

# BoxMART User Manual

---

## Table of Contents

---

1. [Introduction](#)
2. [Prerequisites and Installation](#)
3. [Quick Start](#)
4. [Repository Layout](#)
5. [Building the Model](#)
6. [Running a Case](#)
7. [Input File Reference](#)
8. 7.1 [makefile.local](#)
9. 7.2 [boxmart.init — Namelist Reference](#)
10. 7.3 [input.csv — Meteorology](#)
11. 7.4 [initial.csv — Initial Concentrations](#)
12. 7.5 [specmat.csv — Species Property Matrix](#)
13. [Physical Modules](#)
14. 8.1 [Gas-Phase Chemistry \(KPP/MCM\)](#)
15. 8.2 [Aerosol Size Distribution](#)
16. 8.3 [Condensation / Evaporation](#)
17. 8.4 [Coagulation](#)
18. 8.5 [MOSAIC Thermodynamics](#)
19. [Adding a New Case](#)
20. [Modifying the Chemistry Scheme](#)
21. [Output and Post-Processing](#)
22. [Validation Cases](#)

---

## 1. Introduction

BoxMART (Box Model for Atmospheric Research and Teaching) is an open source atmospheric chemistry and aerosol model, developed for easy and flexible configuration for various simulation cases, as well as suitable to teach and learn complex chemistry and aerosol processes by including many benchmark cases.

Modelled processes include:

| Process                                  | Method   |
|--|--|
| Gas-phase chemistry                      | MCM reactions integrated by KPP  |
| Aerosol thermodynamics                   | MOSAIC model with $\text{H}^+$ , $\text{NH}_4^+$ , $\text{Na}^+$ , $\text{SO}_4^{2-}$ , $\text{NO}_3^-$ , $\text{Cl}^-$ and $\text{H}_2\text{O}$ |
| Condensation/Evaporation/<br>Dissolution | APC or APD schemes   |
| Coagulation                              | Sectional coagulation  |
| Photolysis                               | Solar zenith angle + actinic flux calculation  |

## 2. Prerequisites and Installation

### 2.1 System Requirements

| Dependency          | Minimum version | Notes   |
|---------------------|-----------------|---|
| gfortran            | 9+              | Other Fortran 90 compilers may work with minor makefile edits                               |
| NetCDF-Fortran      | 4.x             | Headers in <code>/usr/local/include</code> , libs in <code>/usr/local/lib</code> by default |
| KPP                 | 2.x             | For generating a chemistry scheme   |
| Python 3            | 3.8+            | Only for post-processing  |
| netCDF4<br>(Python) | —               | <code>pip install netCDF4</code> , only for post-processing                                 |

### 2.2 Installing KPP

Follow the official KPP documentation to install KPP on your system: [https://kpp.readthedocs.io/en/stable/getting\\_started/01\\_installation.html](https://kpp.readthedocs.io/en/stable/getting_started/01_installation.html)

After installation, verify that the `kpp` executable is in your `PATH`:

```
kpp --version
```

### 2.3 Cloning the Repository

Request the access right from Putian Zhou (putian.zhou@helsinki.fi). Then clone from the git repository.

```
git clone git@version.helsinki.fi:putian.zhou/boxmart.git
cd boxmart
```

The directory is self-contained; no further installation step is needed.

### 3. Quick Start

This example builds and runs the `J1994_CASE5` benchmark.

```
# 1. Compile the model for J1994_CASE5
make IN=runs/J1994_CASE5/makefile.local

# 2. Run the simulation
cd runs/J1994_CASE5
./boxmart.exe boxmart.init

# 3. Inspect output (if you have installed NetCDF)
ncdump -h output/output.nc

# 4. (Optional) plot results
cd ../../
python plot/J1994_CASE5.py
```

Expected runtime on a modern laptop: a few seconds for `J1994_CASE5`.

### 4. Repository Layout

```
boxmart/
├── makefile           Main build file – do not edit unless adding source files
├── src/              Fortran source modules
│   ├── main.f90      Main time loop; reads init file, calls sub-models
│   ├── aerosol_mod.f90 Size distribution, condensation, coagulation interfaces
│   ├── io_mod.f90    NetCDF / ASCII I/O
│   ├── parameter_mod.f90 Physical constants, species property arrays
│   ├── time_mod.f90  Time management
│   ├── mosaic/       MOSAIC aerosol thermodynamics
│   ├── thermodynamic_mod.f90 [NOT USED NOW]
│   ├── iso_thermodynamic_mod.f90 [NOT USED NOW]
│   ├── png_mod.f90   [NOT USED NOW]
│   └── thermo_param_mod.f90 [NOT USED NOW]
├── chemistry/
│   ├── mcm_and_kpp/  Shared chemistry definitions for all cases
│   │   ├── second.kpp KPP master configuration
│   │   ├── chem_mcm.def MCM gas-phase reaction definitions
│   │   ├── chem_mosaic.def MOSAIC aerosol chemistry definitions
│   │   └── mcm_module.f90 Interface between KPP output and BoxMART
│   └── <CASE_NAME>/ Per-case KPP-generated Fortran files (second_*.f90)
```

|                     |  |
|---------------------|--|
| ├─ runs/            | One subdirectory per simulation case               |
| ├─ <CASE_NAME>/     |  |
| │ └─ makefile.local | Case-specific compiler variables                   |
| │ └─ boxmart.init   | Fortran namelist: process switches, I/O, time      |
| │ └─ input/         |  |
| │ │ └─ input.csv    | Meteorology input                                  |
| │ │ └─ initial.csv  | Initial gas and aerosol concentrations             |
| │ │ └─ specmat.csv  | Species properties                                 |
| │ └─ output/        |  |
| │ └─ output.nc      | NetCDF output (created at runtime)                 |
| ├─ plot/            | Python post-processing scripts                     |
| │ └─ local.py       | Shared utilities                                   |
| │ └─ batch_run.py   | Batch parameter sweep runner                       |
| │ └─ <CASE>.py      | Case-specific plotting scripts                     |
| ├─ lib/             | Compiled object files (.o, .mod) – created by make |
| ├─ template/        | Template case files for creating new cases         |
| └─ doc/             | Documentation                                      |

## 5. Building the Model

### 5.1 Compile a Case

All `make` commands require the `IN` variable pointing to a case's `makefile.local`:

```
make IN=runs/<CASE_NAME>/makefile.local
```

This compiles:

1. MOSAIC ( `src/mosaic/` ) → `src/mosaic/lib/`
2. Chemistry ( `chemistry/<CHEM_SUBDIR>/` ) → `chemistry/<CHEM_SUBDIR>/lib/`
3. Main source ( `src/` ) → `lib/`
4. Links everything into `runs/<CASE_NAME>/<EXE>` (default `boxmart.exe`)

### 5.2 Regenerate the Chemistry Scheme (KPP)

Only needed when `.def` or `.kpp` files change:

```
make kppgen IN=runs/<CASE_NAME>/makefile.local
```

This runs KPP on `chemistry/<CHEM_SUBDIR>/second.kpp` and overwrites the `second_*.f90` files in that directory.

## 5.3 Cleaning Build Artifacts

```
make clean      # Remove lib/ objects (main source)
make cleantest  # Remove chemistry/*/lib/ objects
make cleanall   # Remove everything (lib/, chemistry libs, MOSAIC lib)
```

## 5.4 Build Modes

The default build mode is `OPT_DEBUG` (bounds checking, all warnings). To change it, edit `makefile` around line 74:

| Mode                      | Flag   | Use case                            |
|---------------------------|--|-------------------------------------|
| <code>OPT_DEBUG</code>    | <code>-g -O0 -fcheck=all -fbounds-check</code> | Development and debugging (default) |
| <code>OPT_OPTIMIZE</code> | <code>-O1</code>                               | Normal production runs              |
| <code>OPT_PROF</code>     | <code>-O2 -pg</code>                           | Profiling with gprof                |

Or pass extra flags through `OPT_EXTRA` in `makefile.local`:

```
OPT_EXTRA := -O1
```

## 6. Running a Case

```
cd runs/<CASE_NAME>/
./boxmart.exe boxmart.init
```

The executable must be run from inside the case directory so that relative paths in `boxmart.init` resolve correctly. The init file name (here `boxmart.init`) is passed as the only command-line argument.

Progress is printed to stdout according to `dt_display`. Output is written to `output/output.nc` (NetCDF, default) at intervals of `dt_output`.

## 7. Input File Reference

### 7.1 `makefile.local`

Located at `runs/<CASE_NAME>/makefile.local`. Sets variables consumed by the main `makefile`.

| Variable       | Required | Example                             | Description                                     |
|----------------|----------|-------------------------------------|---|
| CASE_SUBDIR    | Yes      | J1994_CASE5                         | Subdirectory under runs/ for this case          |
| EXE            | Yes      | boxmart.exe                         | Executable filename                             |
| CHEM_SUBDIR    | Yes      | J1994_CASE5                         | Subdirectory under chemistry/ to use            |
| INCLUDE_NETCDF | Yes      | -I/usr/local/include                | NetCDF Fortran include path                     |
| LIB_NETCDF     | Yes      | -L/usr/local/lib -lnetcdff -lnetcdf | NetCDF link flags                               |
| OPT_EXTRA      | No       | -O1                                 | Extra compiler flags appended to the build mode |

Minimal example:

```

CASE_SUBDIR = J1994_CASE5
EXE         = boxmart.exe
CHEM_SUBDIR = J1994_CASE5
INCLUDE_NETCDF = -I/usr/local/include
LIB_NETCDF   = -L/usr/local/lib -lnetcdff -lnetcdf

```

## 7.2 boxmart.init — Namelist Reference

The init file contains Fortran namelists. Any parameter not specified takes the default value listed below.

### &NML\_CTRL — Process Switches

| Parameter                 | Type    | Default | Description                          |
|---------------------------|---------|---------|--------------------------------------|
| l_use_chemistry           | logical | .true.  | Enable gas-phase chemistry (KPP/MCM) |
| l_use_explicit_cs         | logical | .false. | Use explicit condensation sinks      |
| l_use_aerosol             | logical | .true.  | Enable aerosol module                |
| l_use_condensation        | logical | .true.  | Enable condensation                  |
| l_use_condensation_growth | logical | .true.  | Enable condensation growth           |
| l_use_thermodynamic       | logical | .true.  | Enable MOSAIC aerosol thermodynamics |
| l_use_coagulation         | logical | .true.  | Enable coagulation                   |
| l_apply_kelvin            | logical | .true.  | Apply Kelvin (curvature) effect      |

| Parameter                   | Type    | Default              | Description                             |
|-----------------------------|---------|----------------------|---|
| <code>l_apply_raoult</code> | logical | <code>.true.</code>  | Apply Raoult's law (solute effect)      |
| <code>l_debug</code>        | logical | <code>.false.</code> | Print extra diagnostic output to stdout |

### &NML\_TIME — Time Control

| Parameter               | Type         | Default | Description                                  |
|-------------------------|--------------|---------|--|
| <code>time_start</code> | real(dp)     | —       | Simulation start time [s]                    |
| <code>time_end</code>   | real(dp)     | —       | Simulation end time [s]                      |
| <code>date_start</code> | real(dp) × 6 | —       | Start date: year, month, day, hour, min, sec |
| <code>dt</code>         | real(dp)     | —       | Model time step [s]                          |
| <code>dt_display</code> | real(dp)     | —       | Console output interval [s]                  |
| <code>dt_output</code>  | real(dp)     | —       | File output interval [s]                     |

Example:

```
&NML_TIME
  time_start = 0.0d0,
  time_end   = 43200.0d0,           ! 12 hours
  date_start = 2021.0d0, 10.0d0, 12.0d0, 0.0d0, 0.0d0, 0.0d0,
  dt         = 60.0d0,
  dt_display = 600.0d0,
  dt_output  = 300.0d0,
/
```

### &NML\_IO — Input / Output Paths

| Parameter                       | Type   | Default                    | Description                                 |
|---------------------------------|--------|----------------------------|---|
| <code>root_fdir</code>          | string | —                          | Absolute path to the BoxMART root directory |
| <code>input_fdir</code>         | string | <code>'./input'</code>     | Directory containing input CSV files        |
| <code>input_fname</code>        | string | <code>'input.csv'</code>   | Meteorology file name                       |
| <code>specmat_fname</code>      | string | <code>'specmat.csv'</code> | Species property matrix file name           |
| <code>initial_fname</code>      | string | <code>'initial.csv'</code> | Initial concentrations file name            |
| <code>output_fdir</code>        | string | <code>'./output'</code>    | Directory for output files                  |
| <code>output_fname_ncf</code>   | string | <code>'output.nc'</code>   | NetCDF output file name                     |
| <code>output_fname_meteo</code> | string | <code>'meteo.dat'</code>   | ASCII meteorology output (deprecated)       |

| Parameter         | Type    | Default    | Description   |
|-------------------|---------|------------|---|
| output_fname_chem | string  | 'chem.dat' | ASCII chemistry output (deprecated)   |
| input_spc_line    | string  | ''         | Comma-separated species to read from input.csv (deprecated, use the column in specmat.csv file) |
| output_spc_line   | string  | ''         | Comma-separated species to write to output (deprecated, use the column in specmat.csv file)     |
| l_output_ascii    | logical | .false.    | Write ASCII output (deprecated; use NetCDF)   |

### &NML\_AEROSOL — Aerosol Size Distribution Initialisation

| Parameter                    | Type            | Default | Description   |
|------------------------------|-----------------|---------|---|
| nbins                        | integer         | 10      | Number of particle size bins                          |
| l_init_bins                  | integer         | 1       | Method for defining bin boundaries (see below)        |
| input_matrix_init_bins       | real(dp) matrix | varies  | Parameters for the chosen bin method                  |
| l_init_nsd                   | integer         | 1       | Method for the initial number size distribution (NSD) |
| input_matrix_init_nsd        | real(dp) matrix | varies  | Parameters for the chosen NSD method                  |
| l_init_msd                   | integer         | 1       | Method for the initial mass size distribution         |
| l_kmt_method                 | integer         | 1       | Mass-transfer coefficient method                      |
| l_coagk_method               | integer         | 1       | Coagulation kernel method                             |
| l_use_wet_dp_in_coag         | logical         | .true.  | Use wet diameter in coagulation                       |
| l_condensation_growth_method | integer         | —       | Condensation growth algorithm (1 = default)           |
| p_condensation_growth_method | real(dp) array  | —       | Parameters for the condensation growth method         |

#### Bin boundary methods ( l\_init\_bins ):

| Value | Description                                     | Required input_matrix_init_bins parameters |
|-------|---|--|
| 1     | Log-uniform from Dmin to Dmax                   | row 1: (Dmin, Dmax)                        |
| 2     | Linear-uniform from Dmin to Dmax                | row 2: (Dmin, Dmax)                        |
| 3     | Explicit boundaries, nbins+1 values             | row 3: all boundary diameters [m]          |
| 4     | Explicit midpoints, nbins values                | row 4: all midpoint diameters [m]          |
| 5     | Volume-ratio grid                               | row 5: (Dmin, vrat1, n_vrat1, vrat2)       |
| 6     | Two-regime volume-ratio grid                    | row 6: (Dmin, vrat1, nbins1, vrat2)        |
| 7     | Geometric series from Dmin with ratio and nmono | row 7: (Dmin, nmono, vrat_max)             |
| 8     | Fixed pivot at a given diameter                 | row 8: (D_pivot, vrat_max)                 |

### Number size distribution methods ( l\_init\_nsd ):

| Value | Description                                  | Required input_matrix_init_nsd parameters              |
|-------|--|--|
| 1     | Input directly from input matrix             | row 1: Number concentration in each bin                |
| 2     | Uniform total number + monodisperse diameter | row 2: (Ntot, Dp)                                      |
| 3     | Sum of lognormal modes                       | row 3: (nmodes, N1, Dp1, sigma1, N2, Dp2, sigma2, ...) |

### &NML\_OTHER — Geographical Location

| Parameter | Type     | Default | Description              |
|-----------|----------|---------|--------------------------|
| lat_deg   | real(dp) | 40.0    | Latitude [degrees north] |
| lon_deg   | real(dp) | 100.0   | Longitude [degrees east] |

### 7.3 input.csv — Meteorology

Time-varying meteorological conditions. The file format has four fixed header lines followed by data rows. Values are linearly interpolated in time.

```
time,temp,RH,pres,Rglob
s,K,%,atm,W_m-2
1,1,1,1,1
0,0,0,0,0
```

```
0,298.15,70.0,1,400
86400,298.15,70.0,1,400
```

### Header lines:

| Line | Content                                     |
|------|---|
| 1    | Column names                                |
| 2    | Units                                       |
| 3    | Unit conversion factors (1 = no conversion) |
| 4    | Offsets (0 = no offset)                     |

### Columns:

| Column | Unit  | Description                     |
|--------|-------|---------------------------------|
| time   | s     | Time since simulation start     |
| temp   | K     | Air temperature                 |
| RH     | %     | Relative humidity (0-100)       |
| pres   | atm   | Air pressure                    |
| Rglob  | W m-2 | Incoming global solar radiation |

## 7.4 initial.csv — Initial Concentrations

Sets initial gas-phase and aerosol mass concentrations.

```
name,gas,bin001:019,bin020:040,...
-,ug_m-3,m%,m%,...
H2SO4,10.0d0,0,0,...
```

### Header lines:

| Line | Content  |
|------|--|
| 1    | Column names: name , gas , then bin group names (e.g. bin001:019 ) |
| 2    | Units: ug_m-3 for gas; m% (mass fraction) for aerosol bins         |

### Columns:

| Column | Description                                     |
|--------|---|
| name   | Species name (must match names in specmat.csv ) |

| Column      | Description                                       |
|-------------|---|
| gas         | Initial gas-phase concentration                   |
| bin columns | Initial aerosol mass fraction per bin group [0-1] |

Bin groups are ranges of bins sharing the same initial mass fraction. The fractions for all species in a given bin must sum to 1.

## 7.5 `specmat.csv` — Species Property Matrix

Defines physical and chemical properties for each species. The model reads this file at startup. Column order is fixed.

```
sname,xmm,density,phase,cond,Keq_A,Keq_B,Keq_C,n_C,...,sat_A,sat_B,alpha,Hcp,l_input,l_output,l
-,kg_mol-3,kg_m-3,-,-,...
H2SO4,0.0981,1800,2,5,...
```

### Column reference:

| Column       | Unit                                   | Description   |
|--------------|--|---|
| sname        | —                                      | Species name (must match KPP <code>SPC_NAMES</code> )           |
| xmm          | kg mol <sup>-1</sup>                   | Molar mass  |
| density      | kg m <sup>-3</sup>                     | Species density in particle phase                               |
| phase        | —                                      | Phase code (see below)  |
| cond         | —                                      | Condensation method code (see below)                            |
| Keq_A        | mol kg <sup>-1</sup> atm <sup>-1</sup> | Equilibrium constant pre-exponential                            |
| Keq_B        | —                                      | Keq temperature coefficient B                                   |
| Keq_C        | —                                      | Keq temperature coefficient C                                   |
| n_C ... n_Mg | —                                      | Atom counts: C, H, O, S, N, Cl, Br, I, F, Na, Fe, Ca, Mg        |
| nring_aro    | —                                      | Number of aromatic rings (used in diffusion volume)             |
| nring_cyc    | —                                      | Number of heterocyclic rings                                    |
| sat_A        | —                                      | Antoine equation parameter A: $\log_{10}(\text{svp}) = A - B/T$ |
| sat_B        | K                                      | Antoine equation parameter B                                    |
| alpha        | —                                      | Mass accommodation coefficient                                  |
| Hcp          | mol L <sup>-1</sup> atm <sup>-1</sup>  | Henry's law constant  |
| l_input      | —                                      | Read initial concentration from <code>initial.csv</code> (0/1)  |

| Column                 | Unit | Description                |
|------------------------|------|----------------------------|
| <code>l_output</code>  | —    | Write to output file (0/1) |
| <code>l_display</code> | —    | Print to console (0/1)     |

### Phase codes:

| Code | Phase   |
|------|---|
| 1    | Unknown   |
| 2    | Gas   |
| 3    | Undissociated aqueous species (name suffix <code>_aq</code> ) |
| 4    | Cation in aqueous phase (name suffix <code>_pp...</code> )    |
| 5    | Anion in aqueous phase (name suffix <code>_mm...</code> )     |
| 6    | Solid phase (name suffix <code>_s</code> )                    |

### Condensation codes:

| Code | Method  |
|------|---|
| 1    | Aqueous or solid phase already in particle phase (not applicable) |
| 2    | Not condensable   |
| 3    | APC (Analytical Predictor of Condensation) scheme                 |
| 4    | APD (Analytical Predictor of Dissolution) scheme                  |
| 5    | Other method  |

## 8. Physical Modules

### 8.1 Gas-Phase Chemistry (KPP/MCM)

**Source:** `chemistry/mcm_and_kpp/`, `chemistry/<CASE>/second_*.f90`

KPP auto-generates Fortran code from `.kpp` and `.def` files. The generated `second_*.f90` files implement the stiff ODE solver (Rosenbrock) for the gas-phase reaction network. The interface module `chemistry_mod.f90` / `mcm_module.f90` bridges the KPP state vector to BoxMART's internal species arrays.

**Enabled by:** `l_use_chemistry = .true.`

The KPP configuration in `second.kpp` uses: - Integrator: Rosenbrock - Jacobian: sparse LU  
- Language: Fortran 90, double precision

Photolysis rates ( `J_values` ) are computed from the solar zenith angle derived from the geographic location ( `lat_deg` , `lon_deg` ) and `date_start` .

## 8.2 Aerosol Size Distribution

**Source:** `src/aerosol_mod.f90`

The aerosol is represented as a sectional size distribution with `nbins` fixed dry-diameter bins. For each bin, the model tracks:

- Number concentration [# m-3]
- Dry and wet diameter [m]
- Per-species mass, molar, and number concentrations
- Dry and wet density and volume
- Species molality (for aqueous-phase species)

The size grid is defined at startup via `l_init_bins` / `input_matrix_init_bins` in `&NML_AEROSOL` . The initial number size distribution is set via `l_init_nsd` .

Default particle densities applied when no explicit density is given: - Organic: 1400 kg m-3 - Inorganic: 1800 kg m-3 - General particle: 1400 kg m-3

## 8.3 Condensation / Evaporation / Dissolution

**Source:** `src/aerosol_mod.f90` , `src/main.f90`

**Enabled by:** `l_use_condensation` and the growth enabled by `l_use_condensation_growth`

Two condensation schemes are available:

- **APC** (Analytical Predictor of Condensation, `cond = 1` ): applies to species that condense without equilibrium constraints.
- **APD** (Analytical Predictor of Dissolution, `cond = 2` ): for species that dissolve into the aqueous phase.

The Kelvin effect (curvature correction) and Raoult's law (solute effect) can be toggled independently:

```
l_apply_kelvin = .true.  
l_apply_raoult = .true.
```

The condensation growth method is selected by `l_condensation_growth_method` with method-specific parameters in `p_condensation_growth_method` .

## 8.4 Coagulation

**Source:** `src/aerosol_mod.f90` , `src/main.f90`

**Enabled by:** `l_use_coagulation = .true.`

The sectional coagulation is calculated. The coagulation kernel method is set by `l_coagk_method` (default 1 = Brownian). Set `l_use_wet_dp_in_coag = .true.` to use the wet diameter when computing the coagulation kernel.

## 8.5 MOSAIC Thermodynamics

**Source:** `src/mosaic/`

**Enabled by:** `l_use_thermodynamic = .true.`

MOSAIC (Model for Simulating Aerosol Interactions and Chemistry) handles inorganic aerosol thermodynamic equilibrium: partitioning of H<sub>2</sub>SO<sub>4</sub>, HNO<sub>3</sub>, HCl, NH<sub>3</sub>, Na, and their aqueous/solid counterparts between gas and particle phases, water uptake, and ion activity coefficients.

---

## 9. Adding a New Case

1. **Copy a template** that is close to your intended setup:

```
bash cp -r runs/base runs/<YOUR_CASE>
```

1. **Edit** `makefile.local` : set `CASE_SUBDIR`, `EXE`, and `CHEM_SUBDIR`.
2. **Edit** `boxmart.init` : adjust time settings, process flags, and I/O paths. Set `root_fdir` to the absolute path of your BoxMART installation.
3. **Prepare input files** in `runs/<YOUR_CASE>/input/` :
  4. `input.csv` — meteorology time series
  5. `initial.csv` — initial gas and aerosol concentrations
  6. `specmat.csv` — species properties (can be copied from a similar case)
7. **If reusing an existing chemistry scheme:** point `CHEM_SUBDIR` at the directory that already has compiled `second_*.f90` files (no `kppgen` needed).
8. **Build and run:**

```
bash make IN=runs/<YOUR_CASE>/makefile.local cd runs/<YOUR_CASE> ./boxmart.exe  
boxmart.init
```

---

## 10. Modifying the Chemistry Scheme

The shared chemistry definitions are in `chemistry/mcm_and_kpp/` :

| File                    | Purpose                                      |
|-------------------------|--|
| <code>second.kpp</code> | KPP configuration (solver, language, driver) |

| File                         | Purpose   |
|------------------------------|---|
| <code>chem_mcm.def</code>    | MCM gas-phase reactions and species   |
| <code>chem_mosaic.def</code> | MOSAIC aerosol chemistry definitions  |
| <code>second.def</code>      | Includes <code>chem_mcm.def</code> and <code>chem_mosaic.def</code> , and your chemical reactions and species |
| <code>mcm_module.f90</code>  | BoxMART-KPP interface   |

To create a case-specific chemistry scheme:

1. Create a new chemistry sub-folder.
2. Copy a `second.def` file from other chemistry sub-folder to your chemistry sub-folder, add or remove reactions and species.
3. Regenerate the KPP Fortran source:

```
bash make kppgen IN=runs/<YOUR_CASE>/makefile.local
```

1. Add any new species to `specmat.csv` and `initial.csv`.
2. Rebuild: `make IN=runs/<YOUR_CASE>/makefile.local`

**Important:** Species names in `specmat.csv` must exactly match the names defined in `SPC_NAMES` within the KPP-generated `second_monitor.f90`.

## 11. Output and Post-Processing

### 11.1 NetCDF Output

The primary output file is `output/output.nc` (NetCDF file). It contains:

- Time axis (seconds since simulation start)
- Meteorological variables: temperature, RH, pressure, solar zenith angle
- Gas-phase concentrations for species
- Particle-phase concentrations for species per size bin
- Aerosol number and mass concentrations per size bin

Use `ncdump -h output/output.nc` to inspect variable names and dimensions.

### 11.2 Python Post-Processing

Post-processing scripts are in `plot/`. The utility library `plot/local.py` provides functions for reading NetCDF output, converting units, and generating figures with `matplotlib`.

Run a case-specific script:

```
python J1994_CASE5.py
```

For batch runs with parameter sets, use `plot/batch_run.py`, which can launch multiple BoxMART runs with varying input parameters and collect results.

---

## 12. Validation Cases

---

| Case directory     | Reference                 | Description                                       |
|--------------------|---------------------------|---|
| J1997_APC_TEST     | Jacobson (1997)           | APC condensation scheme                           |
| J1997_APD_TEST     | Jacobson (1997)           | APD dissolution scheme                            |
| J1994_CASE1/23/4/5 | Jacobson et al. (1994)    | Coagulation                                       |
| JT1995_CASE1/2/3   | Jacobson and Turco (1995) | Condensation and coagulation, Kelvin effect       |
| Z2008_CASE1–12     | Zaveri et al. (2008)      | Aerosol thermodynamics                            |
| Z2008_CASE13/14    | Zaveri et al. (2008)      | Aerosol thermodynamics, condensation, coagulation |