System for Integrated modeLling of Atmospheric composition (SILAM)

Purpose

The SILAM model is a chemical transport model, suitable for global, regional- (several thousand kilometres) and meso-scale (50-200 Km) simulations developed in the Finnish Meteorological Institute (FMI) (Sofiev *et al.,* 2006, 2008). Historically has been designed for radioactive emergency-preparedness simulations, and evolved to provide answers about the fate of the main atmospheric compounds concerning air quality purposes: ozone, particulate matter, sulphur oxides, nitrogen oxides, and carbon oxides. Additionally, the model has been developed to include pollen/allergen and fire-related emission production and dispersion.

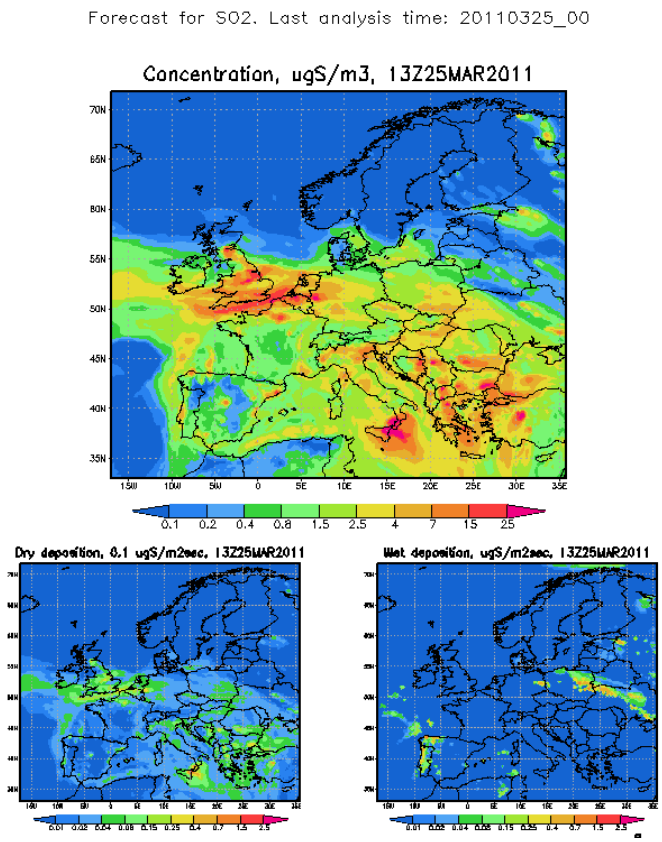

Figure 1. Example of a forecast for SO_2 surface concentrations (top), dry- (left-down) and wet-depositions (right-down) forecast: 25th of March, 2011, at 13 UTC (http://silam.fmi.fi/AQ_forecasts/public/index.html).

The model can be used for simulating episodes or long-term periods, for operational forecasting and for studying emission scenarios. Currently SILAM is the being used in Finnish Meteorological Institute as the main chemical transport model, generating the daily

air quality forecast for Finland- and European scale (silam.fmi.fi). SILAM is also been used in Estonia and Lithuania for air quality forecasts and there are several installations of SILAM in different countries around Europe, with the purpose to use SILAM as an assessment tool.

2.2 Model description

SILAM's dynamic core currently includes both Eulerian and Lagrangian advection-diffusion formulations. The Eulerian core used in the current study is based on the transport scheme of Galperin (1999, 2000), which incorporates the horizontal diffusion term and is combined with the extended resistance analogy of Sofiev (2002) for vertical diffusion. The removal processes are described via dry and wet deposition. Depending on the particle size, mechanisms of dry deposition vary from primarily turbulent diffusion driven removal of fine aerosols to primarily gravitational settling of coarse particles (after Slinn and Slinn, 1980; Zhang *et al*., 2001). Wet deposition distinguishes between sub- and in-cloud scavenging by both rain and snow (Sofiev *et al.*, 2006; Horn *et al*., 1987; Smith and Clark, 1989; Jylhä, 1991).

The principles implemented in the model enable handling of virtually any species with any types of interactions between them. SILAM is capable of computation of dispersion of up to 496 different nuclides, together with their radioactive decays and transformations; inert aerosol; biological material (pollen grains); chemically active gaseous compounds treated by chemical mechanisms with different complexity levels. The system accepts flexible definition of the particle size spectrum. Inside each range, the fluxes are integrated, and volume-weighted mean diameter of particles is accepted as representative for the range. This spectrum representation corresponds to the routine in-situ observations of several monitoring networks, which simplifies the model-measurement comparison. The model also computes probabilities, where the source represents the measurements of a measurement site and the result is the probability of a certain grid cell to be contributing to that measurement. This is set in the adjoint ("inverse") mode of the model.

SILAM internally computes emissions for pollen and sea salt. The pollen sources are based on phenological models and sea salt is based on a parameterization taking into account white cap area, sea surface temperature and salinity. Additionally, SILAM takes into account fire-related species source obtained from the Fire Assimilation System (FAS v1.1; Sofiev *et al.*, 2009). This system is based on data obtained from MODIS instrument in AQUA and TERRA satellites, and produces the emission fluxes originating from fires using the observed

characteristics of burning for each individual fire and speciated according to the available literature.

The internal structure of SILAM corresponds to the process-related split and takes into account the demand for system flexibility. These requirements are met via a strictly modular programming style, using FORTRAN-90 syntax, which is kept as close to the standards of object-oriented programming. Currently, the operational version of the model is v.3.8.1 for emergency response and v.4.5.4 for other applications.

2.3 Input data required

SILAM requires the typical input data for a chemical transport model: meteorological data, boundary conditions, land-use information, emissions. With the exception of the meteorological files, a default set of data can proposed to the user. Refinements on input data have to be made by the user according to the needs of the computation.

- **Meteorology:** standard sources of meteorological data are the forecasts and archives of the HIRLAM and ECMWF NWP models, being the framework capable of assimilating the information from virtually any source in which the fields are packed in the so-called GRIB or NetCDF format. For static meteorological data, model is also able to read text files.

- **Boundary conditions:** boundary conditions can be obtained from SILAM fields or from other global or meso-scale chemical transport models. Default data for boundary is available from the GEMS project (Global and regional Earth-system Monitoring using Satellite and in-situ data) is funded by the European Commission within the 6th Framework Program for Research and Development under the contract SIP4_CT-2004-516099.

- **Land-use**: the land-use information and biogenic emission potentials are based on land cover. SILAM is using information from the USGS classification (http://edc2.usgs.gov/glcc/globdoc2_0.php). The data for deriving biogenic emission potentials only cover Europe.

- **Emissions:** the anthropogenic emissions are obtained from different sources depending of the spatial resolution needed. Most common used is the TNO developed emission database for regional dispersion computations. TNO is an independent Dutch research

organization that has been compiling European emission inventories. For more detailed studies, local emission databases are recommended.

2.4 Operating system and software

The model has been used on PCs with Linux or Windows operating systems. The systems minimum requirements for assessment and forecast purposes, respectively, are:

- Windows: Windows server standard edition; Intel® Xeon® CPU with 16 or 8 dual-core processors, 64-bit processor; Memory (RAM): 24 GB; dual power supply unit; disk space: 500 GB; FORTRAN compiler Intel v.11 and code editor and debugger, Visual Studio 2008.
- Linux: 2x multicore 64-bit processor with 16GB RAM(DDR3) memory; dual power supply unit; FORTRAN compiler Intel v.11 or Gnu FORTRAN (supported by SILAM team with some restrictions).

The most commonly used scripting programs for running the computational routines and visualization of the output of SILAM are Python, Perl and Grid Analysis and Display System (GrADS), for their reliability and being freely available.

For operational purposes, the supervisor Monitor Scheduler (SMS) is utilized. This application is used for enabling a large number of programs which may have dependencies on one another, and in time, in a controlled environment with reasonable tolerance of both hardware and software failures, combined with good restart capabilities

2.5 How to use SILAM

A user guide is available for the different commercial versions available, download of the model code and documentation please visit silam.fmi.fi. The user-guide for the version 4.5.4 is presented in Annex III.

2.6 Toolbox

SILAM includes a set of supplementary tools including a meteorological pre-processor, input-output converters, grid transformers, interpolation routines, etc. These ensure a high flexibility to the user requests, enabling in particular the usage of various numerical weather prediction (NWP) models as data sources, with dynamical determination of internal model parameters such as the computation grid, vertical structure and a list of meteorological quantities needed for computations.

User-guide for SILAM chemical transport model

Introduction

Current document provides a minimum amount of information for manual run of the emergency model SILAM v.4.5.4. There are differences between the various versions and if the user is using other version previous to v4.5, should consult the previous user guides available at http:silam.fmi.fi. The document does not contain information about the model structure or any systematic description of its features, except for a few cases when such knowledge is absolutely necessary for the model configuration. The purpose of this instruction is to describe the files controlling the SILAM runs and available output options.

Substances and their transformation in SILAM v.4.5.4

SILAM v4.5.4 is capable of computation of dispersion of up to 496 different nuclides, together with their radioactive decays and transformations; inert and chemically active size-specific aerosol; biological material (pollen grains); chemically active gases. And this is so called the "forward" mode. The model also computes probabilities, where the source represents the measurements of a measurement site and the result is the probability of a certain grid cell to be contributing to that measurement. This is set in the "inverse" mode of the model.

The principles implemented in the model enable handling of virtually any species with any types of interactions between them. A mixture of the species transported in air is called "cocktail". In the version 4, the cocktail may be:

- radioactive (RADIOACTIVE),
- passive tracer (PASSIVE),
- inert aerosol (INERT),
- pollen (POLLEN),
- sea salt aerosol (SEA_SALT),
- chemically active compounds: sulphur linear chemistry (DMAT_SULPHUR), inorganic chemistry (ACID_BASIC) and CB4 chemistry scheme (CBM),
- persistent organic compounds (PERSISTENT_ORGANICS).

Radioactive type means that cocktail consists of one or more nuclides, with inherent radioactive, transport and deposition features. **Passive** cocktail includes an imaginary inert tracer, which does not interact with the environment or other species and is used for computation of probability distributions. **Inert aerosol**, **pollen** and **sea salt** cocktails handle a size-segregated aerosol with no chemical reactions involved. Pollen and sea salt species have specific emission source description: pollen is based on phenological models and sea salt is based on a parameterization taking into account white cap area, sea surface temperature and salinity. Cocktails are defined via cocktail description files (section 4.5). The user is free to define new cocktails, which must belong to one of the above types but may contain different species. Each emission source may be emitting its own cocktail. However, the types of cocktail must be the same for all sources, as well as the number of aerosol size classes.

In the case of probabilities, the model is then set in the "backwards" mode, where the source is representing the measurements of a measurement site and the result is the probability of probability of a certain grid cell to be contributing to that measurement.

Outline of the initialization/configuration files

SILAM v4.5.4 may have up to ten input files depending on the complexity of the setup (see the structure in Figure 1).

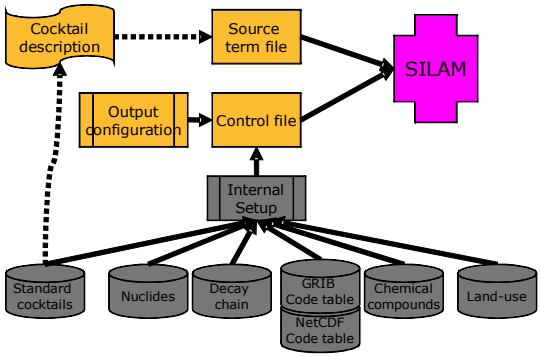

Figure 1. A structure of SILAM v4.5.4 configuration files

The mandatory files, for any run configuration, are:

- source term file: describes the emission sources;
- control file: sets the user-defined parameters of the run;
- internal model setup: referred from the control file, sets the internal model features, usually read-only or fully invisible for users;

- output configuration file: description of the output setup, referred from the control file;
- standard cocktails file: defines the standard cocktails that can be used in the source description; referred from the internal setup file. Users are free to create their own cocktails, adding to the existent file;
- GRIB or NetCDF code table (depending of the type of meteorological files): mandatory, invisible for users, referred from the internal setup file.

Depending of the configuration of the run, there are different files that should be included in the setup configuration:

- nuclide data file: for radioactive simulations, invisible for users, referred from the internal setup file;
- nuclide decay data file: for radioactive simulations, invisible for users, referred from the internal setup file;
- land-use data: for chemical simulations of biogenic emissions, invisible for users, referred from the internal setup file;
- optical properties: for chemical and aerosol simulation, describes the optical properties of substances, invisible for users, referred from the internal setup file;
- chemical properties: for chemical simulations, describes the chemical properties of the compounds chemically active, invisible for users, referred from the internal setup file;

Do NOT alter internal model files referred from the internal setup file; their modification may lead the model to malfunction. Only the source term, the control file, the output configuration file and the standard cocktails file are open to everyone. The description of the mandatory files will be described in this document. Nuclide data file, nuclide decay file, chemical and optical properties files, land use data and GRIB/NetCDF code table files must NOT be altered by the user in any consequences, and therefore are not included in the document.

All ini files are standard text (ASCII) files following one of the two standard formats: fixed-structure file or namelist-type file. There is an on-going work of transferring all configuration files to the namelist format but it is not yet completed. Currently, the namelist format is implemented only in source term file, control file, internal setup, cocktail description, and aerosol description. Output configuration and internal files follow the fixed-structure format.

3.1 General rules for the configuration files:

- Each file consists of a set of lines, with leading and trailing blanks ignored.

- Lines are cAsE-sEnsiTIve (at least some of them).
- Empty lines and commented lines are ignored. All characters after signs “#” or “ !” are considered as comments. **Note:** sign “#” always starts comments, while sign “!” starts comments ONLY if it is placed at the beginning of line or preceded by the empty space.

3.1.1 Rules for the namelist-type format:

- A single file includes a group of namelists, placed one-by-one in arbitrary order.
- A single namelist starts from the line “LIST = <namelist_name>” and ends with the line “END_LIST = <namelist_name>” (spaces around the “=” sign are mandatory). The namelist_name must be understood by the model. The namelist content is placed between these LIST – END_LIST lines.
- A namelist content is a set of lines of the following format:
  <item_name> = <item_value>, blank spaces around the “=” character are mandatory
  <item_name> must be understood by the model
  <item_value> format and meaning fully depends on the item_name and may vary from number to a complicated line with several space-separated fields order of the namelist lines is arbitrary, but all lines must conceive in-between the LISTEND_LIST boundaries of the list they belong to.
- unnecessary lines or lines with unknown item_name will be skipped by the model.

3.1.2 Rules for the fixed-structure format:

- The file consists of a set of lines, each containing one or more field(s), delimited by space.
- The line order is fixed.
- The field order in line is fixed.
- fields are space-separated.

Examples of the configuration files

4.1Source term files

The source file for SILAM v4.5 consists of a list of individual sources, following one-by-one. Each source is treated totally independently from the others. The source is always started from the **Header line** (see below) and ends by **End line**. There are three types of sources supported: bomb source, point source and area source. They all can appear in the same emission file. The only limitation is that the type of emitted species must be the same, as well as the aerosol size distribution. For example, either all sources emit passive tracer, or all sources are radioactive, or all sources are inert aerosol, etc. However, the specific list of

species and mass distribution between the aerosol size classes can vary from one source to another.

Below, the templates for point source v.4 and area source v.2. Examples of these files are presented in Annex I. Since there is an ambiguity in the definitions, some comments after each namelist (bullets and bold) clarify the possible choices. Older versions are described in previous documents about SILAM's operational instructions.

4.1.1 Point source v.4

This source term is compatible for forward and backward runs. The source file may contain several sources of this type, as well other types. Such point source has constant parameters during the whole release duration. The namelists are described below (bullets).

POINT_SOURCE_4

Header line, point-source starts, mandatory

- **source_name**. Source name. The source name has to be different.
- **source_sector_name**. Normally according to EMEP's sector denomination. May be empty.
- **source_longitude**. Source's geographical longitude, degrees and decimals, N positive, E positive.
- **source_latitude**. Source geographical latitude, degrees and decimals, N positive, E positive.

Plume characteristics. These are used only if plume rise routine is activated and ignored otherwise. The source height boundaries are also involved in the plume rise computation.

- **plume_rise** = PLUME_RISE_YES / PLUME_RISE_NO. Activates the buoyant plume rise routine.
- **release_rate_unit** = <mass>/<time>.

  Unit of the release rate (no spaces!!):
  - mass: kg][g][t][Bq][mole][number]
  - time: [yr][mon][day][hr][min][sec]
- **vertical_unit**. Unit of the vertical release boundaries [hpa] or [m]

Time-strength-composition data: The source is assumed to have the claimed parameters EXACTLY at the moment defined. It is also assumed that the source starts at time defined by the first line and stops at time defined by the last line.

1) if time of release is fixed-in-time source, fixed-in-time release is defined via two lines with identical parameters and with start and end time of the release. The source is activated at current moment ("NOW") or at last-most meteorological time ("LAST_METEO_TIME") and will continue constant-in-time release during the given duration.

- **par_str** = [NOW]/[LAST_METEO_TIME] <duration [min]> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>
- **par_str** = [NOW]/[LAST_METEO_TIME] <duration [min]> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>

2) if time of release is varying source, the first line determines the start of the release and last line determines the end of the release. There are an arbitrary number of lines and if two sequential lines have different release parameters, every parameter will be linearly interpolated between these times. A varying source is defined by a 4-digit year and a 2-digit month, day, hour and minute, seconds is a real value with mandatory decimal dot.

- **par_str** = <year> <month> <day> <hour> <minute> <sec> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>
- **par_str** = <year> <month> <day> <hour> <minute> <sec> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>

The release rate (rate) is the value of the release in the units defined by release_rate_unit (above). The horizontal size (xy_size) is the diameter of the source since sources are assumed to be circles. Bottom and top are the vertical boundaries of the emitted cloud (unit: meters or hPa). If the plume-rise routine is activated, the boundaries must be the same and correspond to physical height of the source. The vertical velocity (z-velocity) is the velocity of the plume at the top of stack (unit: meters per second). Temperature at the top of the stack of outgoing gases is defined by (tempr). The release composition (cocktail_name) points to one of the standard cocktails.

Time variation coefficients: represent an extra possibility to adjust the emission rate to the local time and season. This addition is useful only for long-term runs when such type of variation is important while its reproduction via par_str is bulky.

- **hour_in_day_index**. Diurnal relative intensity considering 24 hours in day.
- **day_in_week_index**. Week-day relative intensity considering 7 days in a week.
- **month_in_year_index**. Monthly relative intensity considering 12 months in a year.

END_POINT_SOURCE_4 # End line, point-source ends, mandatory

4.1.2 Area source v.2

This form represents a SILAM source term type: a spatially distributed emission source. Following the general standards, it is defined in some 3-dimensional grid, while the time dimension is represented in a very similar way as par_str in the above point sources. Grid and vertical definitions follow the standards of the GRID format. The source file consists of five main parts: general parameters, grid definition, vertical definition, time definitions and grid cell values. A template of the file is below and the namelists are described (bullets).

AREA_SOURCE_2 # area-source starts. Mandatory

- **source_name**. Source name. The source name has to be different.
- **source_sector_name**. Normally according to EMEP's sector denomination. May be empty.

 # Definition of the area source grid: location, resolution and rotation. All geographical values are in degrees and decimal parts of a degree, NO MINUTES/SECONDS

- **grid_type** = lon_lat. Geographical coordinates grid is so far the only available
- **lon_start**. Area source's longitude of the first grid cell - ksec2(5).
- **lat_start**. Area source's latitude of the first grid cell - ksec2(4).
- **dx**. x-direction increment (lon) - ksec2(9),
- **dy**. y-direction increment (lat) - ksec2(10),
- **nx**. Number of cells along the parallel (varying lon) - ksec2(2)
- **ny**. Number of cells along the meridian (varying lat) - ksec2(3)
- **lon_end**. Area source's latitude of the last grid cell - ksec2(8). Not needed if nx, dx, lon_start are defined
- **lat_end**. Area source's latitude of the last grid cell - ksec2(7). Not needed if ny, dy, lat_start are defined
- **lat_s_pole**. Latitude of the south pole of rotation (-90. for geo) - ksec2(13)
- **lon_s_pole**. Longitude of the south pole of rotation (0. for geo) - ksec2(14)

- **lat_pole_stretch**. Latitude of pole of stretching (0 so far) - ksec2(15)
- **lon_pole_stretch**. Longitude of pole of stretching (0 so far) - ksec2(16)
- **resol_flag**. Resolution flag. DEFAULT: 128 = regular grid - ksec2(6),
- **ifReduced**. Regular/reduced grid flag. DEFAULT: 0=regular - ksec2(17),
- **earth_flag**. Earth-flag, 0=sphere, 64=oblate spheroid. DEFAULT: 0 - ksec2(18),
- **wind_component**. Wind flag, 0=u,v relate to east/north, 8=u,v relate to x/y growing - ksec2(19),
- **reduced_nbr_str**. Number of elements along the reduced direction, in one line - ksec2(23+)

Definition of the emission:

- **cocktail_composition** = COMMON/SPECIFIC. The cocktail composition (cocktail_composition), describes the fractionation of the emission between the species can be defined in each time slot for all the cells (COMMON) or be specific for each cell (SPECIFIC). If cocktail_composition is SPECIFIC, then the number of values is n_substances*(n_aerosol_modes+n_gaseous_modes
- **emitted_substance** = WHOLE_COCKTAIL/"substance". This defines if the definition of the source is for the whole cocktail or for just one substance
- **size_class_split** = COMMON/SPECIFIC. The fractionation distribution between the aerosol size classes and gas phase can be defined in each time slot for all the cells (COMMON) or be specific for each cell (SPECIFIC)
- **emitted_size_mode_nbr**. Number of sizes describing the aerosol.

Vertical characteristics of the area source: vertical levels are set via fixed layers (not necessarily continuous). These are defined by the height of the bottom and top of layer (vertical release boundaries) and the fraction of emission emitted, in as many layers as needed to define the area source.

- **vertical_unit**. [hpa] or [m] according to the type chosen
- **vert_level** = <type> <layer_bottom> <layer_top> <fraction>.

The type of vertical level (type) can be defined as: HEIGHT_FROM_SURFACE (height measured from the surface), ALTITUDE_FROM_SEA (height measured from the sea level) and PRESSURE (pressure levels). The bottom (layer_bottom) and top (layer_top) of the layer sets how the model defines the position of the layer.

Time-strength-composition data: The source is assumed to have the claimed parameters EXACTLY at the moment defined. It is also assumed that the source starts at time defined by the first line and stops at time defined by the last line. The source parameters are:

- **par_str** = <year> <month> <day> <hour> <minute> <sec> <rate> <bottom> <top> <cocktail_name>
- **par_str** = <year> <month> <day> <hour> <minute> <sec> <rate> <bottom> <top> <cocktail_name>

Time variation coefficients (see Point Source above)

- **hour_in_day_index**. Diurnal relative intensity considering 24 hours in day.
- **day_in_week_index**. Week-day relative intensity considering 7 days in a week.
- **month_in_year_index**. Monthly relative intensity considering 12 months in a year.

Definition of the values for each grid cell

- **coordinate_of_values** = COORDINATES
- **val** = <lon> <lat> <value>. All coordinates and emission value are REAL

or

- **coordinate_of_values** = GRID_INDICES
- **val** = <hor index> <vert index> <value>. All grid indexes are INTEGER and emission value are REAL

END_AREA_SOURCE_2 # End line, area source ends, mandatory

There are a few critical differences between the above area source definition and the point source files. They all originate from one more dimension of parameter variations – spatial – that has to be taken into account. In the point source definition, there is only one vertical layer where the emission goes to. All sophisticated considerations are supposed to be solved via a plume rise routine. Such approach does not work with the area sources. Therefore, there are two ways allowed for the description of the vertical distribution: time-varying single layer defined in par_str for corresponding times (resembling the approach of point sources), and multi-layer distribution that is fixed in time but allows split of emission between the layers (see vertical_distribution and vertical_layer in the above example).

Another ambiguity is connected with the composition of the release. Species mass fractions in cocktail may vary between the grid cells. To take this into account, another two-option selection is introduced (switcher is the cocktail_composition line). The first option is the same

as in point source: the cocktail name is taken from par_str, its composition is taken from the cocktail description file (chapter “Standard cocktails” below) and assumed the same for all grid cells.

Time variation of the composition is then reproduced via cocktail definition – as is done in the point source. The second option is to use fixed-in-time but varying-in-space cocktail composition. In this case, the cocktail name in the par_str lines defines only lists of species and aerosol size classes, while the mass fractions are written in the val lines – specifically for each grid cell. In the latter case, there must be an agreement between the number of mass fractions in the val lines and the number of species in the cocktail descriptors references in the par_str lines.

It is also possible to create sources with dynamical emission rates computed with regard to meteorological parameters, which is mandatory for biogenic emission. This is the case of, e.g., POLLEN_COCKTAIL where the source term contains the map of pollen-emitting areas (in area fractions), while the control (example below) file defines the method how the actual emission flux is computed.

4.2 Control file example

Below an example of control v.4 is presented, an example of this file is presented in Annex II. Control file is a namelist group that contains four namelists:

- general_parameters
- meteo_parameters
- chemistry_parameters
- output_parameters.

4.2.1 CONTROL_V4 # Starting line, mandatory

#The **general_parameters** describe the

- **LIST = general_parameters**. Start of the namelist
  - **case_name**. Name of the case study, if "-" => use source name
  - **system_setup**. Full path & name of setup file: standard.setup (lagragian or eulerian)
  - **direction_in_time** = FORWARD/INVERSE. The direction in time of the run.
  - **start_time** = <year> <month> <day> <hour> <minute> <sec>. [yyyy mm dd hh mm sec(real)] or "-" to use emission start time
  - **end_time or computed_period**. Length of dispersion run & unit, integer > 0

- **time_step**. Timestep in calculation & unit, integer > 0
- **nbr_of_particles**. Only used by the lagrangian run
- **nbr_of_out_trajectories** = 0 or >0. Zero for no trajectory output
  - **END_LIST = general_parameters**

The **meteo_parameters** namelist describes the parameters related to meteorology or meteorological files such as location and meteorological time step. The meteorological files can have temporal variation - dynamic files (dynamic_meteo_file) - or not - static files (static_meteo_file). The dynamic meteorological files follow a template described below. The type of meteorological files accepted by SILAM v4 is: ASCII, GRIB and NETCDF.

- **LIST = meteo_parameters**
- **dynamic_meteo_file** = <file type> <file name>. File type: GRIB / ASCII / NETCDF andfilename is described in section 5.
- **static_meteo_file** = <file type> <file name> /- (the dynamic file is used)
- **meteo_time_step**. Weather data time interval number and unit, integer > 0
- if_wait_for_data = YES/NO, if yes, model waits for the meteorological files
  - **abl_parameterization_method** = DRY_ABL/FULL_PARAM. Sets the methodology for the boundary layer height computation. The methods available for the computation are DRY_ABL and FULL_PARAM. DRY_ABL parameterization is computing atmospheric boundary layer without humidity correction and FULL_PARAM includes humidity correction.
  - **number_of_precipitation_fields** = 1/ 2. If only large-scale rain is required and available the user should use 1; if both convective and large-scale rain required and available the user should use 2.
  - **END_LIST = meteo_parameters**

The **chemistry_parameters** namelist sets the parameters needed for the chemical and physical processes undergoing during the computation. Some of the namelists are specific for a cocktail type.

- **LIST = chemistry_parameters**
- **particle_scavenging_method** = STANDARD_3D_SCAVENGING. The only wet deposition method available.
- **particle_dry_depostion_method** = GRAVITATIONAL_ONLY/SIMPLE_DIFFUSION_ONLY/

FULL_DIFFUSION_ONLY/GRAVITATIONAL_AND_SIMPLE_DIFFUSION/

GRAVITATIONAL_AND_FULL_DIFFUSION. The dry deposition method can include gravitational settling or/and diffusion. SIMPLE_DIFFUSION_ONLY is only considering viscous sub-layer resistance.

- **if_full_acid_chemistry** = YES/NO. Sets if nitrogen chemistry is computed or not; method to compute biogenic VOC emissions (only for cocktail types: ACID_BASIC and CB4).
- **biogenic_voc_emission_method** = GUENTHER_METHOD_UPDATED. The only method available to compute VOC emission
- **if_actual_humidity_for_particle_size** = YES/NO. Sets if humidity is time-resolving or not.
- **default_relative_humidity**. Sets the default value for relative humidity.
- **compute_thermodiffusion** = YES/NO. Sets if the model computes thermodiffusion or not.
- **mass_low_threshold** = CRUDE_ACCURACY/STANDARD_ACCURACY/HIGH_ACCURACY. Sets the accuracy for the computation of the low-mass threshold for the Eulerian advection.

FOR SEA SALT COCKTAIL ONLY

- **default_water_temperature**. Sets the default value water surface temperature.
- **§default_water_salinity**. Set the default value for water salinity.
- **min_open_water_area_fraction**. Sets default value for the fraction of open water area in case of icy conditions.
- **sea_salt_emis_depend_on_water_tempr** = YES/NO. Sets if sea salt emission computation is dependent on water temperature or not.
- **sea_salt_emis_depend_on_water_salinity** = YES/NO. Sets if sea salt emission computation is dependent on water salinity or not.
- **sea_salt_emis_depend_on_ice_fraction** = YES/NO. Sets if sea salt emission computation is dependent on water temperature or not.

FOR POLEN COCTAIL ONLY

- **heatsum_scaling_to_degree_second** = 86400. # [deg day]=>[deg sec]:86400; [deg hr]=>[deg sec]:3600

- **low_humidity_threshold**. Full emission up to this value.
- **high_humidity_threshold**. No emission above this value.
- **precipitation_threshold.** No emission above this value.
- **wind_speed_saturation_level**. The impact grows up to this speed
- **wind_speed_max_impact**. To scale the emission scales emission from (a-1) up to a
- **climate_pollen_amount_per_m2**. Number representing the climatologic total number, [nbr grains / m2]
- **uncertainty_of_heat_sum_threshold**. Uncertainty of heat-sum threshold value, [%]
- **uncertainty_of_calendar_day_threshold**. Uncertainty of heat-sum threshold value, [%]
- **uncertainty_of_total_pollen_amt**. Uncertainty of climatologic total-pollen per m2, [%]
- **supplementary_info_file** = TEST_FIELD pollen_amt_4_year_corr SURFACE_LEVEL 1. File to correct the climate total pollen.

END_LIST = chemistry_parameters

#The initial and boundary conditions are described by **initial_and_boundary_conditions namelist**. Both initial and boundary conditions can be void. If existent both include a list of initialised/constrained quantities and the list of input files.

- **LIST = initial_and_boundary_conditions**
- **boundary_type** = ZERO/DIRICHLET. Boundaries can be static (ZERO) or time-resolving (DIRICHLET)
- **if_lateral_boundary** = YES/NO. If lateral boundary is or not set to the values prescribed in the boundaries file.
- **if_top_boundary** = YES/NO. If top boundary is or not set to the values prescribed in the boundaries file
- **if_bottom_boundary** = YES/NO. If bottom boundary is or not set to the values prescribed in the boundaries file
- **boundary_time_step** = <timestep> <unit>
- **boundary_header_filename**. Filename of the file describing the concentrations at the boundaries. The boundary file itself maps input data concentration for boundaries and transport species.

- **initialize_quantity**. Describes which quantity is being initialized. In SILAM's case is concentration.
- **initialization_file** = <file type> <file name>. The file type can be GRIB, GRADS and POINT_DATA. If GRADS type the file to be used is a super ctl file. This file is a standard output of any SILAM v4 run.

- **END_LIST = initial_and_boundary_conditions**

the **output_parameters** namelist sets the output configuration: variables and temporal and spatial resolution.

- **LIST = output_parameters**
- **source_id** = NO_SOURCE_SPLIT/ SOURCE_NAME/SOURCE_SECTOR/ SOURCE_NAME_AND_SECTOR. Controls mixing or splitting of the plumes from individual sources in the output files. In case of MIX_SOURCES, the plumes are mixed, so that all the sources create a single output field or trajectory set. If sources are split – each plume from the corresponding source is put into its own file, thus creating a surrogate for the source-receptor matrix computations. The source may have name and sector – and they both can be used for the creation of the source ID (NO_SOURCE_SPLIT) or according to source name and/or sector.
- **vertical_method** = CUSTOM_LAYERS/OUTPUT_LAYERS. Defines the vertical definition according to the user needs (CUSTOM_LAYERS)

level_type = HEIGHT_FROM_SURFACE/ALTITUDE_FROM_SEA/PRESSURE/HYBRID. There are 3 types of the output vertical allowed: z-, p- and hybrid systems, with corresponding units as: metres, hectoPascals or hybrid relative numbers. If the hybrid layers are selected, they MUST exist in the meteodata. The difference between the levels and layers is that levels are defined at one altitude, while layers cover the whole range between two levels. Dispersion output must be made into layers, while meteorology makes sense at levels too.

Rules: z-, p- systems accept both THICKNESS of the layers and their CENTRAL POINTS; hybrid system accepts the NUMBER of the meteo hybrid and model will get the central point.

- **layer_thickness**.Thickness output levels [m]/[pa]/[hybrid_nbr], reals
- **output_time_step**. Output timestep and unit

- **output_times** = REGULAR
- **file_types** = GRIB_NO TRAJECTORY_NO GRADS_YES ENSEMBLE_NO
- **time_split** = ALL_IN_ONE/ HOURLY_NEW_FILE/DAILY_NEW_FILE/MONTHLY_NEW_FILE/ YEARLY_NEW_FILE. Time
- **template**. Path for output dumping\%case\%case_%y4%m2%d2
- **variable_list**. Path for output_config file and output filename
- **grid_method** = EMIS_GRID/ METEO_GRID/AREA_BASED/CUSTOM_GRID. Grid definition for the output files. The same definition as emission or meteorological files (EMIS or METEO_GRID) or according to specific needs.

If AREA_BASED, the output area and required resolution have to be defined

- **area_borders** = <south> <north> <west> <east>; North positive, east positive; all real.
- **area_title**. A name for the area defined
- **resolution**. Horizontal grid size of output grid, [km]/[m]/[deg], real

If CUSTOM_GRID, a full definition of the grid has to be described:

- **grid_type** = lon_lat. Geographical coordinates grid is so far the only available
- **grid_title**. A name for the grid.
- **lon_start**. Area source's longitude of the first grid cell - ksec2(5).
- **lat_start**. Area source's latitude of the first grid cell - ksec2(4).
- **dx**. x-direction increment (lon) - ksec2(9),
- **dy**. y-direction increment (lat) - ksec2(10),
- **nx**. Number of cells along the parallel (varying lon) - ksec2(2)
- **ny**. Number of cells along the meridian (varying lat) - ksec2(3)
- **lon_end**. Area source's latitude of the last grid cell - ksec2(8). Not needed if nx, dx, lon_start are defined
- **lat_end**. Area source's latitude of the last grid cell - ksec2(7). Not needed if ny, dy, lat_start are defined
- **lat_s_pole**. Latitude of the south pole of rotation (-90. for geo) - ksec2(13)
- **lon_s_pole**. Longitude of the south pole of rotation (0. for geo) - ksec2(14)

- **lat_pole_stretch**. Latitude of pole of stretching (0 so far) - ksec2(15)
- **lon_pole_stretch**. Longitude of pole of stretching (0 so far) - ksec2(16)
- **resol_flag**. Resolution flag. DEFAULT: 128 = regular grid - ksec2(6),
- **ifReduced**. Regular/reduced grid flag. DEFAULT: 0=regular - ksec2(17),
- **earth_flag**. Earth-flag, 0=sphere, 64=oblate spheroid. DEFAULT: 0 - ksec2(18),
- **wind_component**. Wind flag, 0=u,v relate to east/north, 8=u,v relate to x/y growing - ksec2(19),
- **reduced_nbr_str**. Number of elements along the reduced direction, in one line - ksec2(23+)
  - END_LIST = output_parameters
  - END_CONTROL_V4_1 # end of control file, mandatory.

4.3 Model setup file

The model setup file contains internal model configuration and should not be modified without clear understanding of consequences. However, it MUST be adjusted if the model is moved to another location because it contains full paths to other internal files. An example of such file is presented in Annex III.

- **BEGIN_STANDARD_SETUP**
  - **advection_method** = EULERIAN_3D/LAGRANGIAN_3D/LAGRANGIAN_2. Advection method and dimension.
  - **random_walk_method** = IDIOT_SIMPLE, FULLY_MIXED, BULK_GAUSSIAN. Random-walk method
  - **abl_height_method** = CONSTANT/RICHARDSON/PARCEL/COMBINATION/NWP_ABL. ABL computation method.
  - **horizontal_interpolation** = LINEAR/NEAREST_POINT/SECOND_ORDER/CUBIC/ LOG_LINEAR
  - **vertical_interpolation** = LINEAR/NEAREST_POINT/SECOND_ORDER/CUBIC/ LOG_LINEAR
  - **time_interpolation** = LINEAR/NEAREST_POINT/SECOND_ORDER/CUBIC/ LOG_LINEAR
  - **nuclide_database_fnm** = path…\silja_nuclides_mas.dat
  - **nuclide_chain_fnm** = path…\silja_chains_mas.dat

- **standard_cocktail_fnm** = path…\standard.cocktails
- **grib_code_table_fnm** = path…\grib_code_table.silam
- **netcdf_name_table_fnm** = path…\netcdf_name_table.silam
- **land_use_data_meta_file** = path…\land_use_features_USGS_Eurasia.dat
- **allow_zero_forecast_length** = NO/YES. Whether zero-long forecasts are allowed in the simulations. Beware of zero-valued accumulated fields in such forecasts
- **precipitation_low_limit** = <value> <unit>. Cut-off limit for precipitation rate (separate for large-scale and convective)
- **print_debug_info** = DEBUG_INFO_YES/DEBUG_INFO_NO. to print or not a file for debugging.
- **cloud_report_interval** = 1/2. A total mass report frequency. 1 means that the total mass summary is reported in the log file every time the output is stored, 2 is the total mass is not reported. Such report is time consuming.
- **disregard_meteo_data_sources** = YES/NO. If several meteo data sources are allowed for the run. A safe answer is NO but sometimes it is safe to allow this.

- **END_STANDARD_SETUP**

4.4 Output configuration file

The output post-processor allows the user to select flexible averaging for each dispersion variable and to include any SILAM internal meteorological variable to the output. The file contains arbitrary number of lines containing three or four or five fields (see Anexx IV for examples):

- necessity index: 0 – quantity is not needed, 1 – desirable, 2 – mandatory for the output
- variable name
- substance name/lists: if not an individual substance, there can be requested:
  - SOURCE_INVENTORY, just the substances emitted.
  - FULL_INVENTORY when requested all the substances present in the dispersion cloud.
- averaging type for the particular variable. The available types of averaging are:
  - AS_IS – the field comes to the output exactly as it was stored in SILAM internal buffers at the moment of output collection
  - INSTANT – cumulative field is converted to their mean rates between the last two model time steps, while the instant variables go as they are

- CUMULATIVE – the variable is accumulated since the beginning of the simulations
- AVERAGE – the variable is averaged from the previous to the current output time
- MEAN_LAST_**_HR – the field is averaged over the given period preceding the current output. The period must not be longer than the interval between the outputs.

- Wavelength: only for the optical properties.

The output file has different output variable categories:

- General characteristics of the output variables:
- Physiography
- Dispersion
- meteorological

An extraction from the main output configuration file is the following:

LIST = OUTPUT_CONFIG_3_7

General characteristics of the output variables

aerosol_size_modes = SUM or SEPARATE - a way to report aerosol size modes in the output

SILAM dispersion model variables:

Emission fields

- **out_var** = 0/1/2 **emission_flux** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

Permanent fields (physiography)

- **out_var** = 0/1/2 **physiography_field_set** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

Particle counter and vertically integrated particle counter

- **out_var** = 0/1/2 **particle_counter** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **areas_of_risk** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

Nuclides existing in the source inventory - concentr. and deposition

- **out_var** = 0/1/2 **concentration** [SOURCE_INVENTORY] /[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **drydep** [SOURCE_INVENTORY] /[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **wetdep** [SOURCE_INVENTORY] /[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

All species from source inventory AND from transformation chain - conc and dep.

- **out_var** = 0/1/2 **concentration** [SOURCE_INVENTORY]/[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **drydep** [SOURCE_INVENTORY]/[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **wetdep** [SOURCE_INVENTORY]/[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

Diagnostic optical depth

- **out_var** = 0/1/2 **optical_density** [SOURCE_INVENTORY]/[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT %WAVE_LENGTH nm 550.
- **out_var** = 0/1/2 **optical_column_depth** [SOURCE_INVENTORY]/[FULL_INVENTORY] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT %WAVE_LENGTH nm 550.

SILAM meteorological variables

- **out_var** = 0/1/2 **temperature** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0/1/2 **Vd_correction_DMAT** AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

Full list of SILAM nuclides - you are free to choose each of them

- **out_var** = 0 **silam_cocktail** [H_3] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT
- **out_var** = 0 **silam_cocktail_dep** [H_3] AS_IS/AVERAGE/TOTAL_WHOLE_PERIOD/INSTANT

END_LIST = OUTPUT_CONFIG_3_7

There are four different output file formats. Each of them is to be ordered explicitly in the control file.

- GRIB file. The model output is stored into a single GRIB file. Appropriate .ctl file for GrADS visualization is accompanying the main binary. The format requires the GRIB code table to contain the code numbers for all output variables.
- GrADS file. Same content as in GRIB file. Also accompanied with the .ctl file. There is a reserved structure for the ASCII file output format but it is not yet completed. When done, it will follow the GRIB binary structure but write all the values in the text file rather than in the compressed binary.
- Trajectory file. Made for the backward compatibility only. Reproduces the TRADOS style of the trajectory output with the fixed content.

4.5 Standard cocktail file

Cocktail description files contain lists of cocktails. Cocktail description consists of the cocktail name, type, unit of fractions and then a list of species with their fractions (in corresponding unit) in the cocktail. The description starts from header and ends with end line.

The cocktail may contain the aerosol description. Standard cocktails can be used by their names in the source term files. Different examples can be found in Annex V.

COCKTAIL_DESCRIPTION_V3_2

- cocktail_name = random name
- cocktail_type. Coctail types defined in the model (see section 2)
- mass_unit = Bq/number/mass
- gas_phase = YES/NO
- aerosol_mode = <min> <max> <average diameter> <unit>
- aerosol_density = 800. kg/m3 # Value with unit
- aerosol_distribution_shape = FIXED_DIAMETER # so far the only available
- component_fraction = <Component name> <mass fraction in the mixture>
- component_fraction = <Component name> <mass fraction in the mixture>

END_COCKTAIL_DESCRIPTION

Depending on whether the aerosol size classes are defined, the fractions have somewhat different meaning. If the aerosol classes are defined, the fractions represent a distribution of the substance between these classes. A total mass fraction of each substance in the mixture comes as a sum of fractions of the substance in the aerosol classes. Due to a high complexity of this scheme, user does not forced to normalize the total fraction sum to 1 – it will be done automatically while reading the cocktail description.

4.6 Common structures used in the configuration files

4.6.1 Templates of the meteo data files and output

Path and names of the meteodata files and the output files are written in the control file in a usual format <full_path><file_name>, where both <path> and <name> can vary depending on their content. For example: /data/hirENOfields/fc%ay4%am2%ad2%ah2%f2*, where /data/hirENOfields/ is the path to the files and the file name itself depends on the analysis time and forecast length of the fields stored in it: fc%ay4%am2%ad2%ah2%f2*. The file name allows wildcards “*” and “?” with usual meaning:

- “*” substitutes any number of characters, including nothing;

- “?” substitutesnot more than one character.

In the above example, "*" is introduced to show that, for example, two files fc200207190600 and fc200207190600md must be both read to get the data for the corresponding time. ATTENTION. Use wildcards with care – ensure that there are no "strangers" satisfying the provided template. Violation may delay the model work or lead to abort of simulations. Wild cards are not allowed in the output file templates. Allowed templates are:

Templates pointing to the analysis time (not allowed for the output files):

- %ay2; %ay4 – firmly 2- and 4-digit year of the analysis time (e.g. "02" or "2002")
- %am1; %am2; %amc – 1 or 2-digit; firmly 2-digit; 3-character month of the analysis time (e.g."1"; "01" or "JAN")
- %ad1; %ad2% - 1or 2-digit; firmly2-digit day of the analysis time (e.g. "5"; "05")
- %ah1; %ah2; %ah3 – 1 or 2 or 3-digit; 2 or 3-digit; firmly 3-digit hour of the analysis time (e.g."7"; "07"; "007")
- %an2 – 2-digit minutes of the analysis time (e.g. "15")

Templates pointing to the valid time of the fields are constructed in the same way:

- %y2; %y4 – firmly 2- and 4-digit year of the analysis time (e.g. "02" or "2002")
- %m1; %m2; %mc – 1 or 2-digit; firmly 2-digit; 3-character month of the analysis time (e.g. "1";"01" or "JAN")
- %d1; %d2% - 1or 2-digit; firmly2-digit day of the analysis time (e.g. "5"; "05")
- %h1; %h2; %h3 – 1 or 2 or 3-digit; 2 or 3-digit; firmly 3-digit hour of the analysis time (e.g. "7";"07"; "007")
- %n2 – 2-digit minutes of the analysis time (e.g. "15")

Templates pointing to the forecast length are:

- %f2; %f3 – 2- and 3-digit number of hours of the forecast length (e.g. "15"; "015")

Templates pointing to the simulation titles:

- %case; %source; – the name of the case or ID of the source is inserted. In v.3.5, if there is an output option MIX_SOURCES, all names of all sources will be connected one-by-one in a line with underscore as delimiters. A corresponding option in v.3.6 is NO_SOURCE_SPLIT, which would result in "all_sources" appearing as a value of the %source template item.

4.6.2 Time control of the simulations

There are several parameters determining the time limits of the simulations:

- start of the simulations;

- duration of the simulations;
- start of the source release;
- duration of the source release.

The model behavior differs for forward and inverse tasks:

1. Forward task

    a. If start and duration of the simulations are defined – the model will follow them.

    b. If source release is longer than the simulations – it will be cut out.

    c. If source starts later than the simulations – a cloud will be empty until the release start time.

    d. Source start term “NOW” means that the release will start from the closest 6-hour meteotime – 00, 06, 12, 18 UTC of the current day.

2. Inverse (adjoint) task. “Source” actually becomes a receptor receiving the pollution cloud. The main time direction in the model is backward.

    a. The release time defines the period when the receptor receives the particles (“receptor is active”). It is – from start till start+duration.

    b. At least part of the receptor active time must be covered by the simulation time. So, the simulation start must be AFTER the source (receptor) starts. Then the model will go backward in time till the source(receptor) start.

    c. Source start term “NOW” means that the receptor activity starts from the closest 6-hour meteotime – 00, 06, 12, 18 UTC of the current day. Note that the simulation must start AFTER this moment. This is dictated by the inverse advection time in the model.

    d. If the simulation start time is undefined, it is set automatically to <source(receptor) start> + <release duration>. So, the model will go from the end of the active receptor time backward. Covered period is determined by the duration of the simulations.

5 Running the model: command line options

There are two input parameters to be given to the model: source file name and control file name. This can be done via one of the following command line constructions. Notations below are:

- <program> is the path and or name of the SILAM executable,
- <control_file> is the control path and or file name:

- <source_file> is the source path and or file names:

1) ~> <program>. No arguments. The program will open the file “pasi.ini” in the working directory and read names of source and control file names and output directory from it. The file contains a namelist with just two items: source_file and control_file

2) ~> <program> <ini_file_name>. One argument, which is treated as a main ini-file name instead of “pasi.ini” above. This file must contain the namelist as described above.

3) ~> <program> <source_file> <control_file>. Both files are given explicitly.

6 Program code, developments and updates

The ready-made program source code cab be requested through htpp://silam.fmi.fi. All individual developments should be done under own personal user accounts. Developers should keep complete copies of the code under their own accounts and merge them with the operational one after appropriate testing and informing the other users and developers.

Annex I. Requirements for the meteorological data

The database for the WRF-NMM meteorological fields for future high resolution computations should contain data with the meteorological parameters are listed in the table below:

Surface level parameters	Model level parameters
2 meter temperature	Temperature
2 meter dewpoint temperature	Wind: u,v and w component
10 meter wind: u and v component	Specific humidity
Surface pressure or logarithm of surface pressure	Cloud cover
Large scale precipitation	
Convective precipitation	
Land-sea mask	
Surface sensible heat flux	
Surface latent heat flux	
Boundary layer height	
Forecast albedo	
Cloud cover	
Forecast surface roughness	
Forecast log of surface roughness for heat	

There is no restriction for the spatial horizontal resolution but the vertical resolution should be provided in hybrid levels for all the available levels. The time resolution should be of 3 or 6 hours.