CAABA/MECCA-2.5j User Manual

Chemistry As A Boxmodel Application /

Module Efficiently Calculating the

Chemistry of the Atmosphere

Rolf Sander, Patrick Jöckel, Astrid Kerkweg & Hella Riede

Air Chemistry Department
Max-Planck Institute of Chemistry
PO Box 3060, 55020 Mainz, Germany
sander@mpch-mainz.mpg.de

www.mpch-mainz.mpg.de/~sander/messy/mecca/

Date: 2009-02-24

Contents

1 Introduction

MECCA (Module Efficiently Calculating the Chemistry of the Atmosphere) is an atmospheric chemistry module that contains a comprehensive chemical mechanism with tropospheric and stratospheric chemistry of both the gas and the aqueous phase (Sander et al., 2005). For the numerical integration, MECCA uses the KPP software (Sandu and Sander, 2006).

To apply the MECCA chemistry to an atmospheric scenario, MECCA must be connected to a base model. As shown in Fig. 1, the base model can be a complex, 3-dimensional model (e.g. Jöckel et al., 2006) but it can also be a simple box model. The connection is established via the MESSy interface (`http://www.messy-interface.org`) developed by Jöckel et al. (2005).

This manual describes how to install and work with MECCA when it is connected to the box model CAABA (Chemistry As A Boxmodel Application). This combination will be referred to as "CAABA/MECCA". The main features of the CAABA box model are shown in Fig. 2. In addition to MECCA chemistry, CAABA also contains modules for calculating J-values (JVAL), simplified and parameterized photolysis rates (SAPPHO), and simplified emission and depostion (SEMIDEP).

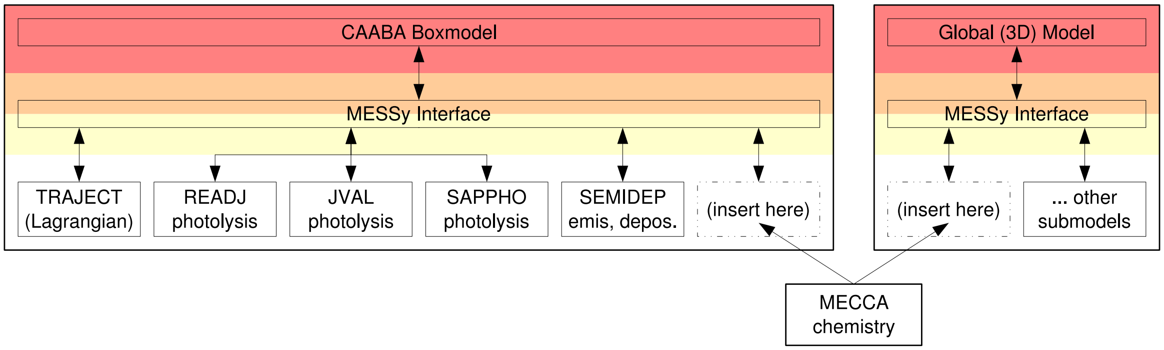

Figure 1: Diagram showing MECCA as part of the CAABA box model or of a global model.

2 Installation

This section can be skipped if CAABA/MECCA is already installed on your computer.

CAABA/MECCA has been tested successfully on several UNIX-like operating systems. The easiest installation is probably on a Linux PC since several auxiliary programs are already included in a typical Linux distribution. Installation under "Windows" is neither recommended nor supported. CAABA/MECCA consists of the Fortran90 files listed in Tab. 1. The prerequisites are:

A Fortran90 compiler (mandatory): Several compilers have been tested successfully: g95 (for Linux), Lahey (for Linux), Intel (for Linux), Compaq (Alpha UNIX). Other compilers can be used as well if they accept standard Fortran90 code. It should be noted that the g95 compiler for Linux is free and can be downloaded from `http://www.g95.org/`.

The Kinetic PreProcessor KPP (mandatory): This flexible numerical integration package by Sandu and Sander (2006) transforms the chemistry mechanism into a set of ordinary differential equations (ODEs) in Fortran90 syntax. MECCA needs the KPP version that is provided in the `mecca/kpp/` directory.

Perl, tcsh, gawk, sed, and make (mandatory): These UNIX tools are standard on Linux systems. Please check that recent versions of them are installed. Especially gawk may lead to strange error messages. To test gawk, type:

```
gawk 'BEGIN {print match("X","[^a-z]")}'
```

The result should be "1". However, you may get "0" as the result on your system. Supposedly, this is not a bug in gawk but a feature. You can solve the problem by setting the environment variable `LC_COLLATE` to "C":

```
export LC_COLLATE=C      (if you use bash)
setenv LC_COLLATE C      (if you use tcsh)
```

When you try the gawk test again, it should work fine.

LaTeX (optional): If you have LaTeX installed on your computer, you can print a table (including rate coefficients and references) of the currently selected mechanism (see Sect. 6.3.3 for details).

netCDF library (optional): The netCDF library is needed to create model output in netCDF format. It can be obtained from `http://www.unidata.ucar.edu/software/netcdf/`. Software for manipulating or displaying netCDF data is listed at: `http://www.unidata.ucar.edu/software/netcdf/software.html`. If you don't have the netCDF library, you can still run the model but produce only ascii output.

ferret (optional): The ferret plotting program is needed to plot the contents of the netCDF output using the ferret scripts in the `jnl/` directory (see Sect. 4 for details).

Once all prerequisites are fulfilled, you can install CAABA/MECCA by simply unpacking the zip archive:

```
unzip caaba_2.5j.zip
```

Please make sure that the directory structure has not changed during the unzipping process. Unfortunately, some unzipping programs seem to put all files into one directory, ignoring the original directory structure.

Next, you have to check that all settings in `Makefile` are correct. If necessary, edit the file: Choose a Fortran90 compiler (`COMPILER`), enter its name (`F90`) and the compiler options (`F90FLAGS`). If you add a new

Table 1: List of CAABA/MECCA Fortran90 files

CAABA box model related files	
`caaba.f90`	main box model file
`caaba_io.f90`	input/output
`caaba_io_netcdf.inc`	netCDF input/output
`caaba_io_ascii.inc`	ascii input/output
`caaba_mem.f90`	declaration of CAABA variables
`messy_main_control_cb.f90`	flow control
`messy_jval_box.f90`	connection of JVAL to CAABA
`messy_mecca_box.f90`	connection of MECCA to CAABA
`messy_sappho_box.f90`	connection of SAPPHO to CAABA
`messy_semidep_box.f90`	simplified emission and deposition, including connection to CAABA
`messy_traject_box.f90`	trajectory calculations (under construction)
static core files	
`messy_main_constants_mem.f90`	physical constants
`messy_main_blather.f90`	print utilities
`messy_main_tools.f90`	auxiliary functions
`messy_main_tools_kp4_compress.f90`	(file exists but is not not used with CAABA)
`messy_jval.f90`	calculation of J-values
`messy_sappho.f90`	simplified and parameterized photolysis rate coefficients
static MECCA core files in the `mecca/smcl/` directory	
`messy_mecca.f90`	MECCA core
`messy_mecca_aero.f90`	aerosol chemistry
`messy_mecca_khet.f90`	(file exists but is not used with CAABA)
KPP- and `xmecca`-produced files in the `mecca/smcl/` directory	
`messy_mecca_kpp.f90`	a wrapper for the KPP files
`messy_mecca_kpp_function.f90`	ODE function
`messy_mecca_kpp_global.f90`	global data headers
`messy_mecca_kpp_initialize.f90`	initialization
`messy_mecca_kpp_integrator.f90`	numerical integration
`messy_mecca_kpp_jacobian.f90`	ODE Jacobian
`messy_mecca_kpp_jacobiansp.f90`	Jacobian sparsity
`messy_mecca_kpp_linearalgebra.f90`	sparse linear algebra
`messy_mecca_kpp_monitor.f90`	equation info
`messy_mecca_kpp_parameters.f90`	model parameters
`messy_mecca_kpp_precision.f90`	arithmetic precision
`messy_mecca_kpp_rates.f90`	user-defined rate laws
`messy_mecca_kpp_util.f90`	utility input-output
namelist files	
`caaba.nml`	CAABA namelist
`jval.nml`	JVAL namelist
`mecca.nml`	MECCA namelist

compiler, be sure to activate the C-preprocessor option. Values for `NETCDF_INCLUDE` and `NETCDF_LIB` are only needed for the model run when netCDF output is selected. To activate netCDF output, you also have to edit the `Makefile`:

- Change the variable `OUTPUT` from `ASCII` to `NETCDF`.
- Enter the correct netCDF library information in `NETCDF_INCLUDE` and `NETCDF_LIB`.

Should there be any problems with the CAABA/MECCA installation, please check the following:

- Confirm that all prerequisites (see above) are fulfilled!

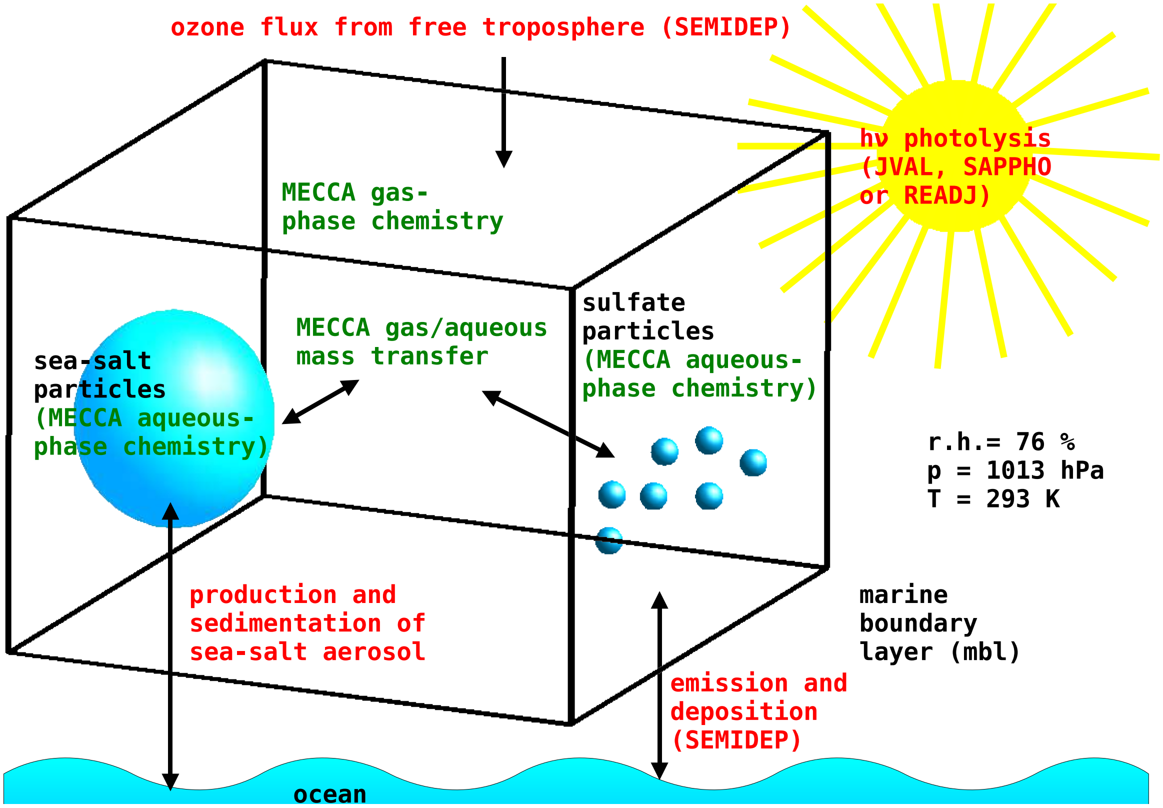
ozone flux from free troposphere (SEMIDEP)
hν photolysis (JVAL, SAPPHO or READJ)
MECCA gas-phase chemistry
MECCA gas/aqueous mass transfer
sea-salt particles (MECCA aqueous-phase chemistry)
sulfate particles (MECCA aqueous-phase chemistry)
r.h.= 76 %
p = 1013 hPa
T = 293 K
production and sedimentation of sea-salt aerosol
marine boundary layer (mbl)
emission and deposition (SEMIDEP)
ocean

Figure 2: The CAABA box model

- Confirm that the perl path in the first line of `sfmakedepend` is correct. It should be the same as the output of the command:
  `which perl`
- Confirm that the tcsh paths in the first lines of `xcaaba` and `xmecca` are correct.

3 Compiling and running the CAABA/MECCA box model with the shell script `xcaaba`

First, go to the directory of the model code:

```
cd caaba_2.5j
```

Now the tcsh script `xcaaba` guides you through the process of running the box model. To execute it, type:

```
./xcaaba
```

First, you are asked if you want to create a chemical mechanism with `xmecca`. If you answer “y”, you can select a new chemical mechanism with `xmecca` as described in detail in Sect. 5. However, for the first tests with CAABA/MECCA it is recommended to answer “n” and use the default mechanism, i.e. marine gas-phase and aerosol chemistry including halogens. Next, when asked for an option, choose “c” to compile the Fortran90 code. After a successful compilation, `xcaaba` asks if you want to run the CAABA boxmodel. After answering “y”, `xcaaba` lists the active contents of the `CAABA` and `CTRL` namelists which control the behaviour of CAABA/MECCA during run-time. The namelists are contained in the files `caaba.nml` and `mecca.nml`, and editing them allows fine-tuning of the model run (see Sect. 6.1). However, for the first tests, the defaults can be used as they are. Answer “Continue?” with “y”, and the model run will start. The model day and the current solar zenith angle (sza) are printed on the screen during the model run. The default is to integrate 8 days. After the model run, `xcaaba` asks if you want to save the model output inside the `output/` directory.

4 Plotting the model results with the ferret software

If you have chosen netCDF output, you can plot the model results with the ferret program (http://ferret.wrc.noaa.gov/Ferret/). Change into the jnl directory, then start the program by typing "ferret". When ferret has started, you can plot the gas-phase species of the latest model run with the ferret script xxxg.jnl by typing:

```
go xxxg.jnl
```

Similarly, xxxa.jnl can be used to plot aqueous-phase species:

```
go xxxa.jnl
```

The file xxxa.jnl accepts several parameters to modify the plots. The first parameter should be "0d" for plotting box model results. The second parameter can be set to "mpl" or "mpm" in order to plot either aqueous-phase concentrations [mol/L] or mixing ratios [mol(aq)/mol(air)], respectively. The third parameter defines the aerosol bin. With two aerosol bins, "A01" refers to sulfate particles, and "A02" to sea-salt particles. As an example, type:

```
go xxxa.jnl 0d mpl A01
```

Photolysis rate coefficients can be plotted with jval.jnl:

```
go jval.jnl
```

To plot results from previous runs which are saved in the output/ directory, edit the file setmodelrun.jnl and enter the paths of the directories in the "GO _define_sensi" command. To compare model runs, you can enter two or more "GO _define_sensi" commands in setmodelrun.jnl. To plot the difference between model runs, activate the line "DEFINE SYMBOL diffplot TRUE" in setmodelrun.jnl.

5 Selecting a chemical mechanism with the shell script xmecca

MECCA contains a very comprehensive set of chemical reactions in both the gas phase and the aqueous phase. For many applications, using the complete mechanism will consume too much CPU time. Therefore, the shell script xmecca has been written which allows to create a custom-made subset of the chemical mechanism interactively. Normally, xmecca is called via xcaaba. However, you can also start it manually:

```
cd mecca
./xmecca
```

xmecca will ask several questions, and recommended answers are given below. If you only press the Return key, you select the default.

```
How many aerosol phases? [0...99, default=0]
```

For a gas-phase only mechanism, type "0". For a mechanism with aqueous-phase chemistry in seasalt and in sulfate particles, type "2".

```
Modify gas.eqn with a replacement file?
[q/number, default=0]
```

Answer "0" unless you have written your own replacement file. More information about the replacement feature can be found in the file rpl/gas.rpl-example.

```
Choose a selection number or type a boolean
expression [q=quit, default=1]
```

Now you can choose a subset of chemical reactions. Not all possible selections have been tested thoroughly. Try one of these:

Troposphere, Minimum chemistry: A very small mechanism

Troposphere, Gas, no halogens: A gas-phase mechanism that includes NMHCs

EVAL: A mechanism that was used for the evaluation of the MECCA chemistry in the global model ECHAM5/MESSy.

Minimum MBL chemistry: A mechanism that contains aqueous-phase chemistry and should only be used if the number of aerosol phases is > 0.

For details about the selection, see Sect. 5.1.

```
Add diagnostic tracers to gas.eqn?
[q/0/?, default=0]
```

When this question shows up, answer "0".

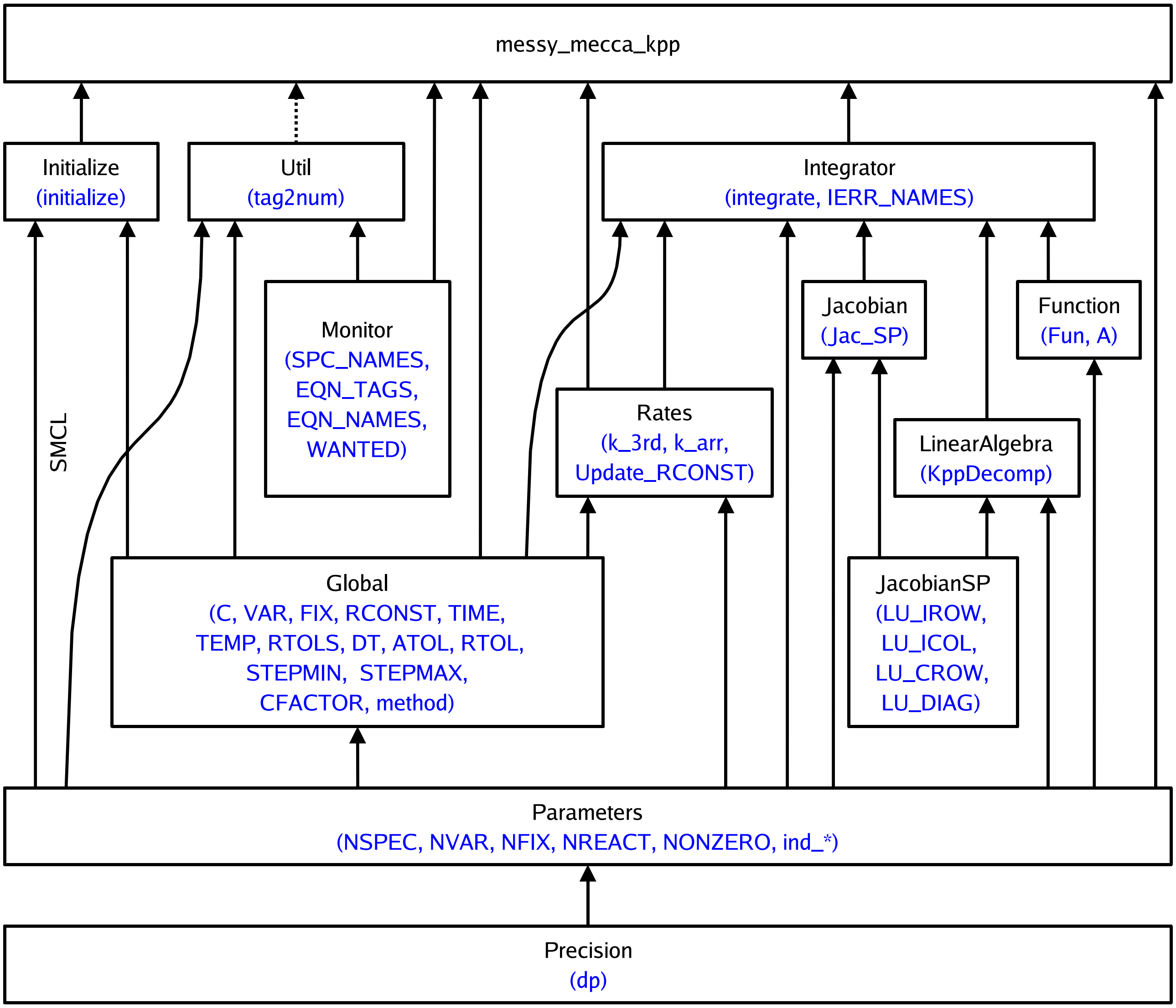
messy_mecca_kpp
Initialize
(initialize)
Util
(tag2num)
Integrator
(integrate, IERR_NAMES)
Monitor
(SPC_NAMES, EQN_TAGS, EQN_NAMES, WANTED)
Jacobian
(Jac_SP)
Function
(Fun, A)
Rates
(k_3rd, k_arr, Update_RCONST)
LinearAlgebra
(KppDecomp)
SMCL
Global
(C, VAR, FIX, RCONST, TIME, TEMP, RTOLS, DT, ATOL, RTOL, STEPMIN, STEPMAX, CFACTOR, method)
JacobianSP
(LU_IROW, LU_ICOL, LU_CROW, LU_DIAG)
Parameters
(NSPEC, NVAR, NFIX, NREACT, NONZERO, ind_*)
Precision
(dp)

Figure 3: Module structure of KPP-produced Fortran90 files. The arrows start at the module which is exporting the variables and subroutines shown in blue. They point to the module importing them via the Fortran90 USE instruction.

```
Add diagnostic tracers to all equations?
[y/n/q, default=n]
```

Answer “n”.

```
Tagging? [y/n/q, default=n]
```

Tagging is under construction, so please answer “n”.

```
Run KPP?
```

Answer “y”.

```
Choose an integrator [q=quit,
default=rosenbrock_posdef]:
```

The default integrator is strongly recommended (see Sect.5.2 for details). Next, KPP will create several Fortran90 files.

```
Remove indirect indexing with decomp?
[y/n/q, default=n]
```

If this question shows up, answer “n”.

```
Create LaTeX listing of selected mechanism?
[y/n/q, default=n]
```

If you answer “y” here, a table of the current reaction mechanism will be produced. Only the selected

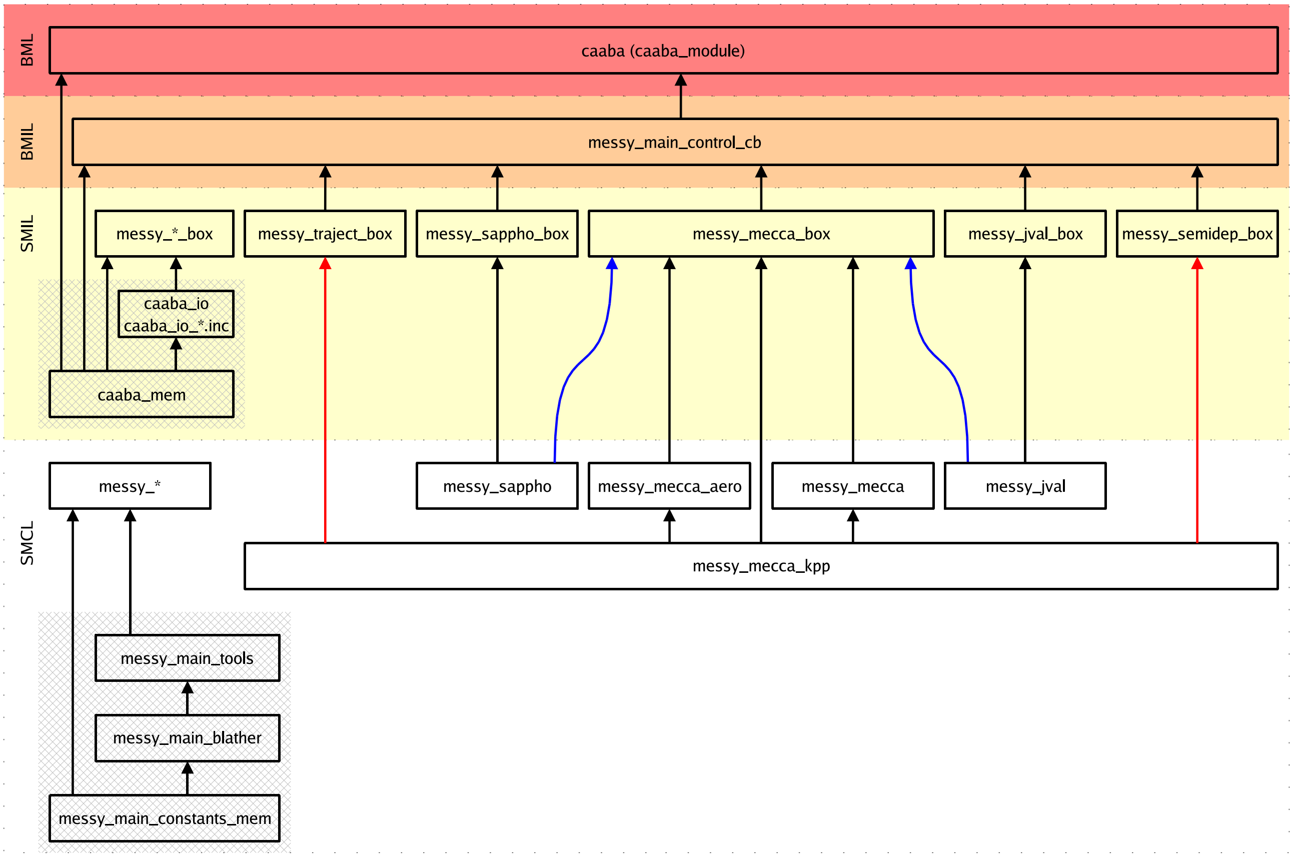

Figure 4: Module structure of MECCA when it is connected to the CAABA box model. The box-model related files are in the colored layers marked with BML, BMIL, and SMIL. The submodel core layer (SMCL) of MECCA is independent of the box model (see Jöckel et al. (2005) for details about the MESSy layers). The arrows start at the module which is exporting the variables and subroutines. They point to the module importing them via the Fortran90 USE instruction. Here, the box `messy_mecca_kpp` represents all KPP-generated files. The KPP-internal structure is shown in Fig. 3.

reactions will be listed. The table also contains the rate coefficients and their references, as described in Sect. 6.3.3.

```
Do you want to delete the temporary
xmecca files?
```

It is okay to delete these temporary files unless you need them for debugging purposes.

When `xmecca` finishes successfully, the Fortran90 code of your selected mechanism has been created. The KPP-produced Fortran90 files (Tab. 1) are moved into the `mecca/smcl/` directory (with lower-case names). An exception is `messy_mecca_kpp_Model.f90`, which is produced by KPP but not needed for MECCA. The modular structure of the KPP-produced Fortran90 files is shown in Fig. 3.

If you need to create a chemical mechanism very often, it is quite tedious to answer all questions every time. To make this easier, you can copy the template `batch/example.bat` to a new name (e.g. `batch/myfile.bat`) and then enter your answers into that file. Now you can create a new chemical mechanism in batch mode with

```
./xmecca myfile
```

5.1 Selecting a set of chemical reactions

All chemical reactions are marked. Each marker consists of several labels which contain information about the altitude (troposphere/stratosphere), the phase where the reaction occurs (gas/aqueous), its relevant chemical elements, and more. See Sect. 6.3.2 for a complete list of labels. To define a set of chemical reactions, you can either choose a pre-defined selection by number or enter a boolean expression based on the labels. Boolean expressions are typed in gawk syntax. The most important operators and expressions are:

`&&`	= AND
`||`	= OR
`!`	= NOT
`()`	= parentheses
`1`	= TRUE
`0`	= FALSE

For example, to select all gas-phase reactions (G) except for those including halogens (Cl, Br, I), type:

```
G && !Cl && !Br && !I.
```

It is important to understand the logic behind this selection mechanism. The expression "`Cl && Br`" selects only those reactions that contain chlorine *and* bromine. Similarly, the expression "`G && Het`" selects only those reactions that occur in the gas phase *and* and are heterogeneous. However, since no reaction has both the "`G`" *and* the "`Het`" label, this results in an empty mechanism. If you want a mechanism that contains both gas-phase and heterogeneous reactions, you must select all reactions that contain either the label "`G`" *or* the label "`Het`", i.e. you must use the expression "`G || Het`".

5.2 Selecting a numerical integrator

Several numerical integrators are defined in the subdirectory `mecca/kpp/int/` and can be used with KPP. The default is the positive definite Rosenbrock solver with automatic time-step control (`rosenbrock_posdef`). It is very robust and capable of integrating very stiff sets of equations (e.g. chemical mechanisms including both gas- and aqueous-phase chemistry). Although a Rosenbrock solver with manual time-step control (`ros2_manual`) is also available, it is strongly recommended not to use it for stiff sets of equations. If you choose it, you do so at your own risk!

6 Modifying CAABA/MECCA

The flow control of a CAABA/MECCA model simulation is shown in Fig: 5.

The CAABA/MECCA model simulation can be modified by changing the namelist files (`*.nml`), the species files (`*.spc`), the equation files (`*.eqn`) and the Fortran90 files (`*.f90`).

6.1 The namelist files `caaba.nml` and `mecca.nml`

The file `caaba.nml` contains the namelist `&CAABA`. Here individual parts of the CAABA model (the so-called "MESSy submodels") can be switched on or off. It is important that the following switches are set to "T" (=true):

```
USE_MECCA   = T
USE_SAPPHO  = T
USE_SEMIDEP = T
```

To use the photolysis rate coefficients from SAPPHO in MECCA, set:

```
photrat_channel = 'sappho'
```

Alternatively, you can switch on the JVAL submodel with `USE_JVAL = T` and then select `photrat_channel = 'jval'`. It is fine to switch on both the JVAL and the SAPPHO submodel, which can be useful for a comparison. However, only the values selected by `photrat_channel` are used for the MECCA chemistry.

You can define the model start, runtime, and time step. e.g.:

```
startday    = 90.
ext_runtime = '10 days'
time_step   = '15 minutes'
```

If you don't set these, the default is a model start on Julian day 80, a model run duration of 8 days, and an output time step of 20 minutes.

The file `mecca.nml` contains the namelists `&CTRL_KPP` and `&CTRL` (the namelist `&CPL` is not used in connection with CAABA). `&CTRL_KPP` is used for fine-tuning the numerical integration. The default selection `icntrl(3) = 2` should normally be suitable.

6.2 The species files `gas.spc` and `aqueous.spc`

The files `*.spc` declare chemical species for KPP. All species that may occur in an equation must be declared here. Additional dummy species may also be declared here.

Gas-phase species are declared in `gas.spc`. Examples for gas-phase species are `O2`, `O1D`, and `NO2`. The names of lumped species start with "`l_`".

MECCA also includes aqueous species which are declared in `aqueous.spc`. The names of cations end

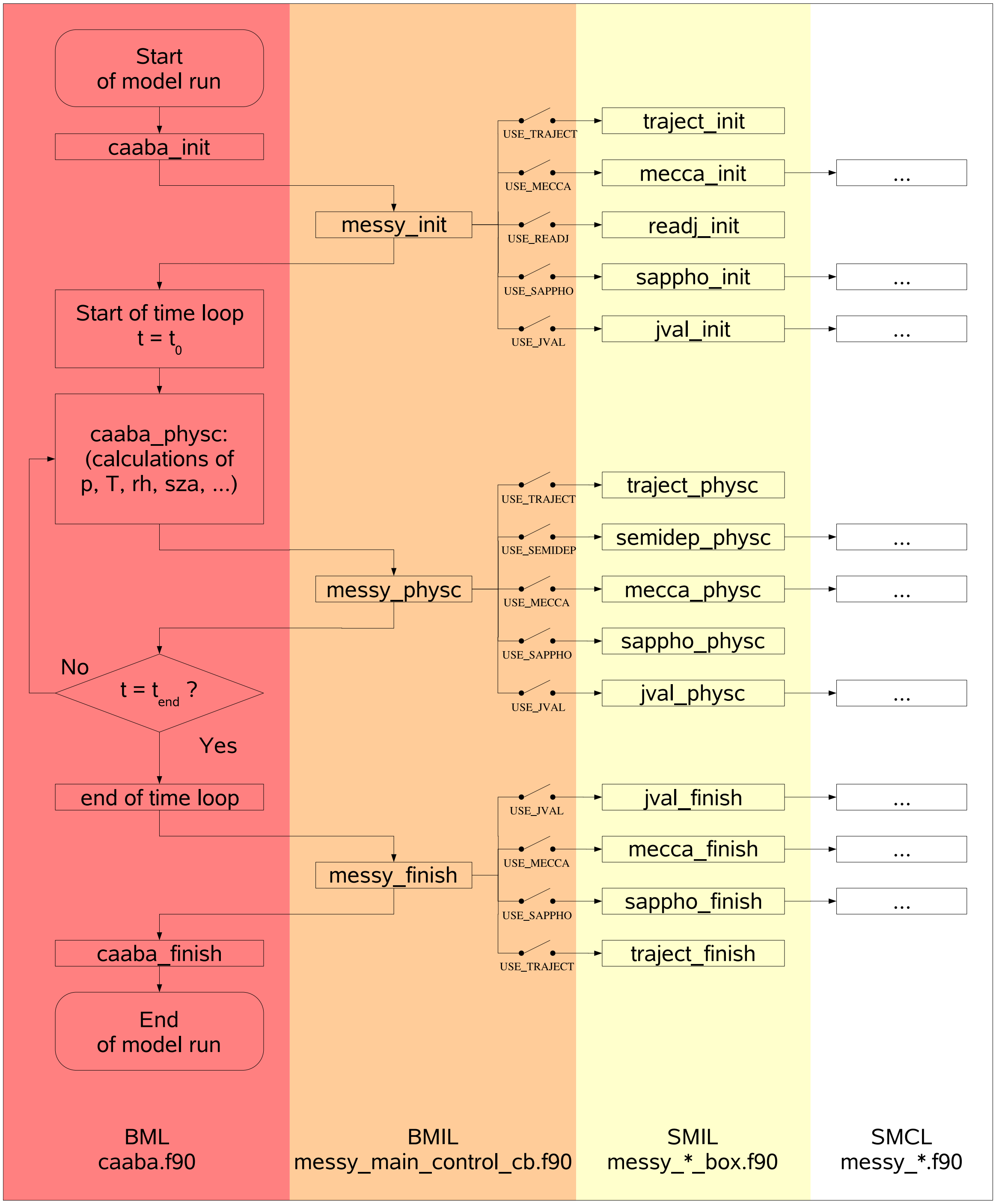

Figure 5: Flow control of a CAABA box model run

with “p” for plus. The names of single-charge anions end with “m” for minus. Doubly-charged anions end with “mm”. Examples for aqueous species are `H2O2`, `Hp`, `NO3m`, and `SO4mm`.

All aqueous-phase species have the suffix “_a##”, which is a placeholder for the aerosol phase number. `xmecca` replaces it by either “_a01” (accumulation soluble) or “_a02” (coarse soluble). This allows separate chemistry calculations for aerosol particles of different size and composition.

All species are defined here with `#DEFVAR`, i.e. KPP considers them as prognostic variables. To treat a species as a constant (e.g. CO_2), it can be added to the `#SETFIX` command in the file `messy_mecca_kpp.kpp`.

6.3 The equation files `gas.eqn` and `aqueous.eqn`

The equation files `*.eqn` define the chemical reaction mechanism for KPP. Each reaction occupies one line in this file. An example is:

```
<G1000> O2 + O1D = O3P + O2 : {%StTrG}
   3.3E-11*EXP(55./temp); {&1945}
```

The line starts with the reaction number, which is enclosed in angle brackets “`<...>`” (see Sect. 6.3.1). The second part (up to the colon) defines the reaction, and the third part (between the colon and the semicolon) defines the rate coefficient. The lines may also contain comments. Comments in equation files are either enclosed in curly braces, or the comment line starts with `//`. When using `xmecca`, some comments have a special meaning. Comments starting with the percent symbol “`{%...}`” are markers (see Sect. 6.3.2). Comments starting with the ampersand “`{&...}`”, the “at”-symbol “`{@...}`”, or the dollar “`{$...}`” are used to store information for the listing of reactions, as explained in Sect. 6.3.3.

If the definition of a rate coefficient is very complex, it can be stored in a Fortran90 variable and the variable is put into the `gas.eqn` file. For example, the rate of the self reaction of HO_2 is quite complex since it depends on humidity. It is predefined and the reaction line can be simplified to:

```
HO2 + HO2 = H2O2 : k_HO2_HO2;
```

The declaration and definition of `k_HO2_HO2` are also in the `gas.eqn` file. They can be found in the so-called KPP “inline types” `F90_GLOBAL` and `F90_RCONST`, e.g.:

```
#INLINE F90_GLOBAL
  REAL :: k_HO2_HO2
#ENDINLINE

#INLINE F90_RCONST
  k_HO2_HO2 = (1.5E-12*EXP(19./temp)+ &
              1.7E-33*EXP(1000./temp)*cair)* &
              (1.+1.4E-21* &
              EXP(2200./temp)*C(KPP_H2O))
#ENDINLINE
```

Another method to add reaction rates with complex dependencies are Fortran90 functions. This is done for example for the oxidation of S(IV) by H_2O_2 (`k_SIV_H2O2`). A function call is given as the rate in the `*.eqn` file. These functions are defined with the “inline type” `F90_RATES`:

```
#INLINE F90_RATES
  ELEMENTAL REAL(dp) FUNCTION k_SIV_H2O2 &
    (k_298,tdep,cHp,temp)
    ...
  END FUNCTION k_SIV_H2O2
#ENDINLINE
```

6.3.1 Reaction numbers

Each reaction in an equation file has a unique reaction “number” (number is not quite correct, since letters are included as well), which is enclosed in angle brackets, e.g.: “`<G1000>`”. The reaction number starts with one or more upper case letters denoting the type of reaction. The following types exist:

A	aqueous-phase reactions
H	Henry’s law (dissolution and evaporation)
EQ	equilibria in the aqueous phase (forward and backward reactions of acid/base and other equilibria)
G	gas-phase reactions
J	J-values of photolysis reactions
HET	heterogeneous reactions (e.g. on polar stratospheric clouds)

The type is followed by a sequence of 3 or 4 digits. The first digit is the number of the main element of the reaction. The following numbers are used:

1)	O	Oxygen
2)	H	Hydrogen
3)	N	Nitrogen
4)	C	Carbon
5)	F	Fluorine
6)	Cl	Chlorine
7)	Br	Bromine
8)	I	Iodine
9)	S	Sulfur
10)	Hg	Mercury

Out of those elements that occur in a reaction, the one with the highest number is called the main element. Accordingly, the second digit is determined by the element with the second highest number (or set to zero if there is no second element in the reaction). There is one exception in this numbering scheme: For the carbon group, the second digit is the number of C atoms in the largest organic molecule.

The following digits have no special meaning. If a reaction branches into several pathways, a suffix "a", "b", "c", … is added.

6.3.2 Markers and labels

Each reaction must contain a marker. A marker contains several labels. The syntax is "`{%...}`" where the dots represent the labels. Labels are used to select specific reactions, as described above (Sect. 5.1). The labels are placed in the marker without separators. The following labels are available and should appear in this order:

1. altitudes at which the reaction occurs (mandatory, include at least one)
   `St` = Reactions relevant in the stratosphere
   `Tr` = Reactions relevant in the troposphere
2. phase (mandatory, include exactly one)
   `Aa##` = Aqueous, aerosol (`##` is a placeholder for the 2-digit aerosol phase number)
   `G` = Gas phase reactions
   `Het` = heterogeneous reactions (e.g. on polar stratospheric clouds)
3. elements (include all elements that occur in the reaction, except for H and O)
   `N` = Nitrogen
   `C` = Carbon with $>$ 1 C atom (only used for C,N,O species but not for halogenated or sulfur-containing organics)
   `F` = Fluorine
   `Cl` = Chlorine
   `Br` = Bromine
   `I` = Iodine
   `S` = Sulfur
   `Hg` = Mercury
4. other
   `J` = Photolysis reactions
   `Mbl` = Minimum reaction mechanism for MBL chemistry
   `Sc` = Scavenging chemistry mechanism
   `Scm` = Scavenging chemistry mechanism, minimum selection

See Sect. 6.5.8 for a description how to add new labels to `xmecca`.

6.3.3 Creating a table of the chemical mechanism

To ensure that the documentation of the chemical mechanism is always up to date, the necessary information is contained inside the species and equation files. The gawk scripts `spc2tex.awk` and `eqn2tex.awk` convert information from the selected reactions into a LaTeX table. BibTeX citations are included in comments starting with an ampersand "`{&...}`". If there is a second ampersand "`{&&...}`", additional information about reactions can be found in `meccanism.tex` as a footnote to the tables. Comments starting with the at symbol "`{@...}`" or the dollar "`{$...}`" can be used to put LaTeX commands directly into the `*.eqn` files. `eqn2tex.awk` produces several `*.tex` files which are included into `meccanism.tex`.

Creating the document only works if you have the programs `latex`, `bibtex`, `dvips`, and `gv` installed on your system. If you want to use different programs (e.g. pdfLaTeX instead of LaTeX or a postscript viewer other than `gv`) you can enter the appropriate commands into `xmecca`.

6.4 Fortran90 files

The CAABA/MECCA simulations can be modified by changing the Fortran90 files. The modular structure of the Fortran90 files is shown in Fig. 4. Most of the files need only be changed by model developers. Those that are also interesting for model users, are briefly explained below.

6.4.1 `caaba.f90`

This file contains the main Fortran90 code ("`PROGRAM caaba`"). Here, the values of temperature (`temp`), pressure (`press`), relative humidity (`relhum`), and the height of the boundary layer (`zmbl`) can be defined.

6.4.2 `caaba_mem.f90`

This file contains variable declarations which are needed by several CAABA files.

6.4.3 `messy_main_control_cb.f90`

Flow control. Editing this file is only necessary when a new submodel is added.

6.4.4 `messy_jval_box.f90`

This file contains the connection of JVAL to CAABA.

6.4.5 messy_jval.f90

This file contains the calculation of J-values.

6.4.6 messy_mecca_box.f90

The chemical composition of seawater is defined in `SUBROUTINE mecca_init`. Aerosol properties (radius, liquid water content (LWC), and their chemical composition) are defined in `SUBROUTINE define_aerosol`. Initial mixing ratios of chemical species are defined in `SUBROUTINE x0`.

6.4.7 messy_sappho_box.f90

This file contains the connection of SAPPHO to CAABA.

6.4.8 messy_sappho.f90

Simplified parameterized photolysis rate coefficients are defined here.

6.4.9 messy_semidep_box.f90

Simplified emission fluxes and deposition velocities are defined here.

6.4.10 messy_mecca_aero.f90

Several variables needed to calculate rate coefficients are defined in `messy_mecca_aero.f90`. The accommodation coefficients (`alpha`) and the mean velocity (`vmean`) are used for the calculation of the mass transfer coefficients (`ykmt`). Together with the inverse dimensionless Henry's law coefficients (`yhenry`), they are needed to calculate equilibria between the gas and the aqueous phase. Heterogeneous reactions are described with the forward (`k_exf`) and backward (`k_exb`) rate coefficients. The variable `xaer` is set to 1 or 0 to switch aqueous-phase chemistry on or off, respectively. The factor `cvfac` converts the aqueous-phase unit mol/L (refering to the volume of the liquid) to the gas-phase unit molecules/cm^3 (referring to the gas-phase volume).

6.5 How to expand the chemical mechanism

This section contains brief descriptions for experienced model developers explaining where to make changes to the code for certain model expansions.

6.5.1 Adding a new gas-phase species

- `gas.spc`:
  Add the new species, its elemental composition, the name in LaTeX syntax, and a comment, e.g.:
  `C4H10 = 4C + 10H ; {@C_4H_<10>} {n-butane}`
  Note that curly brackets needed by LaTeX must be entered as angle brackets.

- `jnl/xxxg.jnl`:
  Add one line per new species. Check if the new species is part of an existing familiy, e.g. add new reactive bromine species to `Brx`.

- `jnl/tools/_kppvarg.jnl`:
  Add one line per new species.

6.5.2 Adding a new aqueous-phase species

- `aqueous.spc`:
  Add the new species, the name in LaTeX syntax, and a comment, e.g.:
  `SO4mm_a## = IGNORE; {@SO_4^<2->\aq} {sulfate}`
  The suffix `_a##` is mandatory. The elemental composition is currently ignored. Note that curly brackets needed by LaTeX must be entered as angle brackets.

- `jnl/xxxa.jnl`:
  Add one line per new species.

- `jnl/_families_a.jnl`:
  Check if the new species is part of an existing familiy, e.g. add new bromine species to `Brtot`.

- `jnl/tools/_kppvara.jnl`:
  Add one line per new species.

6.5.3 Adding a new gas-phase reaction

First, choose an appropriate reaction number. To avoid that several developers assign the same number to different new reactions, it is strongly recommended that a preliminary reaction number is used initially. This can be done by adding the developer's initials as a suffix, e.g. John Doe would use `G0001JD`, `G0002JD`, `G0003JD`, and so on. When the new code is merged with other development branches, the final reaction numbers will be assigned.

- `gas.eqn`:
  Add one line per new reaction.

- `latex/meccanism.tex`:
  If necessary, add a footnote about the new reaction here.

- `jnl/rates.jnl`:
  Add one line per new reaction.

6.5.4 Adding a new gas-phase photolysis reaction

First, choose an appropriate reaction number, as explained in Sect. 6.5.3.

- `gas.eqn`:
  Add one line per new reaction. Check that the necessary photolysis rate coefficient is provided by SAPPHO, READJ, and/or JVAL.

- `latex/meccanism.tex`:
  If necessary, add a footnote about the new reaction here.

- `jnl/rates.jnl`:
  Add one line per new reaction.

6.5.5 Adding a new aqueous-phase reaction

First, choose an appropriate reaction number, as explained in Sect. 6.5.3.

- `aqueous.eqn`:
  Add one line per new reaction.

- `latex/meccanism.tex`:
  If necessary, add a footnote about the new reaction here.

- `jnl/rates.jnl`:
  Add one line per new reaction.

6.5.6 Adding a new Henry's law equilibrium

First, choose an appropriate reaction number, as explained in Sect. 6.5.3.

- `aqueous.eqn`:
  Add two lines per new equilibrium, one for the forward and one for the backward reaction.

- `messy_mecca_aero.f90`:
  Add calculation of `vmean`, `alpha`, and `zhenry` which is needed for `k_exf` and `k_exb`.

- `latex/meccanism.tex`:
  If necessary, add a footnote about the new equilibrium here.

- `jnl/rates.jnl`:
  Add two lines per new equilibrium, one to define the net rate and one to plot it.

- `jnl/tools/_kppvarrates.jnl`:
  Add one line per new equilibrium.

6.5.7 Adding a new acid-base equilibrium

First, choose an appropriate reaction number, as explained in Sect. 6.5.3.

- `aqueous.eqn`:
  Add two lines per new equilibrium, one for the forward and one for the backward reaction.

- `latex/meccanism.tex`:
  If necessary, add a footnote about the new equilibrium here.

- `jnl/rates.jnl`:
  Add two lines per new equilibrium, one to define the net rate and one to plot it.

- `jnl/tools/_kppvarrates.jnl`:
  Add one line per new equilibrium.

6.5.8 Adding a new label

First, choose a name for the new label. The name must start with an upper case letter and can be followed by one or more lower case letters or numbers. Element symbols must not be used because they are reserved for reactions of that element. For example, since S is sulfur, the symbol S could not be used for the stratosphere. To avoid that several developers introduce new labels with the same name for different purposes, it is strongly recommended that a preliminary label is used initially. This can be done by adding the developer's initials as a prefix, e.g. John Doe would use `Jd1`, `Jd2`, `Jd3`, and so on. When the new code is merged with other development branches, a final label name can be assigned.

- `xmecca`:
  In the generation of `awkfile1`, add another `locate` function, and print the new label to the logfile.

6.5.9 Adding a new emission

- messy_mecca_semidep.f90:
  Add one line to emission_default (or one of the other emission_* subroutines).

6.5.10 Adding a new deposition

- messy_mecca_semidep.f90:
  Add one line to drydep.

6.6 How to add a new MESSy submodel

- Choose a name (up to 7 alphanumerical characters, starting with a letter). Here, "xyz" is used as an example.
- caaba_mem.f90:
  LOGICAL :: USE_XYZ = .FALSE.
- messy_xyz.f90:
  Put all generic subroutines here, i.e. all subroutines that are used for the CAABA boxmodel as well as for a global model.
- messy_xyz_box.f90:
  Put CAABA-specific code here. Generic code is not included directly here. Instead, the generic subroutines in messy_xyz.f90 are called from here. This file contains up to four subroutines:
  - If the submodel needs an initialization, put subroutine xyz_init here.
  - If the submodel performs calculations during the time loop, put subroutine xyz_physc here.
  - If the submodel prints results, put subroutine xyz_result here.
  - If the submodel needs to close any open files at the end of the model run, put subroutine xyz_finish here.
- messy_main_control_cb.f90:
  - Add "USE_XYZ" to "USE caaba_mem"
  - If subroutine xyz_init exists, add:
    USE messy_xyz_box, ONLY: xyz_init
    IF (USE_XYZ) CALL xyz_init
    to subroutine messy_init.
  - If subroutine xyz_physc exists, add:
    USE messy_xyz_box, ONLY: xyz_physc
    IF (USE_XYZ) CALL xyz_physc
    to subroutine messy_physc.
  - If subroutine xyz_result exists, add:
    USE messy_xyz_box, ONLY: xyz_result
    IF (USE_XYZ) CALL xyz_result
    to subroutine messy_result.
  - If subroutine xyz_finish exists, add:
    USE messy_xyz_box, ONLY: xyz_finish
    IF (USE_XYZ) CALL xyz_finish
    to subroutine messy_finish.
- caaba.f90:
  Edit subroutine caaba_read_nml:
  - Add "USE_XYZ" to "USE caaba_mem".
  - Add "USE_XYZ" to namelist /CAABA/.
  - Print value of USE_XYZ (see "selected MESSy submodels")
  - If applicable, perform consistency checks for interaction of new submodel with other submodels.
- nml/default/caaba.nml:
  Add sensible default values for USE_XYZ and possibly other options.
- manual/caaba_mecca_manual.tex:
  Mention new submodel in this user manual (Sect. 6.4, Tab. 1, and Fig. 5).

7 Revision history

The major changes between different CAABA/MECCA versions are listed here. An almost complete listing can be found in the file CHANGELOG.

7.1 New in version 2.5

The following major changes were made since version 2.4:

- Output of more information at the start of a model run.
- It is not necessary anymore to have the netcdf library for running CAABA/MECCA with ascii output. The corresponding output functions are now in the caaba_io* files.
- For clarity, different initializations and emissions are put into individual subroutines.
- xmecca-generated infos are now available as f90 strings in messy_mecca_kpp_global.f90.

7.2 New in version 2.4

The following major changes were made since version 2.3:

- `xmecca` runs in batch mode with `*.bat` files.
- Aerosol chemistry is switched on or off automatically (depending on the selected reaction mechanism) via `l_aero`.
- New directory structure: The former `boxmodel/` subdirectory is now the main CAABA directory. The MECCA code is now in the subdirectory `mecca/` in the main CAABA directory.
- The kpp program is now included in the `kpp/` directory in the CAABA/MECCA distribution.
- JVAL was added as a new submodel.
- `xmecca` skips parts that are only needed for global model applications when it is used in the CAABA distribution.

References

Jöckel, P., Sander, R., Kerkweg, A., Tost, H., and Lelieveld, J.: Technical Note: The Modular Earth Submodel System (MESSy) – a new approach towards Earth System Modeling, Atmos. Chem. Phys., 5, 433–444, 2005.

Jöckel, P., Tost, H., Pozzer, A., Brühl, C., Buchholz, J., Ganzeveld, L., Hoor, P., Kerkweg, A., Lawrence, M. G., Sander, R., Steil, B., Stiller, G., Tanarhte, M., Taraborrelli, D., van Aardenne, J., and Lelieveld, J.: The atmospheric chemistry general circulation model ECHAM5/MESSy1: consistent simulation of ozone from the surface to the mesosphere, Atmos. Chem. Phys., 6, 5067–5104, 2006.

Sander, R., Kerkweg, A., Jöckel, P., and Lelieveld, J.: Technical Note: The new comprehensive atmospheric chemistry module MECCA, Atmos. Chem. Phys., 5, 445–450, 2005.

Sandu, A. and Sander, R.: Technical Note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1, Atmos. Chem. Phys., 6, 187–195, 2006.