t gives the resistances needed for the parameterization of dry deposition of gases for the 13 landuse classes and five seasonal categories. This file must not be changed by the user.

DRY DEPOSITION

=====

AFTER WESELY, 1989

=====

1 to 11: Landuse types after Wesely; 12 .. snow, 13 .. rainforest

=====

Values are tabulated for 5 seasonal categories:

- 1 Midsummer with lush vegetation
  - 2 Autumn with unharvested cropland
  - 3 Late autumn after frost, no snow
  - 4 Winter, snow on ground and subfreezing
  - 5 Transitional spring with partially green short annuals
- =====

	1	2	3	4	5	6	7	8	9	10	11	12	13	
ri	9999.	60.	120.	70.	130.	100.	9999.	9999.	80.	100.	150.	9999.	200.	1
rlu	9999.	2000.	2000.	2000.	2000.	2000.	9999.	9999.	2500.	2000.	4000.	9999.	1000.	
rac	100.	200.	100.	2000.	2000.	2000.	0.	0.	300.	150.	200.	0.	2000.	
rgss	400.	150.	350.	500.	500.	100.	0.	1000.	0.	220.	400.	100.	200.	
rgso	300.	150.	200.	200.	200.	300.	2000.	400.	1000.	180.	200.	10000.	200.	
rcls	9999.	2000.	2000.	2000.	2000.	2000.	9999.	9999.	2500.	2000.	4000.	9999.	9999.	
rclo	9999.	1000.	1000.	1000.	1000.	1000.	9999.	9999.	1000.	1000.	1000.	9999.	9999.	
ri	9999.	9999.	9999.	9999.	250.	500.	9999.	9999.	9999.	9999.	9999.	9999.	200.	2
rlu	9999.	9000.	9000.	9000.	4000.	8000.	9999.	9999.	9000.	9000.	9000.	9999.	1000.	
rac	100.	150.	100.	1500.	2000.	1700.	0.	0.	200.	120.	140.	0.	2000.	
rgss	400.	200.	350.	500.	500.	100.	0.	1000.	0.	300.	400.	100.	200.	
rgso	300.	150.	200.	200.	200.	300.	2000.	400.	800.	180.	200.	10000.	200.	
rcls	9999.	9000.	9000.	9000.	2000.	4000.	9999.	9999.	9000.	9000.	9000.	9999.	9999.	
rclo	9999.	400.	400.	400.	1000.	600.	9999.	9999.	400.	400.	400.	9999.	9999.	
ri	9999.	9999.	9999.	9999.	250.	500.	9999.	9999.	9999.	9999.	9999.	9999.	200.	3
rlu	9999.	9999.	9000.	9000.	4000.	8000.	9999.	9999.	9000.	9000.	9000.	9999.	1000.	
rac	100.	10.	100.	1000.	2000.	1500.	0.	0.	100.	50.	120.	0.	2000.	
rgss	400.	150.	350.	500.	500.	200.	0.	1000.	0.	200.	400.	100.	200.	
rgso	300.	150.	200.	200.	200.	300.	2000.	400.	1000.	180.	200.	10000.	200.	
rcls	9999.	9999.	9000.	9000.	3000.	6000.	9999.	9999.	9000.	9000.	9000.	9999.	9999.	
rclo	9999.	1000.	400.	400.	1000.	600.	9999.	9999.	800.	600.	600.	9999.	9999.	
ri	9999.	9999.	9999.	9999.	400.	800.	9999.	9999.	9999.	9999.	9999.	9999.	200.	4
rlu	9999.	9999.	9999.	9999.	6000.	9000.	9999.	9999.	9000.	9000.	9000.	9999.	1000.	
rac	100.	10.	10.	1000.	2000.	1500.	0.	0.	50.	10.	50.	0.	2000.	
rgss	100.	100.	100.	100.	100.	100.	0.	1000.	100.	100.	50.	100.	200.	
rgso	600.	3500.	3500.	3500.	3500.	3500.	2000.	400.	3500.	3500.	3500.	10000.	200.	
rcls	9999.	9999.	9999.	9000.	200.	400.	9999.	9999.	9000.	9999.	9000.	9999.	9999.	
rclo	9999.	1000.	1000.	400.	1500.	600.	9999.	9999.	800.	1000.	800.	9999.	9999.	
ri	9999.	120.	240.	140.	250.	190.	9999.	9999.	160.	200.	300.	9999.	200.	5
rlu	9999.	4000.	4000.	4000.	2000.	3000.	9999.	9999.	4000.	4000.	8000.	9999.	1000.	
rac	100.	50.	80.	1200.	2000.	1500.	0.	0.	200.	60.	120.	0.	2000.	
rgss	500.	150.	350.	500.	500.	200.	0.	1000.	0.	250.	400.	100.	200.	
rgso	300.	150.	200.	200.	200.	300.	2000.	400.	1000.	180.	200.	10000.	200.	
rcls	9999.	4000.	4000.	4000.	2000.	3000.	9999.	9999.	4000.	4000.	8000.	9999.	9999.	
rclo	9999.	1000.	500.	500.	1500.	700.	9999.	9999.	600.	800.	800.	9999.	9999.	