FLEXPART training course 2013

D. Arnold[a] with input from many others

[a] Zentralanstalt für Meteorologie und Geodynamik (ZAMG), Vienna, Austria

L1

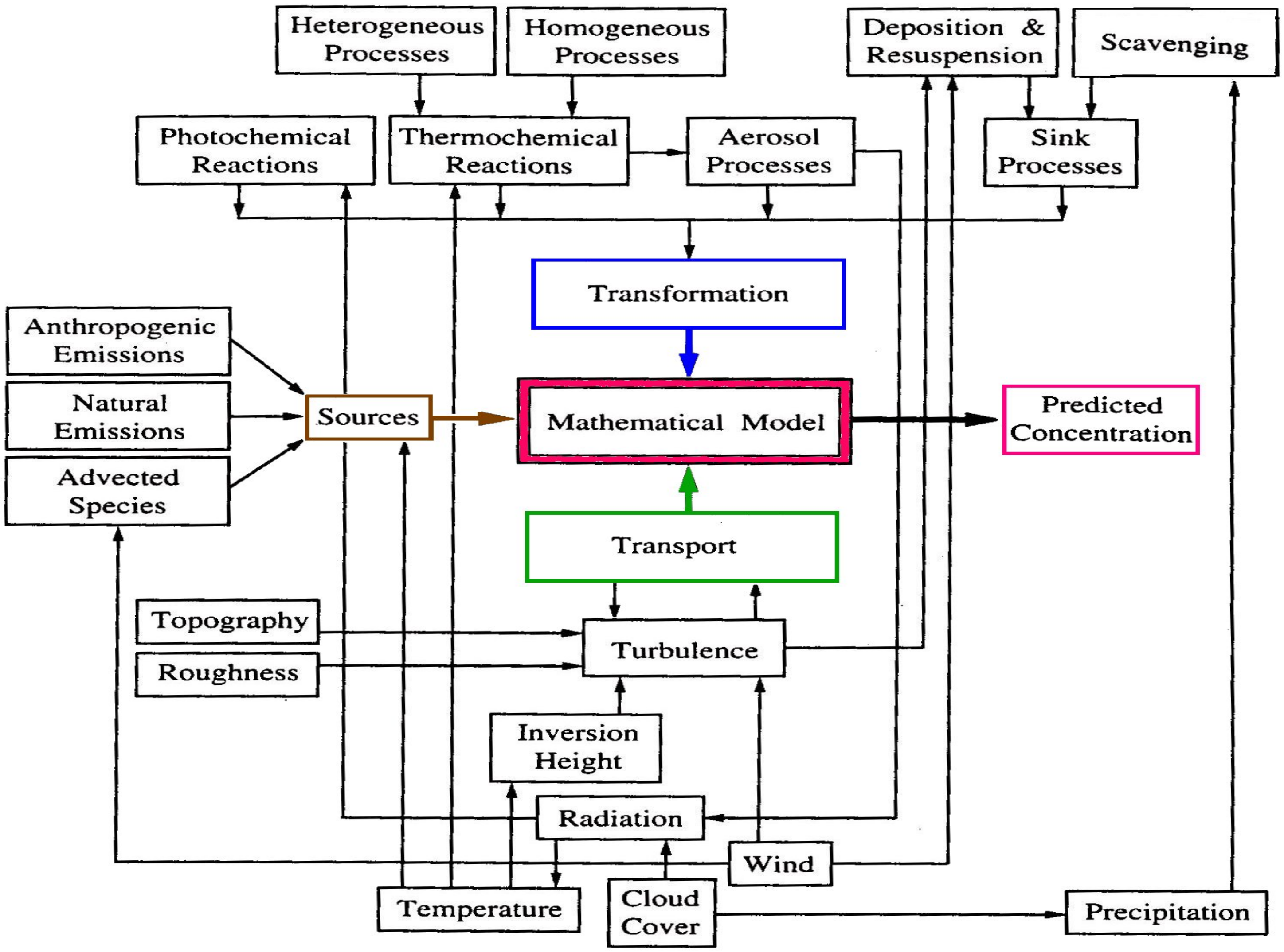
Dispersion modelling background

This is ***not*** intended to be a detailed explanation. Please refer to usual textbooks such as:

- Zannetti, P. (1990). Air pollution modeling : theories, computational methods, and available software. Van Nostrand Reinhold. ISBN 0-442-30805-1.
- Arya S P 1998 Air Pollution Meteorology and Dispersion Oxford University Press (New York)
- Intercontinental Transport of Air Pollution, Andreas Stohl (Ed.) in The Handbook of Environmental Chemistry, Vol. 4, Part G, pp 99-130, Springer-Verlag, 2004.
- Turner, D.B. (1994). Workbook of atmospheric dispersion estimates: an introduction to dispersion modeling (2nd Edition ed.). CRC Press
- References from Uliasz Mezcua, A. Stohl, R.Pielke …

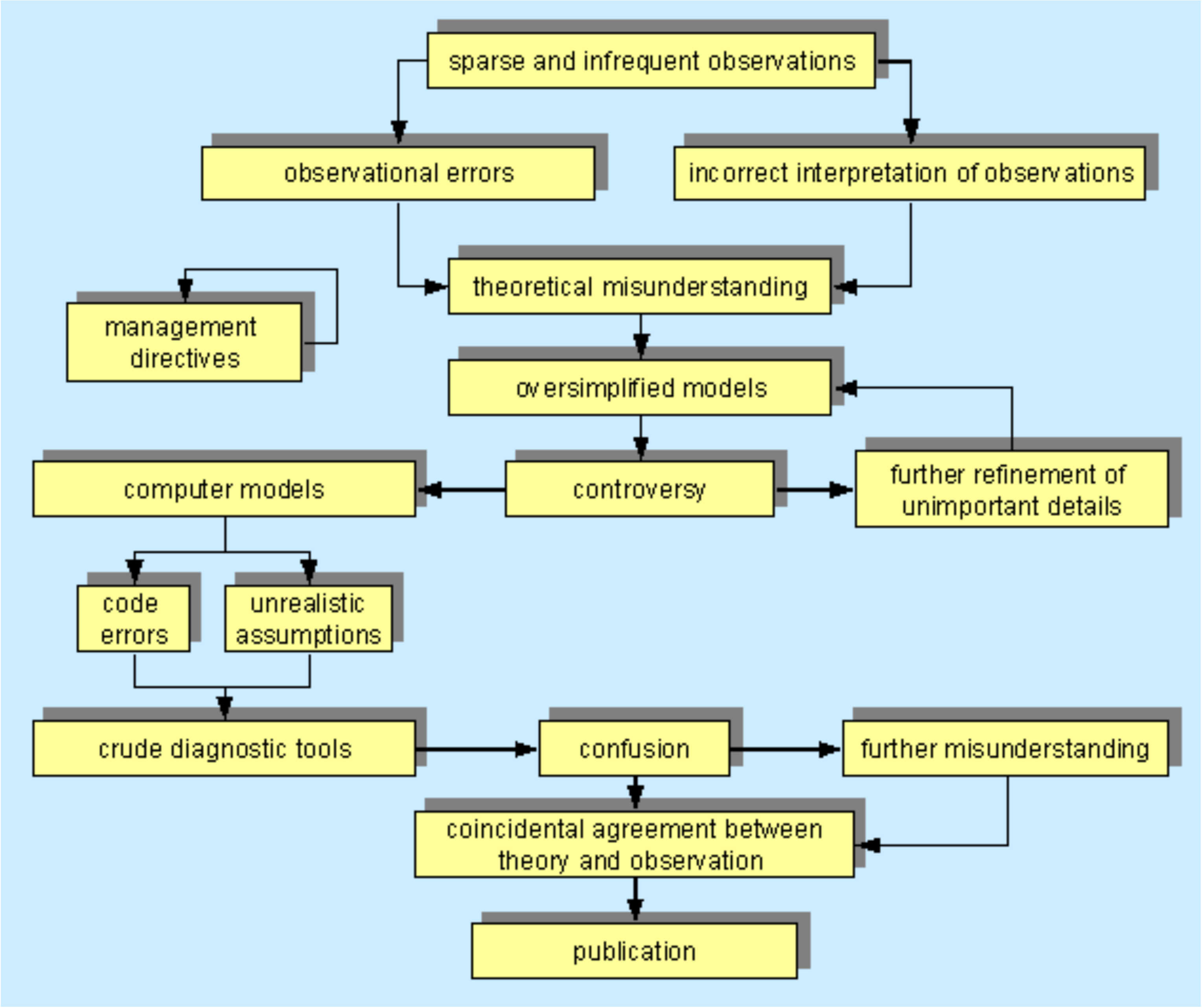
Dispersion modelling background

Seinfeld and Pandis 1998



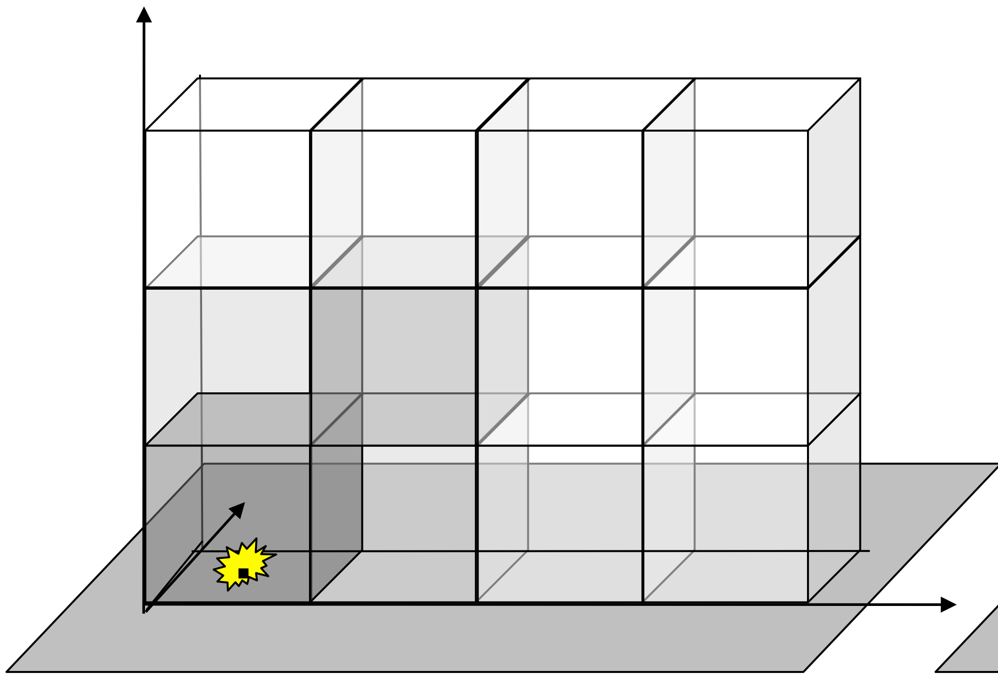
Dispersion modelling background

Uliasz

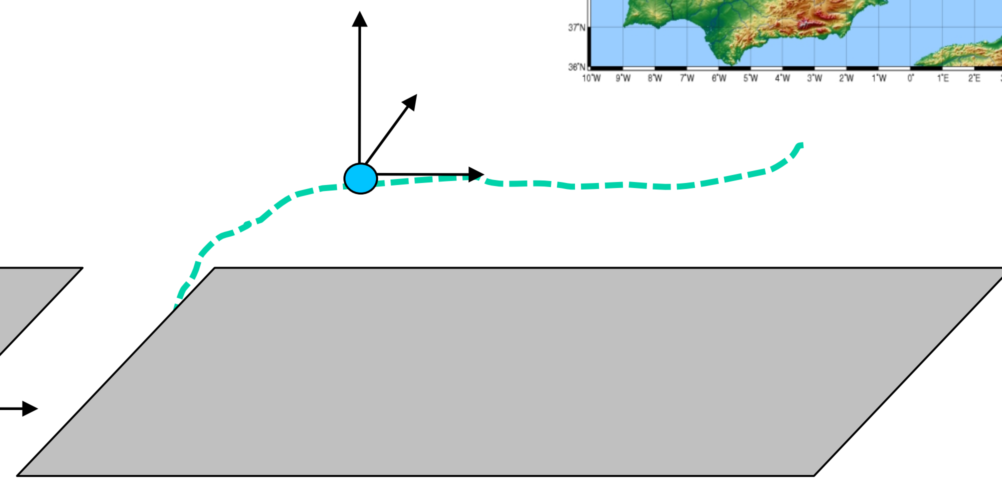


Modelling background

Eulerian



Lagrangian



$$\frac{\partial c_i}{\partial t} + \boxed{u_x \frac{\partial c_i}{\partial x} + u_y \frac{\partial c_i}{\partial y} + u_z \frac{\partial c_i}{\partial z}}$$

Divergence of the advected flux

$$= \frac{\partial}{\partial x}\left(K_{xx}\frac{\partial c_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{yy}\frac{\partial c_i}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_{zz}\frac{\partial c_i}{\partial z}\right) + R_i(c_1, c_2 \cdots, c_n) + E_i(x,y,z,t) - S_i(x,y,z,t)$$

Chemical reactions

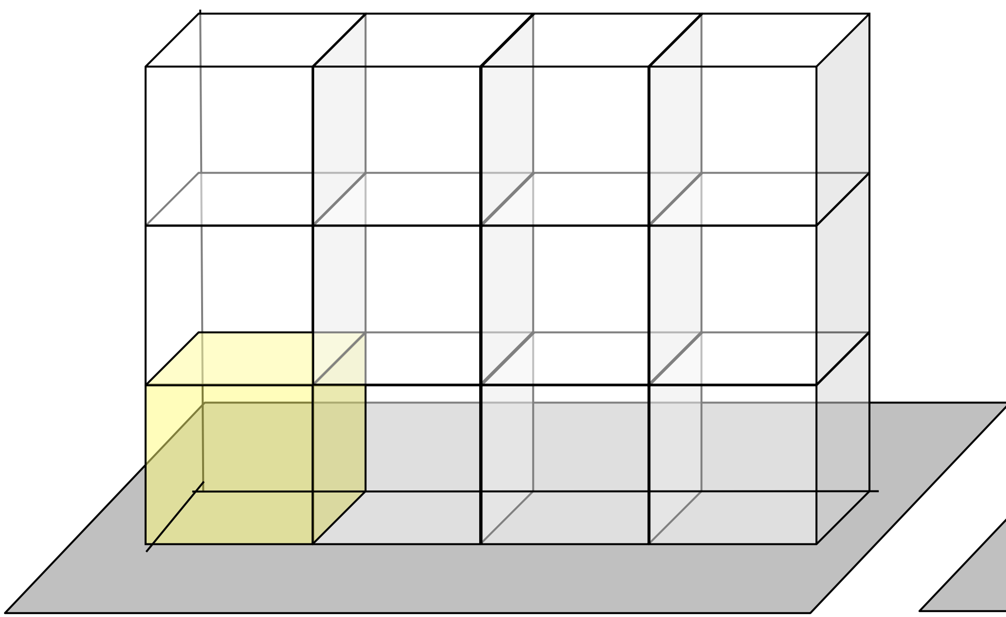
Sinks

Divergence of the turbulent fluxes

Emissions

Modelling background

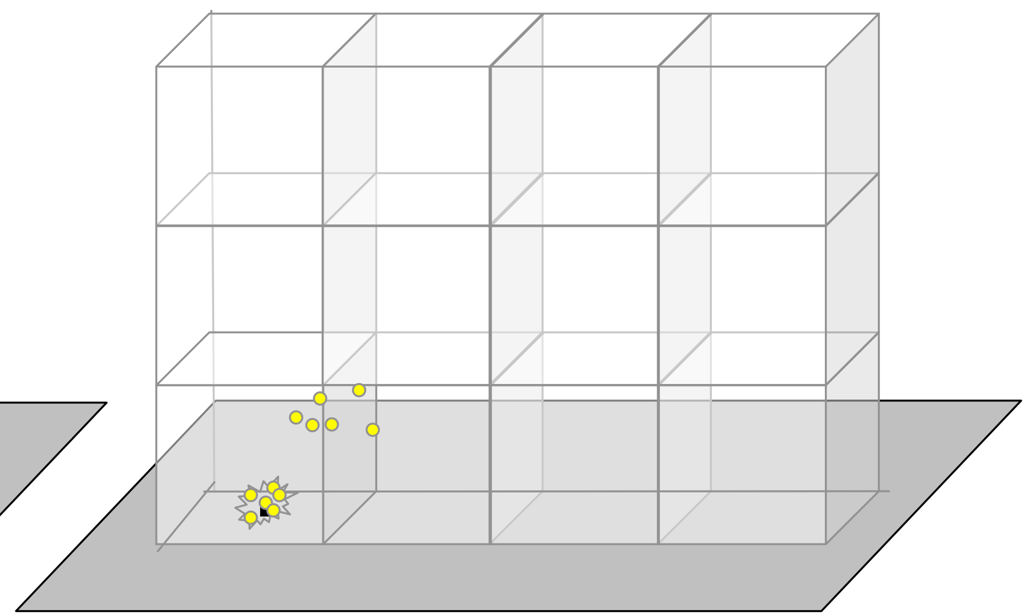
Eulerian



Immediate dilution in the grid cell

Point source sub-model then needed

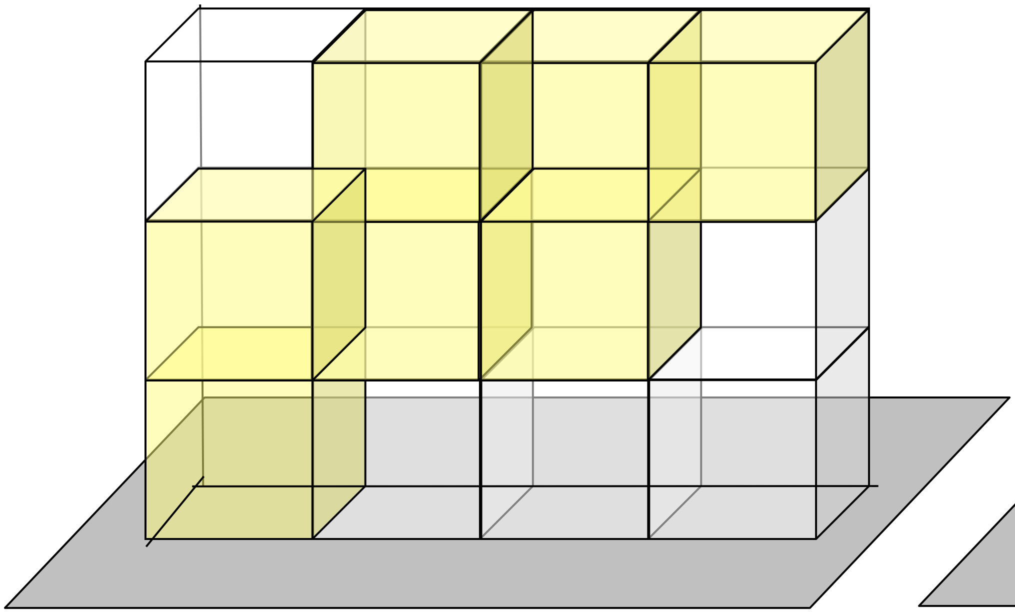
Lagrangian



LPDM can deal naturally with point sources

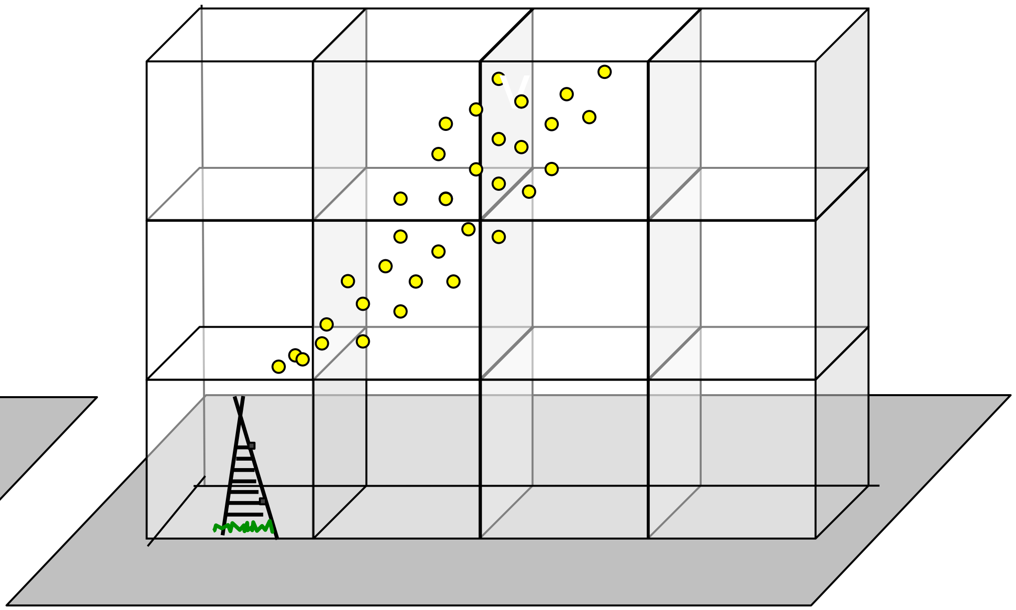
The grid is only applied to output fields

Eulerian



Problems with representing narrow plumes

Lagrangian



Modelling background

Eulerian

Fig. 10a Initial isolated puff

MSC-W Note 2/92, August 1992.EMEP "An Evaluation of Eulerian Advection Methods for the Modelling of Long Range Transport of Air Pollution". By Erik Berge and Leonor Tarrasón. EMEP_1992_N2.pdf

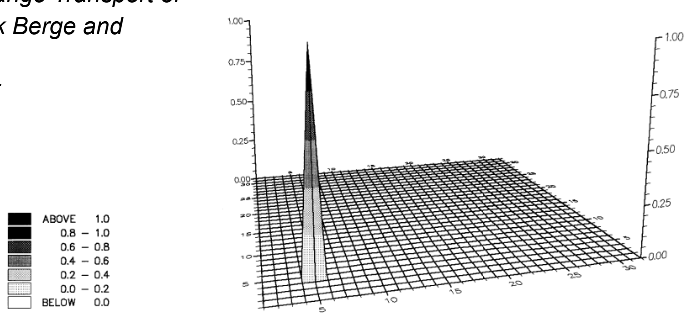


Fig. 10d BOS: Diagonal puff

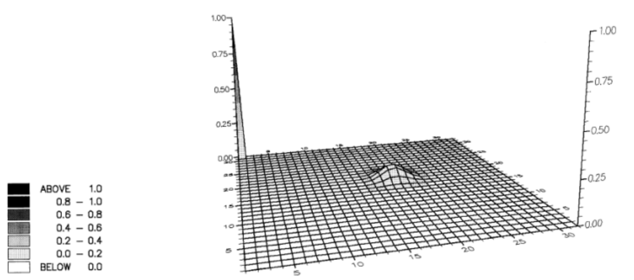
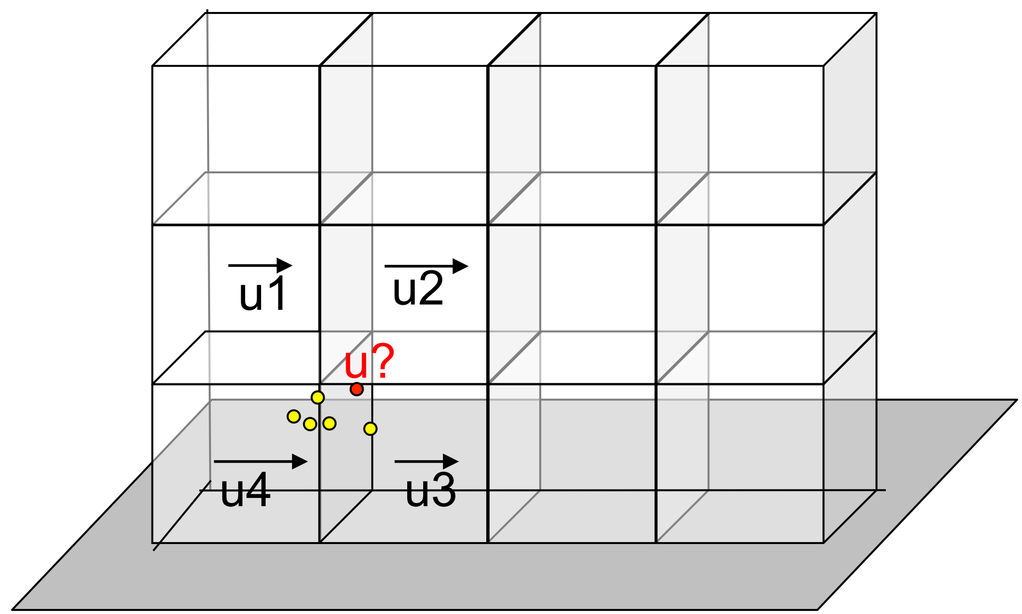


Fig. 10e PSS: Diagonal puff

Numerical diffusion in the advection

Lagrangian



Interpolation errors (of all variables to particle position)

Why Lagrangian?

- Can be computationally very efficient (depending on size of plume): only the fraction covered with particles is simulated.
- Turbulent processes are included in a more natural way unlike Eulerian models
- There is no numerical diffusion due to a computational grid
- Grid and/or kernels are used only for output purpose therefore no artificial diffusion is due to the averaging process
- Model is “self-adjoint” – can run backward in time, too.
- Many first order processes can be easily included with a prescribed rate: radioactive decay, dry deposition, washout, etc.
- One particle can carry more than one species
- Gravitational settling is easily included (as long as particles carry a single species)
- *However: it is quite difficult and computationally expensive to include non-linear chemical reactions and the process of gridding the output make as well loose some of the advantages of Lagrangian modelling.*

What is FLEXPART – validation, users

The FLEXPART is…

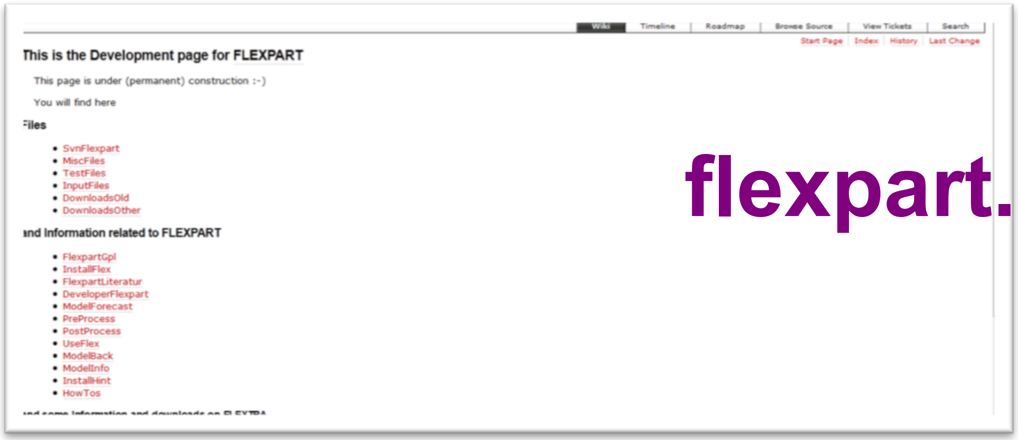
… a Lagrangian Particle Dispersion Model, originally developed at the University of Natural Resources and Life Sciences in Vienna, further developed by its main developer Andreas Stohl at the Norwegian Institute for Air Research in the Department of Atmospheric and Climate Research and with by group of developers in different institutions

It is released under the GNU General Public License V3.0

- Countries – 15
- Users http://transport.nilu.no/flexpart/flexpart-and-flextra-users >35
  - Operational → CTBTO, ZAMG, Meteoswiss, ...
  - Research → NILU, BOKU-Met, NOAA, INTE, ARSC, ...
  - Others

flexpart.eu

What is FLEXPART – validation, users


This is the Development page for FLEXPART

This page is under (permanent) construction :-)

You will find here

Files

- SvnFlexpart
- MiscFiles
- TestFiles
- InputFiles
- DownloadsOld
- DownloadsOther

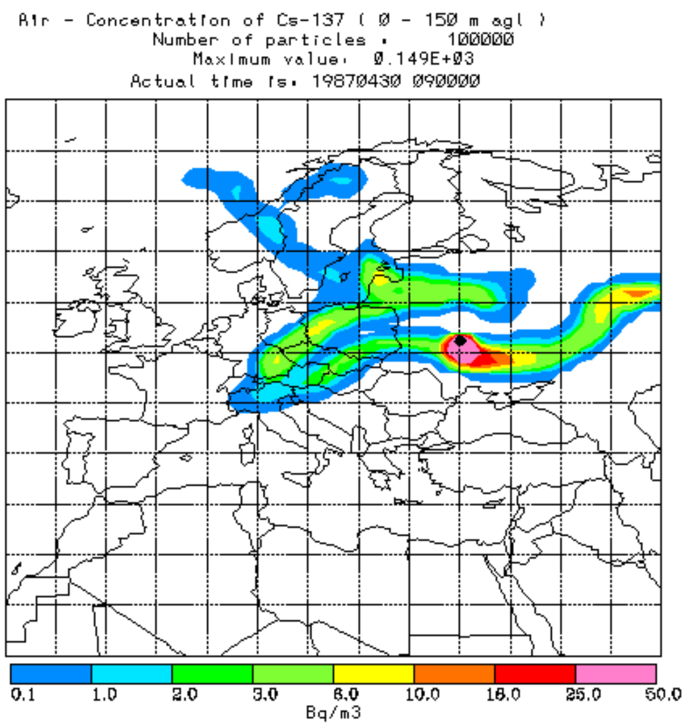
and Information related to FLEXPART

- FlexpartGpl
- InstallFlex
- FlexpartLiteratur
- DeveloperFlexpart
- ModelForecast
- PreProcess
- PostProcess
- UseFlex
- ModelBack
- ModelInfo
- InstallHint
- HowTos

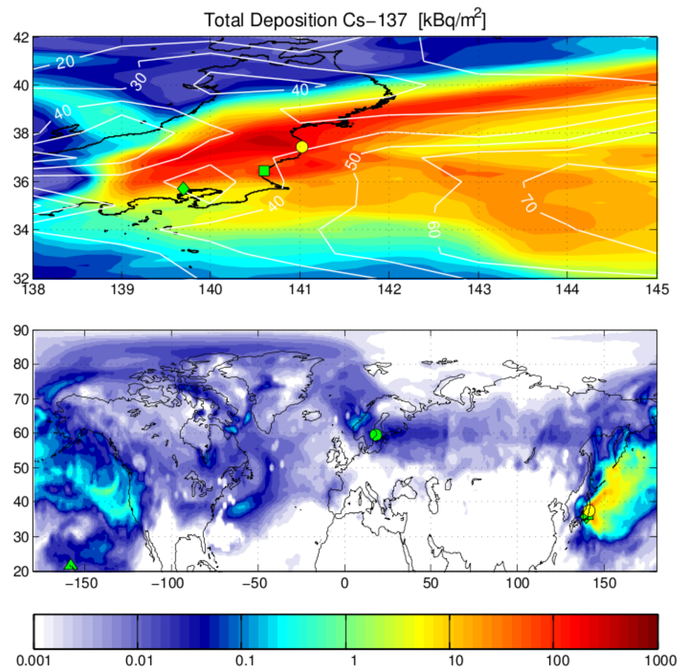
flexpart.eu

- Get source code
- Generate tickets that will be addressed by developers
- Get updates and references
- Get post-processing software
- Get test data
- … and the course notes and data UNDER DEVELOPMENT

Nuclear applications – to study consequences and source terms of events such as Chernobyl and Fukushima



From P. Seibert and A. Frank (BOKU- Met) unpublished



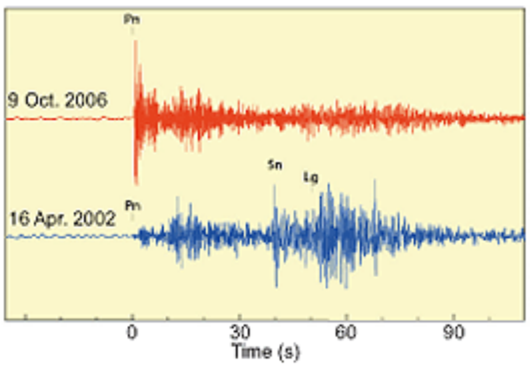
From Stohl et al. (2012)

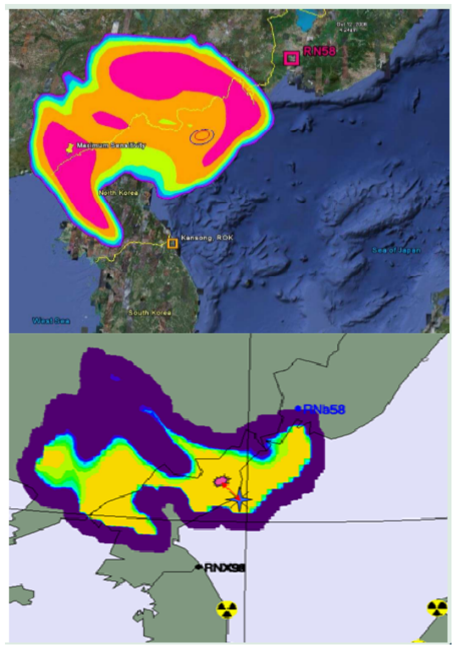
- Source in a single location – complicated release shape in time and height
- From regional to global scales
- Fwd runs

Examples of Applications

Nuclear applications - Operational

CTBTO - to monitor compliance with the comprehensive ban on nuclear testing

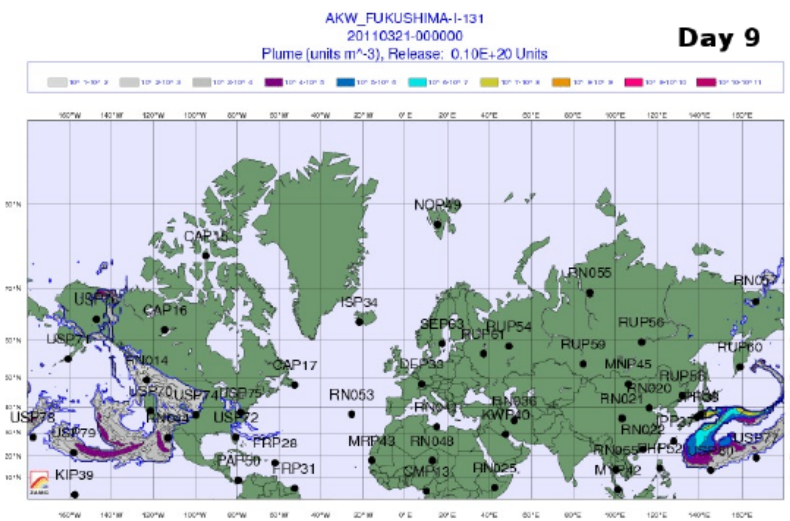




http://www.ctbto.org/fileadmin/user_upload/ISS_2009/Poster/ATM-10E%20%28PTS%29%20-%20A_Becker%20and%20G_Wotawa.pdf

- Point source (measurement site)
- From regional to global scales
- Bwd runs

ZAMG - forecast

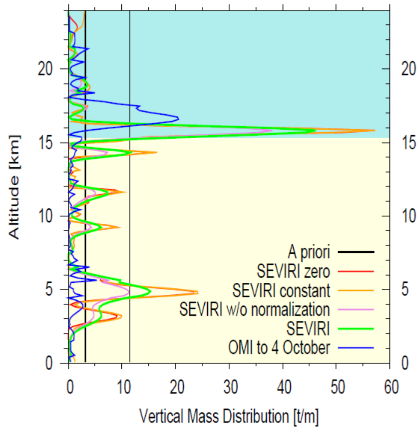


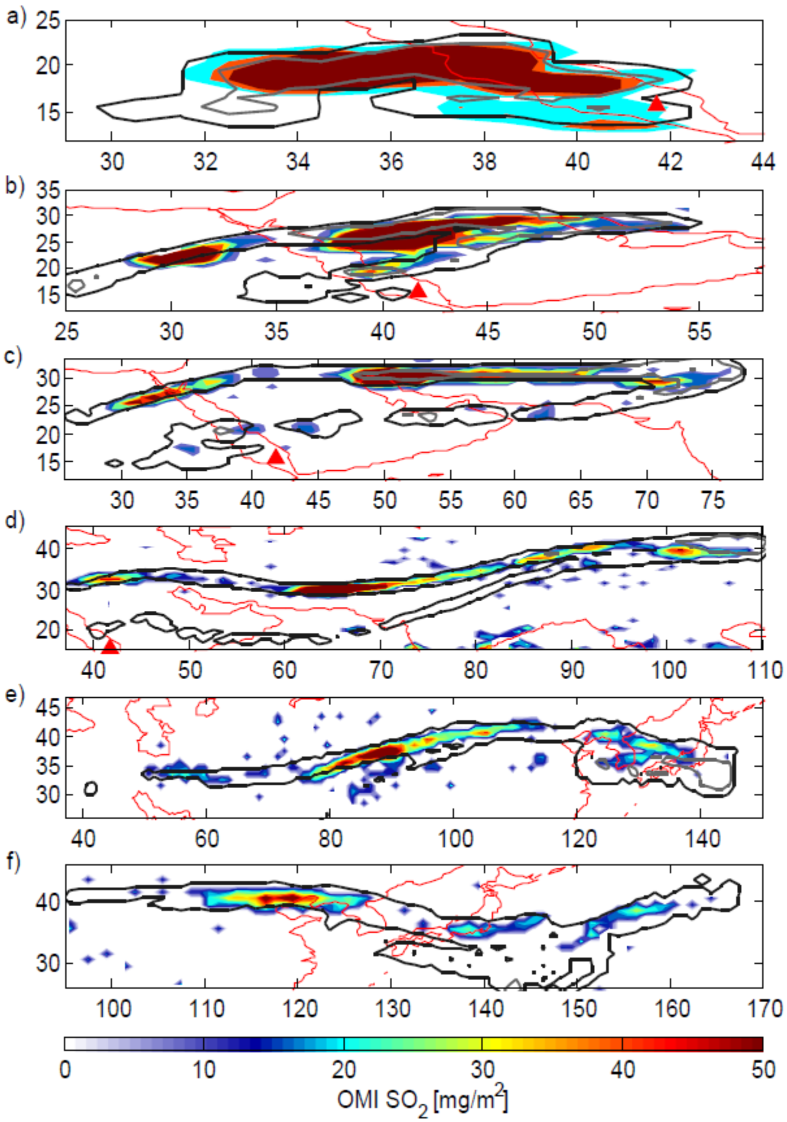
http://www.harmo.org/Conferences/Proceedings/_Kos/publishedSections/PPT/H14-333-PR.pdf

- Point source complicated release
- From regional to global scales
- Fwd runs

Volcano applications – forecast, source term estimation...

- Column source with a complex and variable vertical structure, source term (important the fine ash fraction) unknown.
- From regional to global scales
- Fwd runs





With author's permission- Eckhardt, S., Prata, A. J., Seibert, P., Stebel, K., and Stohl, A.: Estimation of the vertical profile of sulfur dioxide injection into the atmosphere by a volcanic eruption using satellite column measurements and inverse transport modeling, Atmos. Chem. Phys., 8, 3881-3897, doi:10.5194/acp-8-3881-2008, 2008.

FLEXPART: how it works in a nutshell

First:

- Correctly track the particles in a given velocity field.

Second:

- Model the Sub-grid scale (SGS) unresolved physical processes that affect the particles dispersion:
  - Boundary Layer Turbulence
  - Mesoscale Turbulence
  - Cumulus turbulent convection

Third:

- Modify particles properties based on locally acting processes, e.g. radioactive decay

Fourth:

- Count particles in a volume and extract concentration value

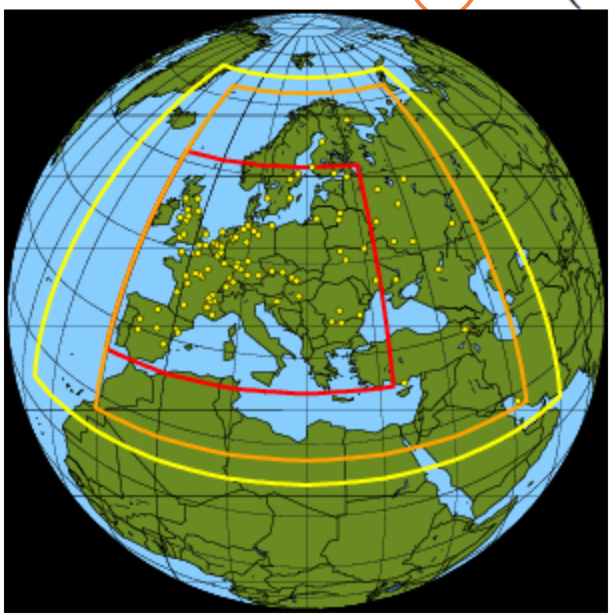
Vertical and horizontal coordinates:

- FLEXPART needs the vertical velocity in m s-1 because of the parametrized random velocities

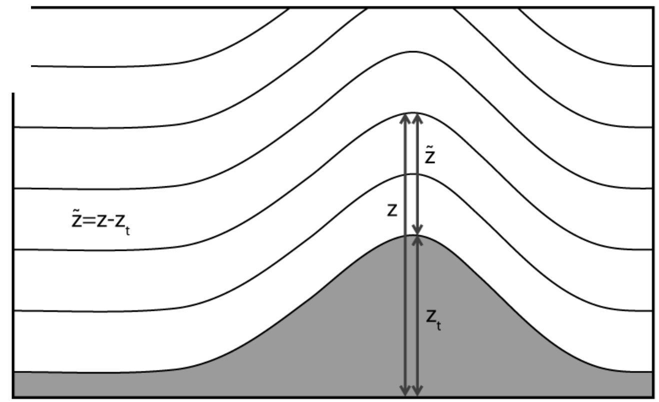
```
verttransform.f
```

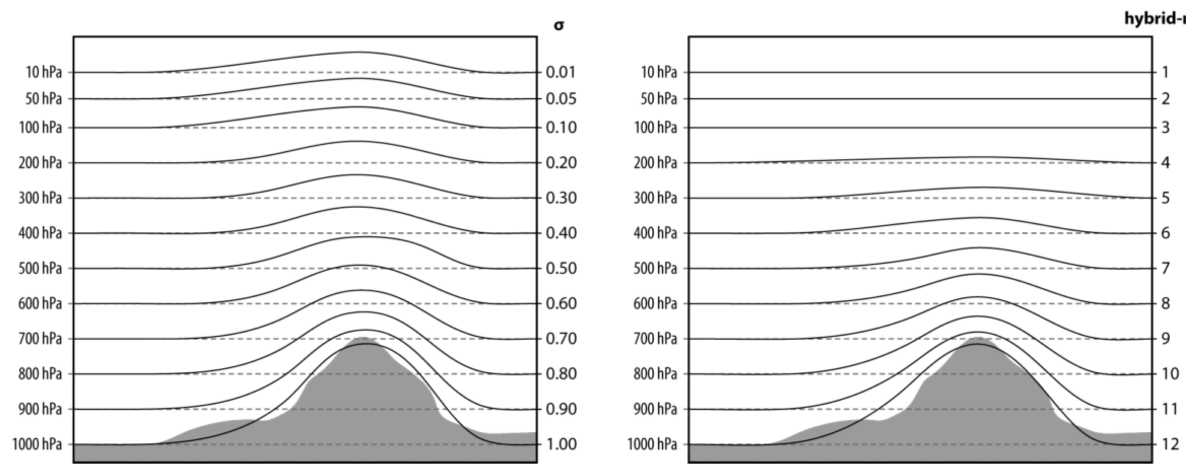
```
verttransform_gfs.f
```

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



flexrisk.boku.ac.at





Parameterization of the boundary layer (BL)

- Total surface (Reynold's) stress: the total vertical flux of horizontal momentum, measured near the surface

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

Absolute value and not the direction

verttransform.f

verttransform_gfs.f

GFS data has no stress nor flux data

- Friction velocity: scaling velocity for boundary layer parameterizations

$$u_* = \sqrt{\tau/\rho}$$

In calcpar.f → scalev.f

In calcpar_gfs.f → scalev.f

Parameterization of the boundary layer (BL)

- If (in readwind.f) surface stress and surface sensible heat fluxes are not available, then the pbl profile method (Berkowicz and Prahm 1982), based in M-O similarity theory using the lowest levels, is used.

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

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

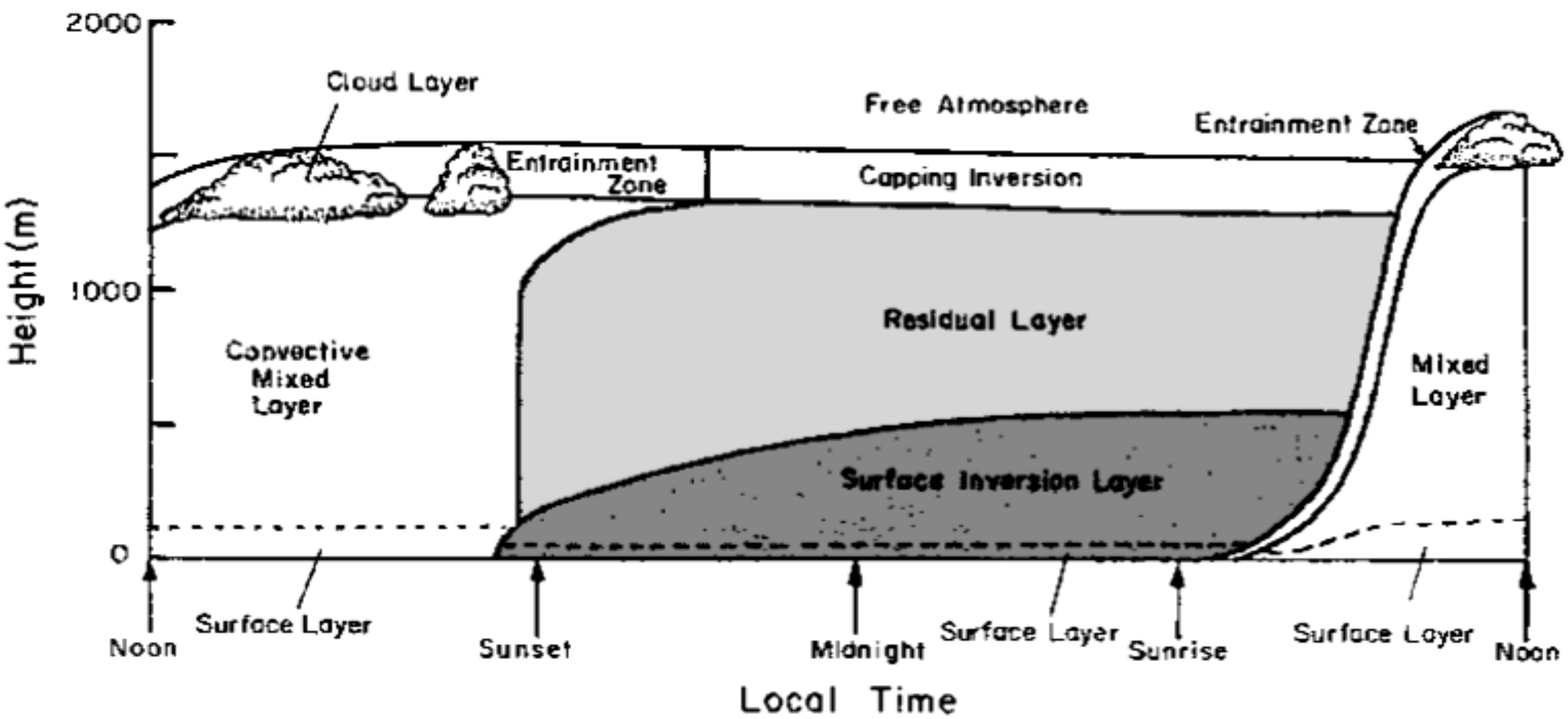
$$L = \frac{\overline{T} u_*^2}{g \kappa \Theta_*} ,$$

```
pbl_profile.f → includes
(max 10) iterations
```

- L is the the Obukhov length (m) and can be interpreted as the height where the absolute value of the shear production term and the buoyant term have the same absolute value.

Boundary layer height

- The Boundary Layer: the bottom layer of the troposphere that is in contact with (and affected by) the surface of the earth



From Sull (1988)

Boundary layer height

- Boundary layer height calculated using critical Richardson number (Vogelezang and Holtslag, 1996)

```
In calcpar.f → richardson.f
```

```
In calcpar_gfs.f → richardson_gfs.f
```

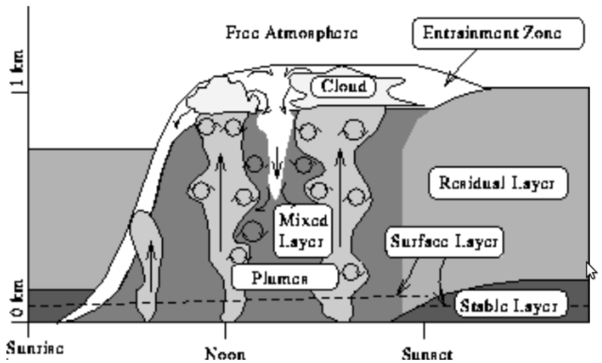
if $$Ri_l = \frac{(g/\Theta_{v1})(\Theta_{vl} - \Theta_{v1})(z_l - z_1)}{(u_l - u_1)^2 + (v_l - v_1)^2 + 100u_*^2}$$ > 0.25 → *l* is PBLH

- If convective (unstable) situations then one iteration is made (max number iterations 3):

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

Temp. excess from rising thermals

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

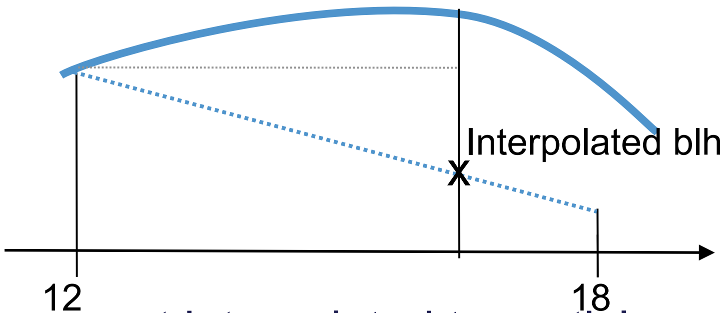


http://lidar.ssec.wisc.edu/papers/akp_thes/node6.htm

The guts of FLEXPART : physics

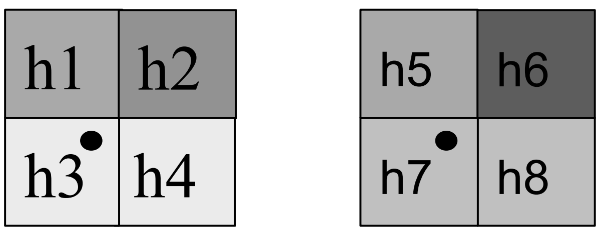
Boundary layer height

- Related to the temporal interpolation of the BL height:



Nothing can be done here → importance of having frecquent input data (3hours)

- Spatial heterogeneities – not interpolated to particle position, but the max in space and time



t1? t2?

→ Particle sees blh = h6

```
advance.f
```

- FLEXPART workaround – envelope mixing height

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

If unstable hmix + sigz

If stable then depending on Froude number

Remember to set an appropriate hmixmin in includepar or similar module!

The guts of FLEXPART

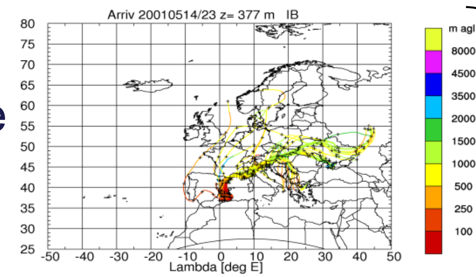
Transport and diffusion:

- FLEXPART calculates **trajectories** of *computational* particles (each particle carries a certain amount of mass or mixing ratio of species – *computational* -, as defined in the releases) *(change of mass described later)*

Integration (1st order, zero acceleration scheme)

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{v}[\boldsymbol{X}(t)] \longrightarrow \boldsymbol{X}(t+\Delta t) = \boldsymbol{X}(t) + \boldsymbol{v}(\boldsymbol{X},t)\Delta t$$

$\overline{\boldsymbol{v}}$ Grid scale wind $\rightarrow$ what simple trajectory models use (e.g. FLEXTRA)



$\boldsymbol{v}_t$ Turbulent wind fluctuations

$\boldsymbol{v}_m$ Mesoscale wind fluctuations (meandering)

$$\boldsymbol{v} = \overline{\boldsymbol{v}} + \boldsymbol{v}_t + \boldsymbol{v}_m$$

```
advance.f
```

$\boldsymbol{v}_t$ Turbulent wind fluctuations

- FLEXPART calculates the turbulent motions assuming a Markov process (for the velocity) based on the Langevin equation (Thomson 1987)

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

drift

diffusion

Wiener process - stochastic

We need to assume certain statistical propierties of the atmosphere to sample dW and solve the equation – gaussian turbulence (not valid for skewed turbulence in convective)

w(0) = w_r

σ^2_u= avg(u^2)

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

`advance.f`

Average speed non-zero

The guts of FLEXPART

$\boldsymbol{v}_t$ Turbulent wind fluctuations

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

drift diffusion Wiener process - stochastic

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

HOWEVER two corrections are needed:

1)Drift correction (McNider at al. 1988) – to prevent accumulation of particles in areas of low-turbulence

2)Density correction (Stohl andThomson 1999) – to account for the decrease of air density with height

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

```
advance.f
```

Sample only for the vertical velocity, horizontal no drift/density corrections

v_t Turbulent wind fluctuations

Rewritten following Wilson et al. (1983)

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

Less robust against increase of integration time step → both formulations are used, this one for shorter time-steps

This still needs to get discretizised

`advance.f`

Sample only for the vertical velocity, horizontal no drift/density corrections

v_t Turbulent wind fluctuations

`advance.f`

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

Direct solution of the differential equation: $u_{n+1} = u_n R(\Delta t) + \sigma_u [1 - R(\Delta t)^2] u''_{n+1}$

In FLEXPART:

$$r_w = \exp(-\Delta t/\tau_{L_w})$$

$$(\Delta t/\tau_{L_w}) \geq 0.5$$

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

stochastic

Taylor expansion

$$(\Delta t/\tau_{L_w}) < 0.5$$

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

Which time step?

advance.f

COMMAND

Two possibilities:

1. Fixed time step (=synchronisation time) without adaptation to the lagrangian timescales → FASTER but LESS ACCURATE (COMMAND ctl <0) may be useful for long range applications if computational resources are limited

2. Time step adapting to the vertical lagrangian time scales → turbulence is described in a more accurate/realistic way (COMMAND ctl >0, ifine) and thus needed for any BL studies – the time step for the horizontal will be defined as:

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

The time step in the vertical is splitted: $\Delta t_w = \Delta t_i / \mathtt{ifine}$

σ_{v_i} τ_{L_i}

Vertical profiles of the turbulent quantities inside the ABL

Depend on the state of the turbulent atmosphere. Following Hanna 1982.

`hanna.f`

`hanna1.f`

`hanna_short.f`

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

1. Unstable

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

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

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

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

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

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

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

$z/h > 0.1$

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

The guts of FLEXPART

$$\sigma_{v_i} \quad \tau_{L_i}$$

`hanna.f`

`hanna1.f`

`hanna_short.f`

2. Neutral

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

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

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

3. Stable

$$\frac{\sigma_u}{u_*} = 2.0 \left(1 - \frac{z}{h}\right) \quad \frac{\sigma_v}{u_*} = \frac{\sigma_w}{u_*} = 1.3 \left(1 - \frac{z}{h}\right)$$

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

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

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

`advance.f`

What about above the ABL?

In the free atmosphere turbulence is in small places coming from gravity waves, arround jet streams... it is not yet parameterized in detail.

FLEXPART treats the stratosphere with a constant vertical diffusivity (Legras et al. 2003)

$$D_z = 0.1 \text{ m}^2\text{s}^{-1}$$

And a constant horizontal diffusivity in the free troposphere

$$D_h = 50 \text{ m}^2\text{s}^{-1}$$

with an intermediate zone from free-troposphere to stratosphere. Turbulent velocity scales are then calculated by

$$\sigma_{v_i} = \sqrt{D_i/dt}$$

The guts of FLEXPART

v_m Mesoscale wind fluctuations

- Fluctuations neither resolved by ECMWF nor by the turbulence parameterization, created by mountain waves, pulsating drainage flows, wake vortices, ...
- FLEXPART solves another Langevin equation and assumes that the variance of the wind a the grid scale provides sub-grid scale information (Maryon 1998) → this acts as a source of dispersion (*without really physically representing the phenomena*)

```
r=exp(-2.*float(abs(lsynctime))/float(lwindinterv))
rs=sqrt(1.-r**2)
```

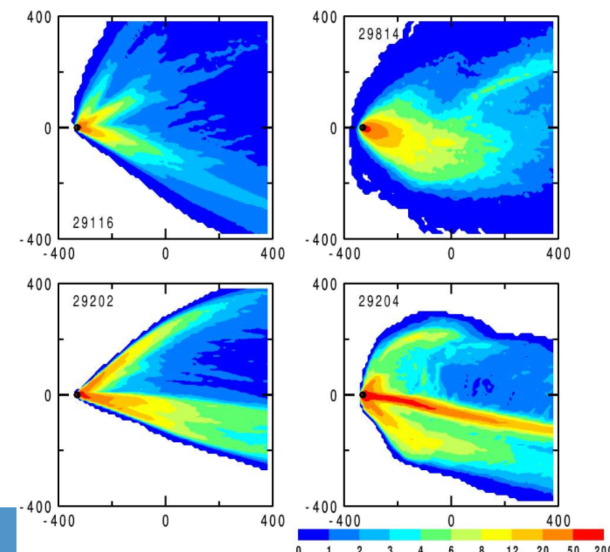
→ Correlation, with a correlation time scale half the wind interval

```
usigold=r*usigold+rs*rannumb(nrand)*usig*turbmesoscale
vsigold=r*vsigold+rs*rannumb(nrand+1)*vsig*turbmesoscale
wsigold=r*wsigold+rs*rannumb(nrand+2)*wsig*turbmesoscale
```

→ set in includepar

```
zt=zt+wsigold*float(lsynctime) ...
```

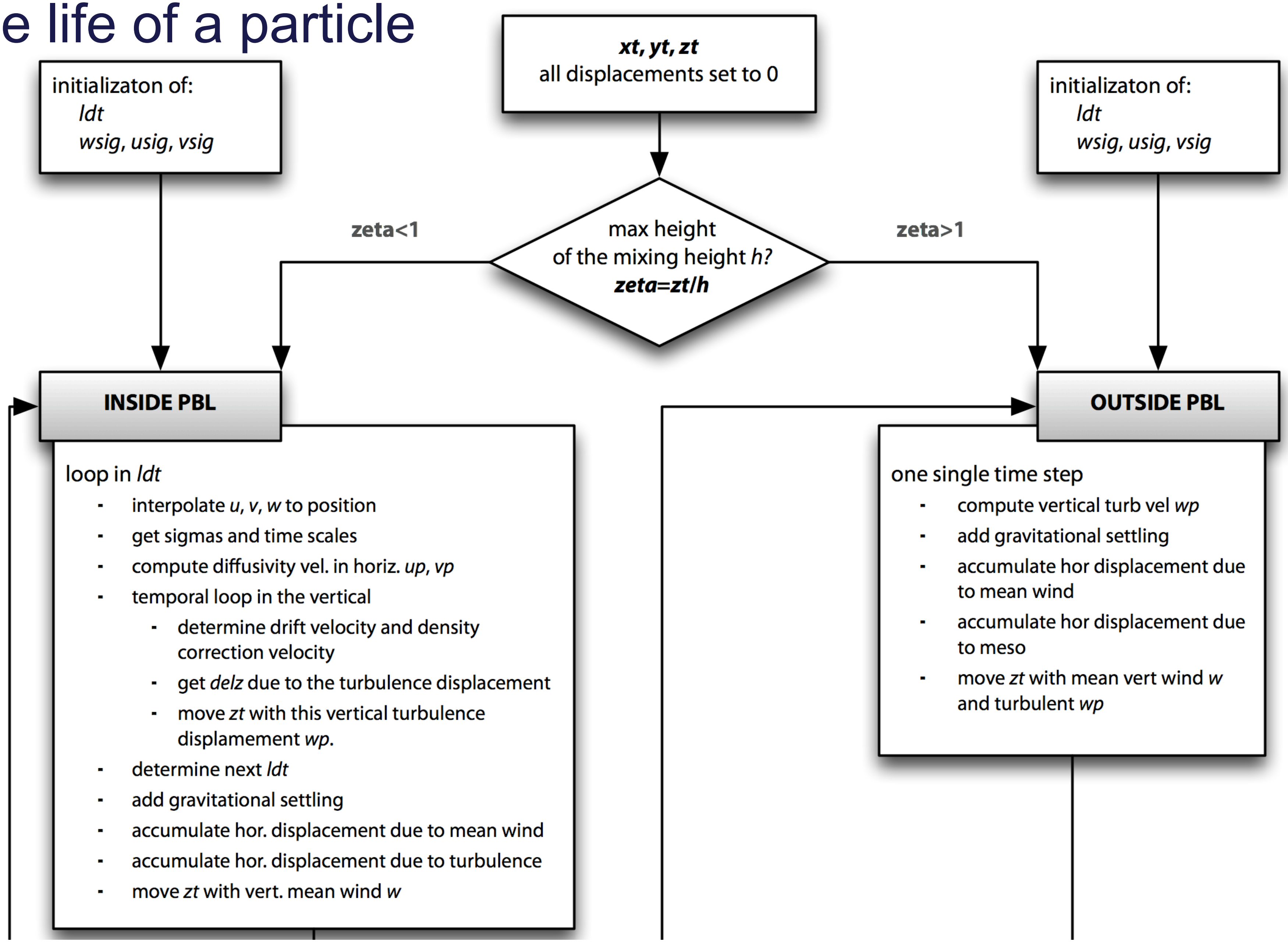
`advance.f`



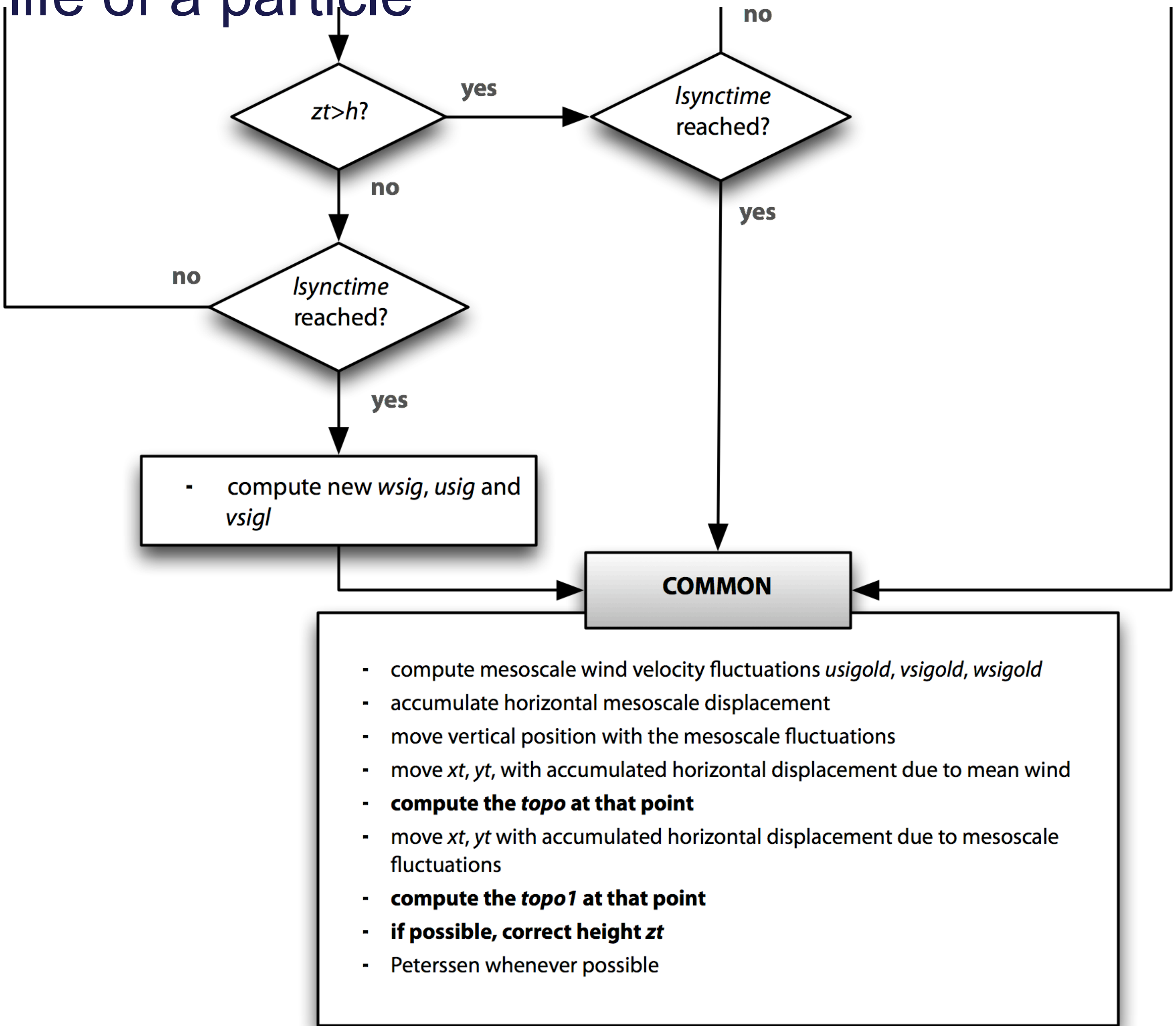
Vickers, D., Mahrt, L. and Belusic, D. 2008. Particle simulations of dispersion using observed meandering and turbulence. Acta Geophys. 56 (with permission of authors)

The guts of FLEXPART

The life of a particle

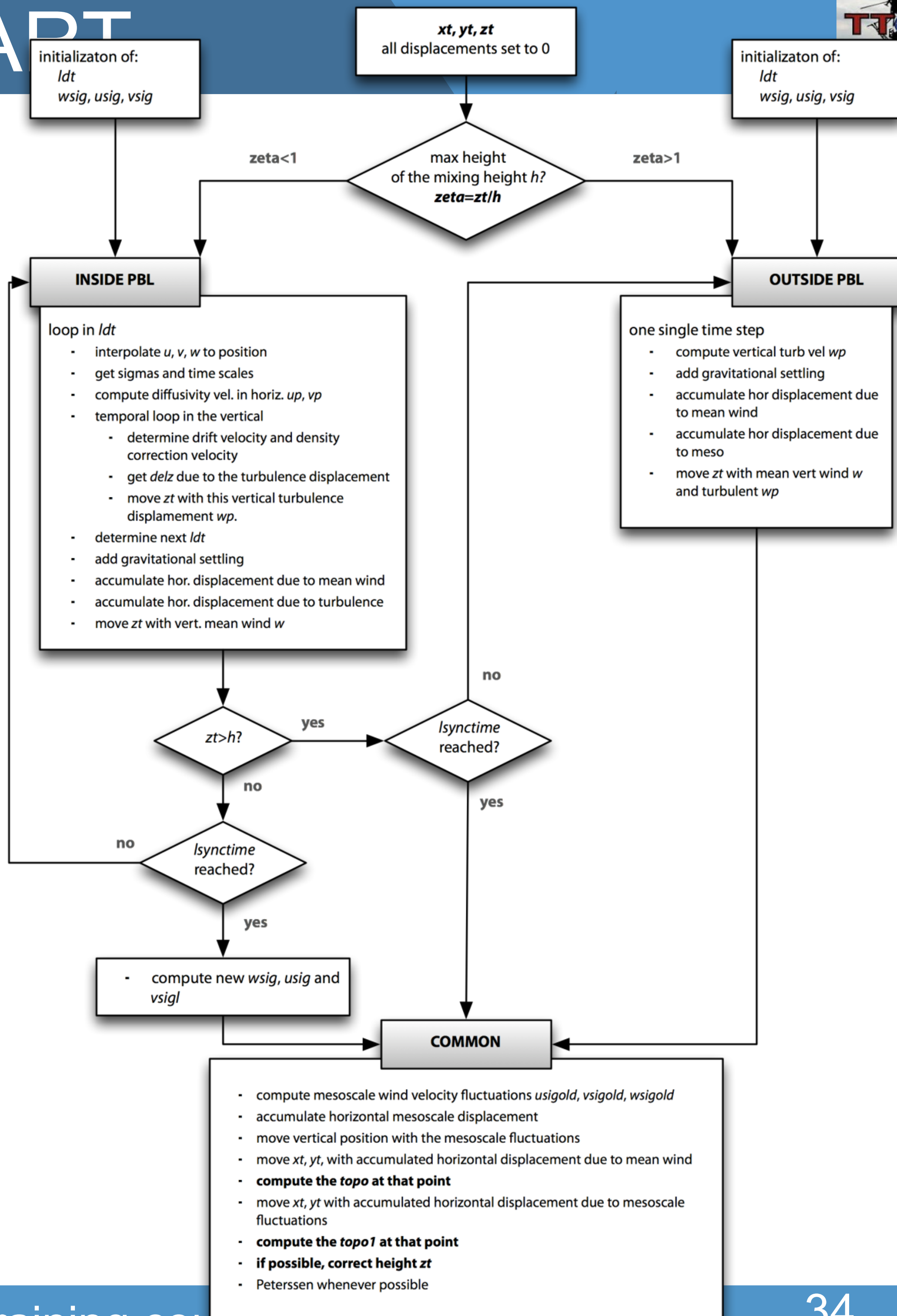


The life of a particle



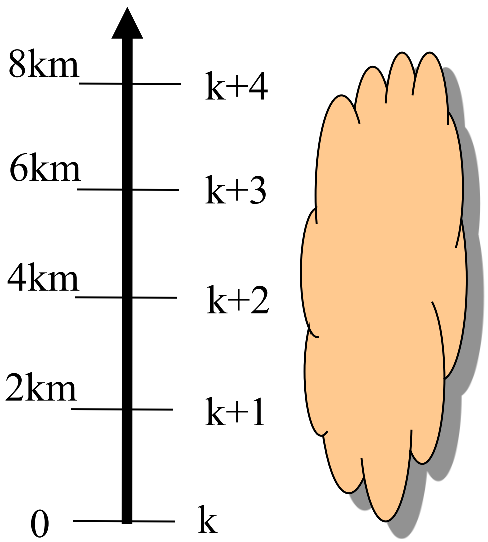
The guts of FLEXPART

The life of a particle

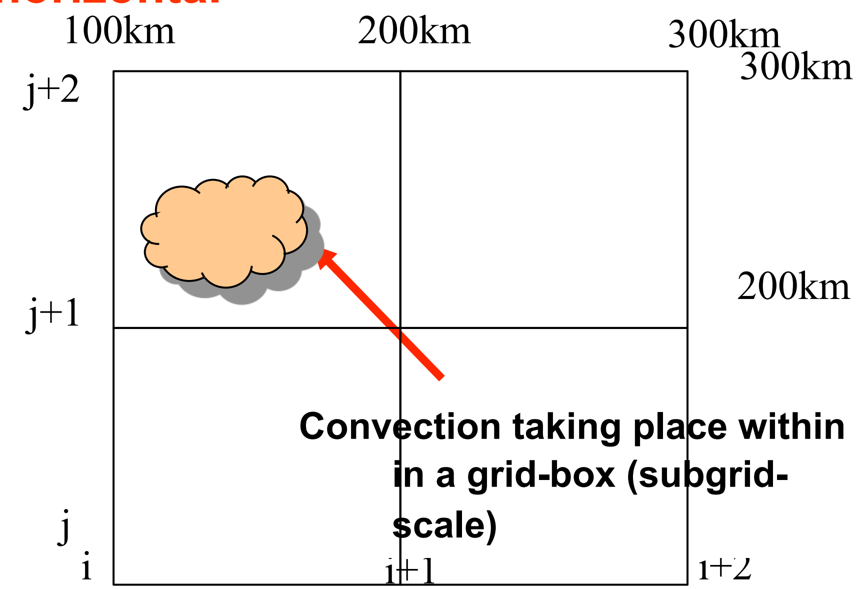


Convection in models

convection is grid-scale in the vertical



but subgrid-scale in the horizontal



Meteorological parameters (temperature, humidity, wind etc.) given at horizonital model grid points (i,j), (i,j+1), (i,j+2) etc., but there is no information inbetween

→ **convection has to be parameterized:** convection takes place under certain large-scale conditions

Additional feature – Moist convection

COMMAND

timemanager.f

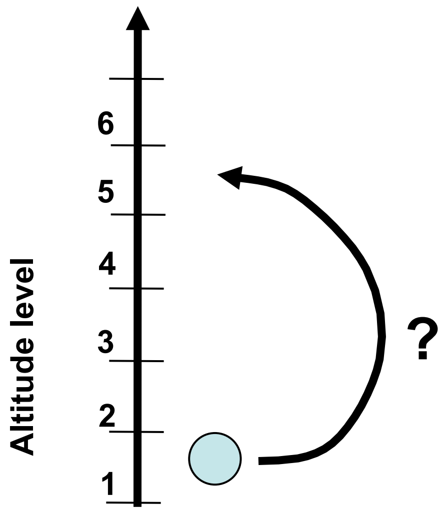
convmix.f, calcmatrix.d

convect43c.f

convmix_gfs.f, calcmatrix_gfs.f

Transport due to updrafts in convective clouds (grid-scale in the vertical, sub-grid in the horizontal) in FLEXPART the Emanuel and Zivkovic—Rothman (1999) scheme is used to redistribute the particles in the vertical column – large increase in the computational time!

Every time called new meteorological data is read in → reads all vertical meteorological information (model levels) – grid scale temperature and humidity-, the convection scheme is called per grid column and distributes the particles then adding a compensating vertical velocity to the remaining ones to represent (without numerical diffusion) the compensating subsidence of the updrafts necessary to know how the particles shall be redistributed vertically, i.e. destination level of each particle must be known



The particles carry mass fractions in the model
→ mass fraction M displaced from level i to level j must be known

Matrix M(i,j)
i: source level
j: destination level

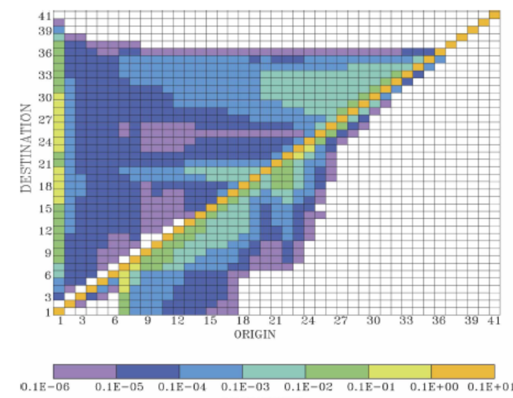


FIG. 1. Example of a mean convective redistribution matrix along 10° latitude for October 1983 calculated from the EZ99 scheme. The colors indicate the probability for a particle to be displaced from its origin to its destination level. White colors indicate probabilities below 10^{-6}. The sum of each column is 1. Origin and destination are given in numbers of model levels. For the height of these model levels, see Table 1.

FLEXPART interface:

construct a matrix of condtional probabilities P(i,j) that a particle is displaced from level i to level j given that it is in level i

P(i,j)=M(i,j)|t/m(i)

Assume that all convective fluxes (the matrix) are balanced by compensating subsidence (a downward velocity) in the environment; the subsidence acts on those particles that are not displaced by the matrix

Forster, C., A. Stohl, and P. Seibert, 2007: Parameterization of convective transport in a Lagrangian particle dispersion model and its evaluation, J. Appl. Meteorology, Vol. 46, No. 4, 403-422.

For a wonderful review of deposition processes (specially for radionuclides) check Sportisse 1007

In FLEXPART **dry deposition** is described by a deposition velocity

$$v_d(z) = -F_C/C(z)$$

Two possibilities:

1. A constant deposition velocity set in SPECIES_nnn

2. More sophisticated calculation coming from the chemical and physical proprieties of the substance also defined in SPECIES_nnn

```
SO4-aero            Tracer name
-999.9              Species half life
 5.0E-06            Wet deposition - A
0.62                Wet deposition - B
-9.9                Dry deposition (gases) - D
 1.0E-09            Dry deposition (gases) - Henrys const.
                    Dry deposition (gases) - f0 (reactivity)
 2.0E03             Dry deposition (particles) - rho
4.0E-7              Dry deposition (particles) - dquer
3.0E-1              Dry deposition (particles) - dsig
-9.99               Alternative: dry deposition velocity
 -9.99              molweight
-9.9E-09            OH Reaction rate at 25 deg, [cm^3/sec]
-9                  number of associated specias (neg. none)
-99.99              KOA - organic matter air partitioning
```

SPECIES_nnn

Dry deposition for gases

Calculated with the resistance method (Wesely and Hicks, 1977, 2000) – analogy to electrical resistance

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

Aerodynamic resistance between z and a the top of the vegetation canopy

Quasilaminar sublayer resistance

Bulk surface resistance

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

Profile function

Ability of the eddies to bring the material close to the surface, except for large particles, dependent on the flow

getvdep.f

raerod.f

getrb.f / getrc

Dry deposition for gases

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

Following Erisman et al. (1994)

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

Schmidt number – viscuous diffusion rate/molecular diffusion rate

Prandtl number – viscuous diffusion rate/thermal diffusion rate

Resistance to transfer across the final layer, dependent on physical form of the gas

Following Wesely (1989)

$$\frac{1}{r_c} = \frac{1}{r_s + r_m} + \frac{1}{r_{lu}} + \frac{1}{r_{dc} + r_{cl}} + \frac{1}{r_{ac} + r_{gs}}$$

Dependant on the vegetation, canopy type, leave resistance, soil resistance --- > surfdepo.t /surfdata.t

Dependent on the affinity of the surface to the gas

getvdep.f

raerod.f

getrb.f / getrc

Removal Processes: dry deposition

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 grow shrubs	0.05
12	snow and ice	0.001
13	rainforest	1.

surfdepo.f

surfdata.f

Removal Processes: dry deposition

```
================================================================================
INPUT RESISTANCES (s/m) FOR THE COMPUTATION OF SURFACE RESISTANCES TO
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.	

surfdepo.f

surfdata.f

Dry deposition for particles

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

Gravitational settling – ONLY is the particles carry one SINGLE species!

Following Erisman et al. (1994) , gravitational settling velocity is:

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

Dependent on the pàrticle diameter

Dynamic viscosity of air

SO4-aero	Tracer name
-999.9	Species half life
5.0E-06	Wet deposition - A
0.62	Wet deposition - B
-9.9	Dry deposition (gases) - D
1.0E-09	Dry deposition (gases) - Henrys const.
	Dry deposition (gases) - f0 (reactivity)
2.0E03	Dry deposition (particles) - rho
4.0E-7	Dry deposition (particles) - dquer
3.0E-1	Dry deposition (particles) - dsig

advance.f

part0.f

partdep.f

Removal Processes: dry deposition

Dry deposition for particles

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

Ra calculated as for gases.

Rb is the same as gases but with an impaction factor.

advance.f

part0.f

getrb.f

partdep.f

New in FLEXPART 8.2 – temperature dependence on the dynamic viscosity based on Naestlund and Thaning, 1991) –

get_settling.f in advance.f

Removal Processes: dry deposition

Loss of mass due to dry deposition

Gravitational settling is applied throughout all the particle trajectory but it does not cause a loss of mass.

When the particle is at *zt* < 2 href (*href = 15 m* set in includepar) the there is loss of mass due to dry deposition

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

Removal Processes: dry deposition

Dry deposition in the code

```
--> timemanager --> wetdepo --> interpol_rain
                               --> interpol_rain_nests
                               --> wetdepokernel
                               --> wetdepokernel_nest
                --> ...
                --> getfields -->  ------->        --> calcpar --> scalev --> ew
                                                                --> ...
                                                                --> getvdep --> caldate
                                                                                --> getrb
                                                                                --> raerod --> psih
                                                                                --> getrc
                                                                                --> partdep
                -->...
                --> …
                --> advance -->,,,
                               --> ...interpol_all
                               --> ...
                               → get_settling
                --> calcfluxes
                --> drydepokernel
                --> drydepokernel_nest
                --> ...
```

Removal Processes: wet deposition

For a wonderful review of deposition processes (specially for radionuclides) check Sportisse 1007

In FLEXPART (v 8 →) **wet deposition** is separated into:

1. In-cloud scavenging (also called rainout) – very efficient process
2. Below-cloud scavenging (also called washout)
3. No specific treatment of snow scavenging

SO4-aero	Tracer name
-999.9	Species half life
5.0E-06	Wet deposition - A
0.62	Wet deposition - B
-9.9	Dry deposition (gases) - D
1.0E-09	Dry deposition (gases) - Henrys const.
	Dry deposition (gases) - f0 (reactivity)
2.0E03	Dry deposition (particles) - rho
4.0E-7	Dry deposition (particles) - dquer
3.0E-1	Dry deposition (particles) - dsig
-9.99	Alternative: dry deposition velocity
-9.99	molweight
-9.9E-09	OH Reaction rate at 25 deg, [cm^3/sec]
-9	number of associated specias (neg. none)
-99.99	KOA - organic matter air partitioning

SPECIES_nnn

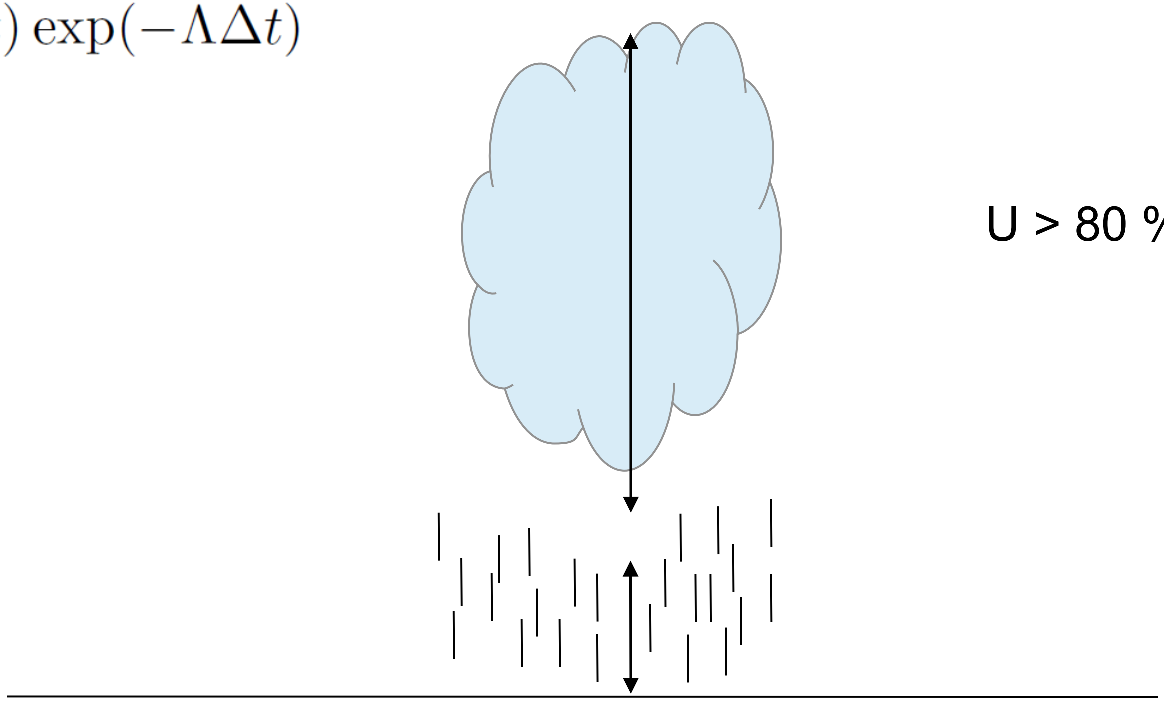
Removal Processes: wet deposition

In FLEXPART (v 8 →) **wet deposition** is separated into:

1. In-cloud scavenging (also called rainout) – very efficient process
2. Below-cloud scavenging (also called washout)
3. No differences with snow scavenging processes in FLEXPART

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

Change of mass



U > 80 % → CLOUD

wetdepo.f

In cloud (nucleation) scavenging

It follows Hertel et al. 1995

Cloud liquid water content

$$\Lambda = \frac{S_i I}{H_i}$$

particles

$$S_i = 0.9/cl$$

$$cl = 2 \times 10^{-7} \cdot I^{0.36}$$

gases

$$S_i = 1/cl_{\text{eff}}$$

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

Very difficult to parameterize due to the involvement of cloud microphysics (nucleation), solubility, collection efficiency, size and distribution dependance...

wetdepo.f

Removal Processes: wet deposition

Below cloud (impaction) scavenging

It follows McMahon 1979. Scavenging here is parameterized (justification of the parameterization in Sportisse 2007 amongst others) according to $\Lambda = AI^B$

A and B come from experimental studies. They may be set by the user in SPECIES_nnn → mind that usually and in previous versions of FLEPXART the coefficients A and B consider in- and below-cloud scavenging as a single bulk process. If the new scheme is used one needs to be certain to adapt the A and B so as not to overestimate below cloud scavenging

wetdepo.f

Sub-grid variability

Rationale: it will not rain with a constant precip rate for the whole period (met file) and not all over the grid cell

Hertel et al. 1995 – defines the area affected by precip (convective, c, and large scale, l) as:

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

CC is the total cloud cover (0 or 1). And:

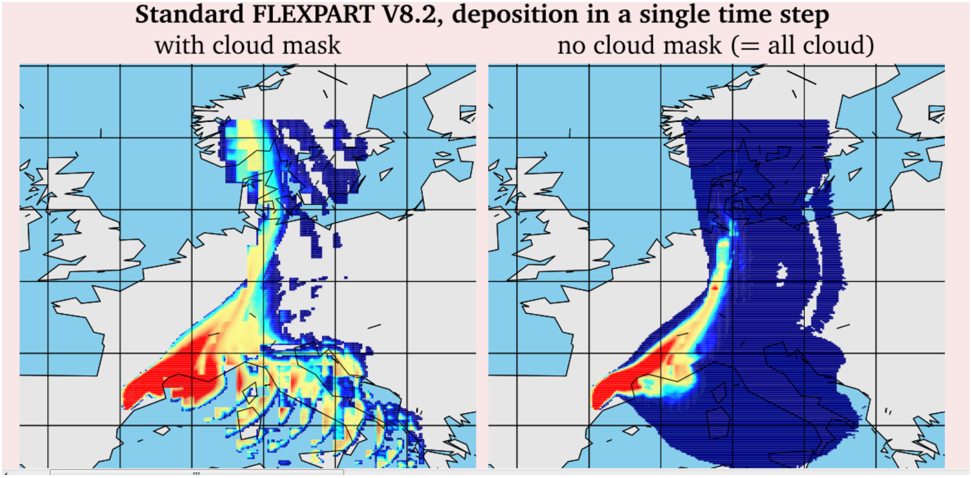
	I_l and I_c				
Factor	$I \leq 1$	$1 < I \leq 3$	$3 < I \leq 8$	$8 < I \leq 20$	$20 < I$
fr_l	0.50	0.65	0.80	0.90	0.95
fr_c	0.40	0.55	0.70	0.80	0.90

HOWEVER! Hertel did this for grid sizes larger than 100 km and times between met fields of 6 hours. For the most recent ECMWF fields, for instance, one should reconsider this fractions.

wetdepo.f

To consider and problems

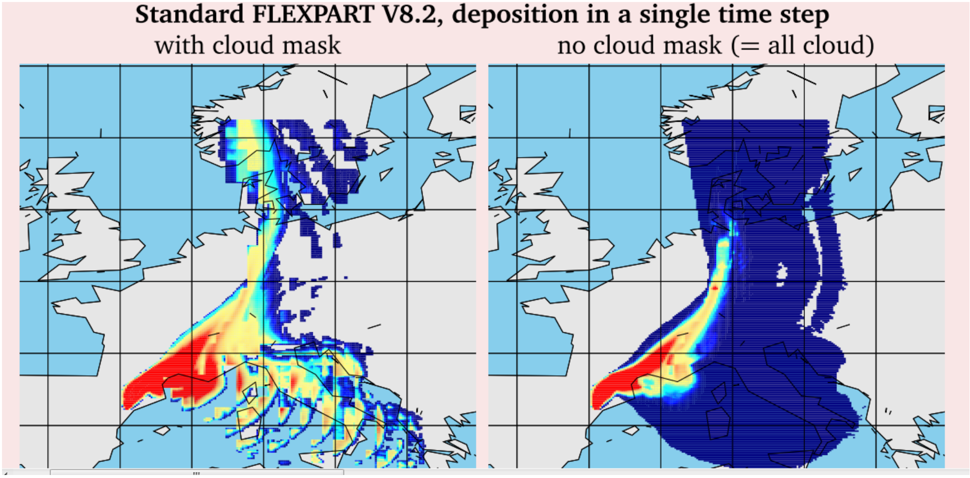
1. Clouds are not interpolated and this may at times lead to checkerboard patterns and bands. This is more visible when the resolution of the input meteorological fields is much coarser than the output grid.
2. The precip, in the ECMWF extraction routines is deaccumulated and thus there is a spatial and temporal smoothing.
3. Sub-grid scale variability may not be good for high resolution met input data
4. Please do mind the definition of A and B

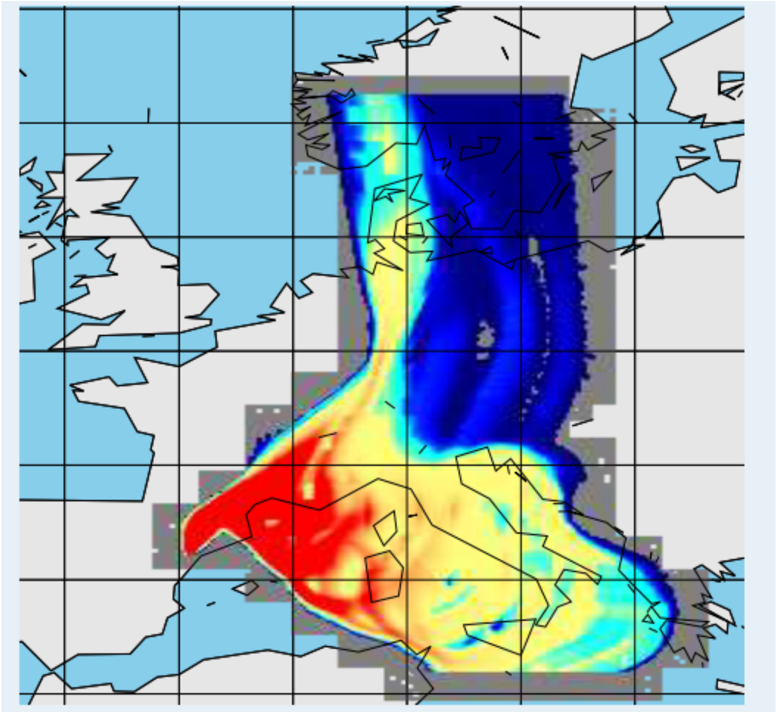


Removal Processes: wet deposition

To consider and problems

P. Seibert implemented a (non-prefect) bug fix for 1 and partially 2. This bug fix interpolates the clouds to the particle position, interpolates better the rain and includes a hardcode use of the old scheme where incloud and below-cloud are used in a bulk way when there is rain but no cloud is diagnosed. Careful! hard-coded.





Removal Processes: wet deposition

Wet deposition in the code

--> timemanager --> wetdepo --> interpol_rain
--> interpol_rain_nests
--> wetdepokernel
--> wetdepokernel_nest
--> ...
--> getfields --> -------> --> calcpar --> scalev --> ew
--> ...
--> getvdep --> caldate
--> getrb
--> raerod --> psih
--> getrc
--> partdep
-->...
--> …
--> advance -->,,,
--> ...interpol_all
--> ...
→ get_settling
--> calcfluxes
--> drydepokernel
--> drydepokernel_nest
--> ...

Removal Processes: others

Radioactive decay

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

SPECIES_nnn

Although radioactive decay can be directly used on runtime, it is easy and computationally much more efficient to introduce this as a simple post-processing even by having pre-calculated exponentials in look-up tables.

Additional features … many

Plume trajectories

plumetraj.f

clusering.f

Particles are clustered into a number of trajectories (ncluster in includepar) via k-means clustering using the **horizontal** distance as the distance function. The centroids are given by x,y,z but z is not used in the clustering algorithm

Warm start

partoutput.f

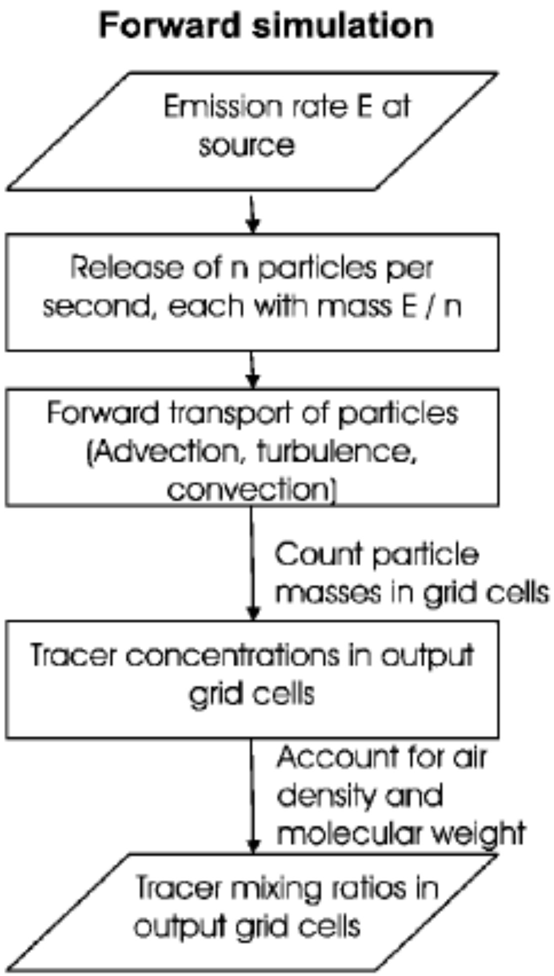
readpartpositions.f

COMMAND

Particle positions may be dumped (with additional information) and used for a warm start.

Forward and backward runs

Forward: the particles are released and followed downwind → one may get the output in concentrations or mixing ratios

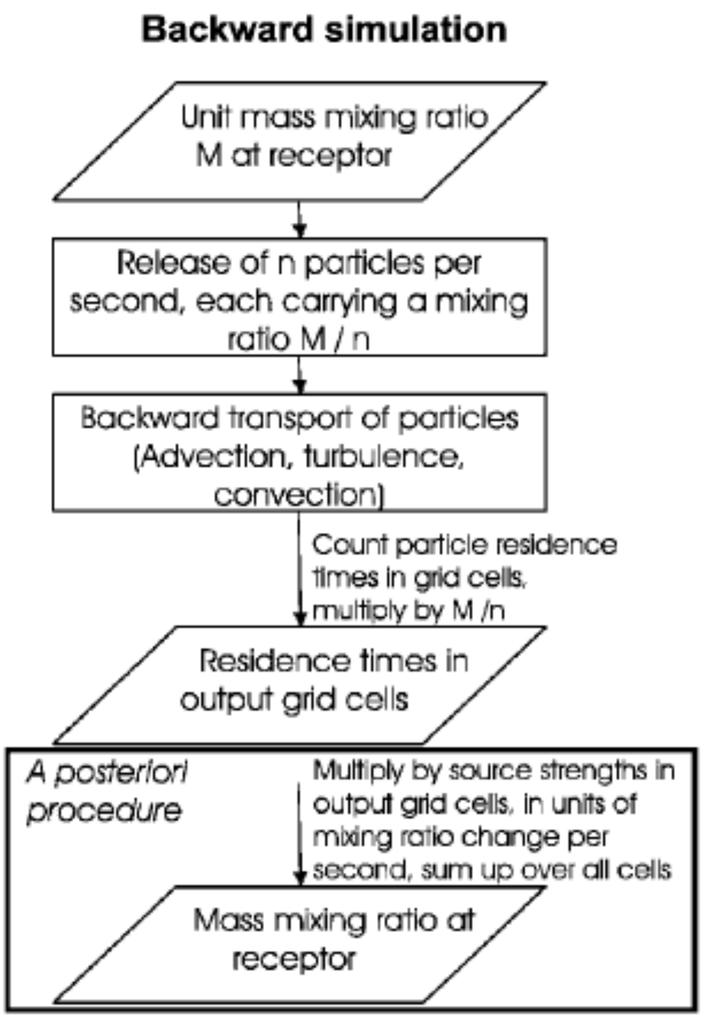


COMMAND

Forward and backward runs

Backward: the particles are released from a receptor (measurement site) and a 4-D (space + time) sensitivity function to the emision is given. This is especially useful to calculate source-receptor sensitivities when the number of receptors < number of potential sources. Everything is normalized with the mass released so any value other than 0 is correct. Dry and wet deposition correct the sensitivities. (Seibert 2001, Stohl et al. 2003, Seibert and Frank 2004)

Deposition fields are not output



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

COMMAND

Forward and backward runs

Units:

Output scaled with the emission

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

Internal 10E12 factor!!!

The output of a backward run can be understood as a gridded "residence time" of the tracked air mass, i.e., the output informs where and how long an air mass has resided. Units, thought, depend on setting

Forward and backward runs: units

Let's try to understand the concept and application

Land-sea mask (1/0)
x
^{222}Rn inventory ($Bqm^{-2}h^{-1}$)

t_1

0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

FLEXPART output
SRS (h^{-1})

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{t1} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

Forward and backward runs: units

Let's try to understand the concept and application

Land-sea mask (1/0)
x
^{222}Rn inventory ($Bq\,m^{-2}h^{-1}$)

FLEXPART output
SRS (h^{-1})

t_1

0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{t1} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

t_2

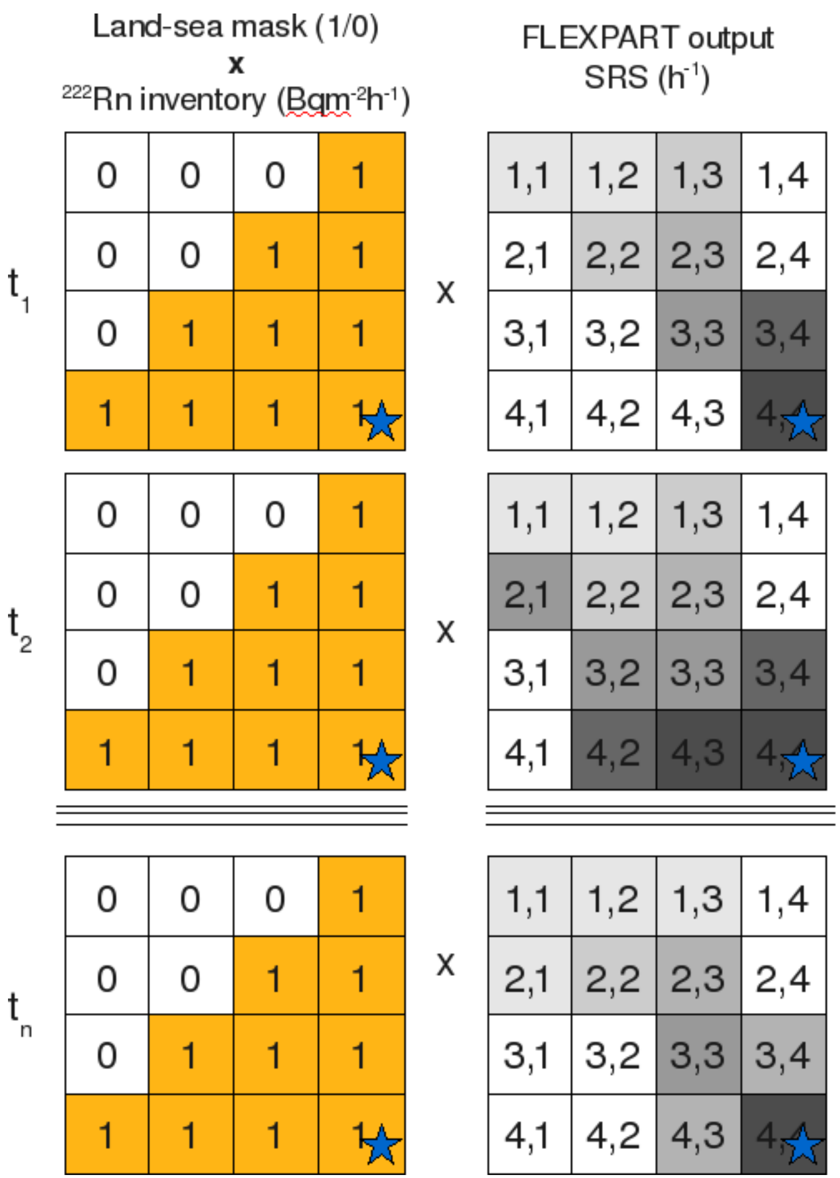
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{t2} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

Let's try to understand the concept and application



$$C_{t1} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

$$C_{t2} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

$$C_{tn} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

Let's try to understand the concept and application

Land-sea mask (1/0) **x** ^{222}Rn inventory ($Bqm^{-2}h^{-1}$)

FLEXPART output SRS (h^{-1})

t_1

0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{t1} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

t_2

0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{t2} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$

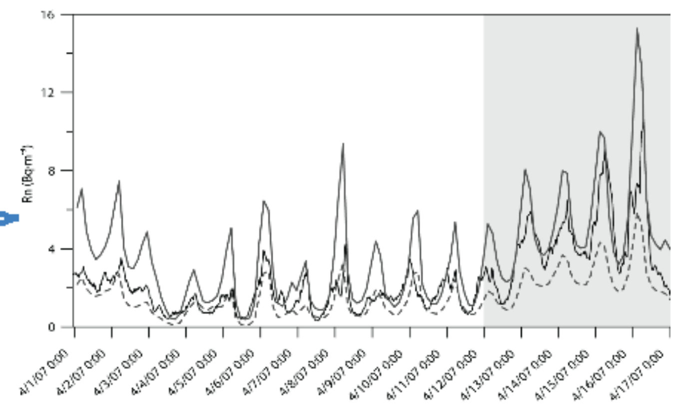
t_n

0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

x

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

$$C_{tn} = 50 \sum_i \sum_j LSM_{ij} \times SRS_{ij}$$



Forward and backward runs: units

Let's try to understand the concept and application: source for Birkenes

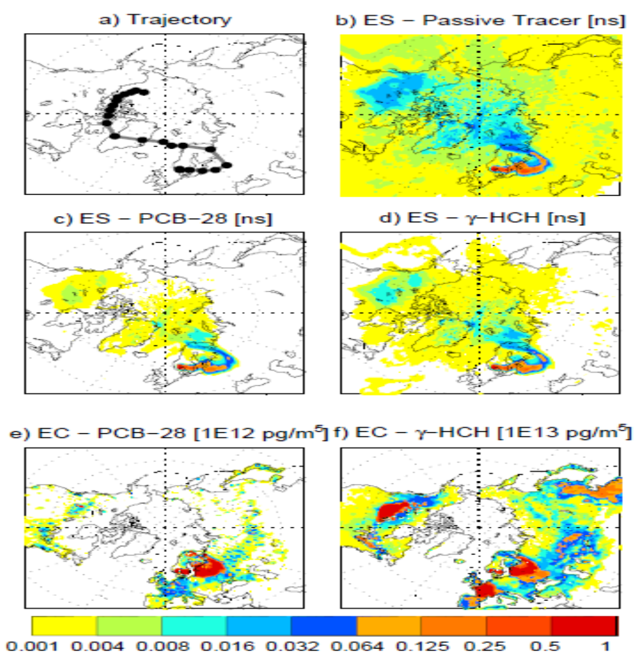


Fig. 4. Model results for the case study of a measurement sample taken at Birkenes from 7–8 June 2007: **(a)** Mean retroplume trajectory, with the grey line indicating the retroplume mean trajectory and black dots marking daily intervals. **(b)** emission sensitivities (ES) in the footprint layer for a passive tracer experiencing no removal processes; **(c)** same as (b) but for simulated PCB-28, and **(d)** ES for simulated γ-HCH. **(e)** Emission contributions (EC) for simulated PCB-28 and **(f)** for simulated γ-HCH.

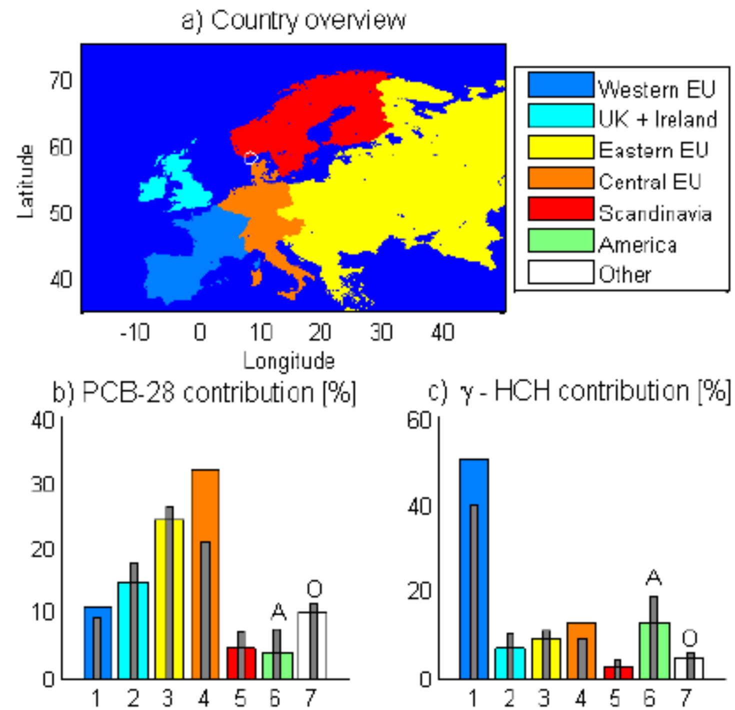


Fig. 12. Panel **(a)** shows the different regions for which relative source contributions for the Birkenes station (marked with a white circle) have been calculated (1, Western Europe ; 2, UK and Ireland; 3, Eastern Europe; 4, Central Europe; 5, Scandinavia; 6, America (denoted with A); 7, other countries (denoted with O)) Panels **(b)** and **(c)** show the regional source contributions (in % of the total) for simulated PCB-28 and γ-HCH, respectively, with the colors indicating the contributions from the different regions as colored in panel (a). The grey bars show the contributions when they are weighted by the ratio of the measured/modeled concentrations.

Emission contribution for the two different tracer

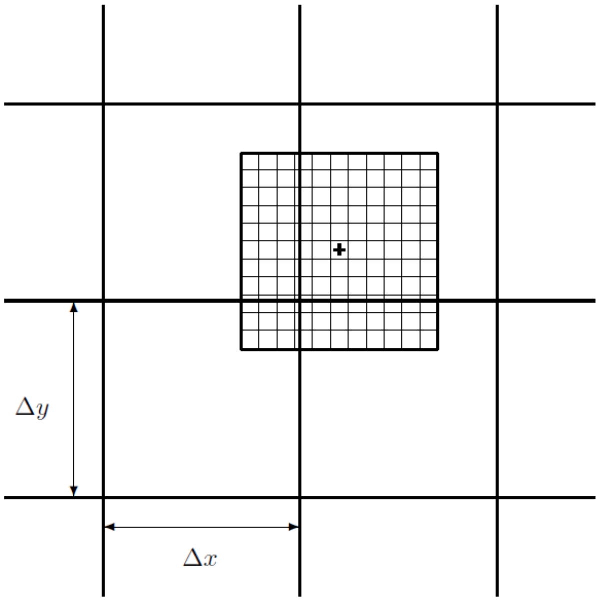
FLEXPART output : kernels

Receptor parabolic kernel

The mass of a particle is distributed into the adjacent grid cells using a parabole

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

Concentration /residence time uniform kernel



conccalc.f

wetdepokernel.f

drydepokernel.f

The mass of a particle is distributed into the adjacent grid cells.

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

It is not used the 3 hours after the particle release to avoid smoothing → CAREFUL! The default flexpart version does use the kernel for dry and wet depo regardless the time. Ask (me, others) for the corrected code

To know:

1. Concentrations are given as time averages whereas deposition is accumulated within the interval time and along the run

2. Radioactive decay IS applied to the deposited substance

Condensed sparse matrix output

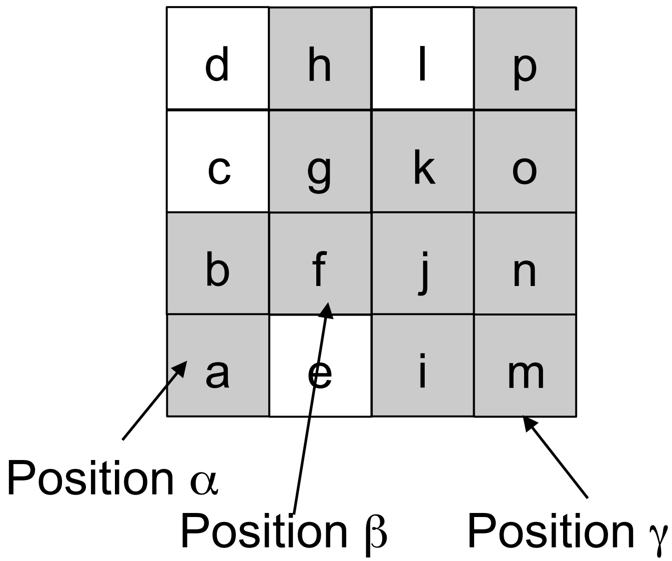
pseudocode:

```
For species
  For maxpoint
     For ageclasses
        (wetdepo)
        sp_count_i=0
        sp_count_r=0
        sp_fact = -1
        sp_zer = .true.
             For y direction
                For  x direction
                   If wetgrid > 0 then
                        If sp_zer = .true. Then → first non 0 value
                             sp_count_i = spcount _i + 1  (total number of non-zero segments)
                             sparse_dump_i  →  index of non-zero segment
                             sp_zer= . False.
                             sp_fact = sp_fact*-1
                         sp_count_r = sp_count +1
                         sparse_dump = value of wet deposition (FACTORS! 1E12)
                     Else no wetgrid and sp_zer = .true

                            write(unitoutgrid) sp_count_i
                   write(unitoutgrid) (sparse_dump_i(i),i=1,sp_count_i)
                   write(unitoutgrid) sp_count_r
                   write(unitoutgrid) (sparse_dump_r(i),i=1,sp_count_r)
```

Condensed sparse matrix output

Naïve example:



3

α β γ

12

a b -f -g -h -i -j -k m n o p