Geoscientific
Model Development

Supplement of

GenChem v1.0 – a chemical pre-processing and testing system for atmospheric modelling

David Simpson et al.

Correspondence to: David Simpson (david.simpson@met.no)

S1. extra_mechanisms used in boxChem and EMEP

The *extra_mechanisms* system allows for a large number of extra chemical packages to be added to a given base mechanism, in a mix n' match system. Thus, one may include, exclude, or change various reaction and emissions schemes, for example related to SOA or PM. Tables S1–S2 give a brief summary of the main extra mechanisms available in the GenChem v1.0 release, as used in boxChem and EMEP CTM modelling. Extra information on each package can usually be found in the README.md files accompanying the Species and Reactions files of each mechanism.

S2. User-defined chemical schemes

Adding and modifying chemical mechanisms is rather easy in the GenChem system. We take the example of a gender-neutral user Jan, who wants to add a gas-phase chemistry *JansChem* for EMEP usage. The steps needed are as follows.

S2.1 Own chemical mechanisms

Jan is of course free to add new mechanisms to **base_mechanisms** and **extra_mechanisms**. Just follow the formatting guidelines and examples from the currently available mechanisms. Start testing from a temporary boxChem directory, following Sect. 3, which in our example means:

Example S1.1.   `scripts/box_setup.py tmp_work`

Then, from tmp_work:

Example S1.2.   `./do.testChems JansChem`

Once the code compiles and runs fine with do.testChems, the next step for those wanting to run the EMEP model is to modify emep_setup.py, as described below.

S2.2 Modification of emep_setup.py

Jan can then edit *emep_setup.py* (maybe renaming it as *Jans_setup.py*, but we retain *emep_setup.py* below). If selecting from the provided base_mechanisms and extra_mechanisms you only need to extend the possible command lines as provided by the **cmdx** dictionary in that script.

The -b argument gives the base mechanism (only one allowed), and then the -e argument allows the addition of any number of compatible extra mechanisms. Any keys from *cmdx* can be used by *emep_setup.py*, e.g. if our user Jan has their own gas-phase chemical scheme (*JansChem*) one could edit *emep_setup.py* a new option::

Example S1.3.   `cmdx['JansChem'] ='-b JansChem -e common'`

Jan could then do::

Example S1.4.   `emep_setup.py JansChem`

Example S1.4 creates the directory ZCM_JansChem with all the files needed for EMEP runs. These files can then be transferred directly to the EMEP system in the same way as is done for the pre-defined mechanisms (Sects. 3, 4). Note that *emep_setup.py* has a number of differently defined verions of 'common' with regard to extra mechanisms; the user should assess carefully which is needed.

S2.3 Modification of emissplit files

As noted in Sect. 6.4, the default emissplit files for sox, nox, co and nh3 are identical across all provided schemes, and provided in the input/emissplit_defaults directory as files such as emissplit_defaults_nox.csv. For NMVOC and PM inventories specific files are needed for each chemical mechanism, and sometimes depending on available inventories. In these emissplit files, splits are given for each of the 11 SNAP sectors, so that for example SNAP1 (power generation) has very different splits from SNAP7 (road traffic).

In the EMEP CTM these default speciations are usually replaced by country-specific values derived from more recent emission inventories, but for input to boxChem a much simpler procedure is used. For boxChem these SNAP splits are merged into one NMVOC speciation using the relative emissions for each SNAP given by average UK emission profiles from Passant (2002) and emissions from 2010. The procedure and SNAP fractions used can be found in the Fortran module *EmisGet_mod.f90*. In principle the user could modify the numbers used in *EmisGet_mod.f90*, but since boxChem users just need an overall NMVOC speciation the simplest approach is to give each SNAP sector the same NMVOC split. Then the weighted NMVOC split will also be given these values.

Thus, a user-configured emissplit file can be used, ideally stored with a different name and in a different directory to the boxChem defaults. The directory with these new emissplit files can be set in config_box.nml.

Table S1. extra_mechanism packages provided by GenChem 1.0 for boxChem

BoxAero	Simple reactions for SO_2 oxidation and gas-aerosol uptake of N2O5 and HNO3.
BoxDep	Simple deposition reactions, for HNO3, SO2, O3, and NO2
BVOC_IsoMT1_emis	Emissions of isoprene and α-pinene†

Notes:

† This simple package is provided for both boxChem and EMEP since all base-mechanisms can handle isoprene and α-pinene chemistry, and often α-pinene is used as a surrogate for all monoterpenes. For use of more complex BVOC mixtures (e.g. with β-pinene, limonene, etc), the user may replace this package with a more complex emission package.

Table S2: extra_mechanism packages provided by GenChem 1.0 for EMEP CTM modelling

General extras

Aero2017nx	Adds aerosol uptake[a] of N2O5, NO3, HNO3, HO2, H2O2 and CH3OOH
Ash	Volcanic ash tracers (fine and coarse)
Aqueous_EmChem16x	Gas and aqueous phase reactions converting SO2 to SO4
DustExtended	Emissions of Dust_f and Dust_c
PM_FFireInert	Emissions of inert species ffire_OM, ffire_BC and ffire_remPPM25
PM_WoodFFuelInert	Emissions of inert POM, EC and other components
Pollen	Emissions of pollen[b] from birch, olive, ragweed and grass
SeaSalt	Emissions of SeaSalt_f and SeaSalt_c
ShipNOx	Adds 'shipNOx' compound and its reactions for dealing with NOx plumes from shipping[c]

Biogenic VOC (BVOC) options

BVOC_IsoMT1_emis	Emissions of C5H8 and α-pinene[d]
BVOC_IsoMT2_emis	Emis. C5H8, α-pinene, and β-pinene, can be used with CRIv2R5Em
BVOC_IsoMT3_emis	Emis. C5H8, α- and β-pinene, and other monoterpenes (XTERP), can be used with extended EmChem19 or CRIv2R5Em schemes
BVOC_XTERP_CRI	Reactions of "other monoterpenes" (XTERP) for use with CRIr2V5Em (optional)
BVOC_SQT_NV	Adds simple sesquiterpene emissions — treated as immediately transformed into non-volatile (particulate) secondary organic aerosol (SQT_SOA_NV)
BVOC_ExtraMTs	Reactions of β-pinene and XTERP for use with EmChem19a if BVOC_IsoMT3_emis is used

Secondary Organic Aerosol, SOA-associated extras

PM_VBS_EmChem19	EMEP standard volatility basis set (VBS) SOA reactions for EmChem19-like chemistry[e] (can also be used with CRIv2R5Em); includes ASOA/BSOA from α-pinene surrogate
PM_VBS_ExtraMTs	VBS BSOA reactions for (extended) EmChem19 and CRIv2R5Em-like schemes, for β-pinene and XTERP; simple extension of [e] with additional monoterpenes

(continued on next page)

Table S2 cont.

PM_VBS_CB6r2Em	ASOA/BSOA reactions for CB6r2Em; adaption of PM_VBS_EmChem19 to CB6r2Em
PM_Hodzic_EmChem19	SOA formation reactions for EmChem19-like chemistry[f]
PM_Hodzic_Aromatics	ASOA formation from aromatics, for CRIv2R5Em scheme[f]
PM_Hodzic_BPINENE	BSOA formation from β-pinene, for extended EmChem19a (using BVOC_ExtraMTs) or CRI schemes[f]
PM_Hodzic_XTERP	BSOA formation from XTERP monoterpene surrogate, for extended EmChem19a (using BVOC_ExtraMTs) or extended CRI schemes (using BVOC_XTERP_CRI)[f]
PM_Hodzic_CB6	ASOA and BSOA formation reactions for CB6r2Em[f]
PM_JPAC_MT3	Semi-empirical BSOA formation reaction for C5H8, α- and β-pinene, and XTERP, as used in McFiggans et al. 2019 (combined with PM_Hodzic_Aromatics)

Notes: [a] c.f. Stadtler et al. (2018); Simpson et al. (2018); [b] c.f. Sofiev et al. 2017; [c] Simpson et al. (2015); [d] As in boxChem, Table S1. [e] Default is 'NPAS' VBS scheme used, see Sect. 5, Simpson et al. (2012); [f] Loosely based upon VBS system by Hodzic et al. (2016).

S3. Time-steps for boxChem simulations

The config_box.nml file allows for a user-defined 'external' time-step, dt, which is typically set at 30 seconds. This time-step governs the frequency at which emissions, photolysis, and rate-coefficients are updated. The chemical solver (ChemSolver) is called once per dt time-step, but then ChemSolver implements between 5–15 sub-timesteps. First, 5 very short time-steps (between 0.2 to 1 s, depending on dt) are used to bring the chemistry into rapid equilibrium, and then 5–10 longer timesteps are used for the remainder of dt.

Using calculations with dt=1 s as a reference (which gives an internal dif time-step dt_i of 0.2 s), we have calculated the relative root mean square error, $RRMS_k$, for several key species k for dt ranging from 3 s to 600 s. $RRMS_k$ is defined as:

$$RRMS_k = \sqrt{\frac{\sum_{n=1}^{N}(c_k^n(1) - c_k^n(dt))^2}{\sum_{n=1}^{N}(c_k^n(1))^2}}$$

where $c_k^n(dt)$ is the calculated concentration of species k at hour n with timestep dt (in seconds). In the examples presented here, we use a 24h (N=24) simulation, starting at noon.

Table S3. Calculated $RRMS_k$ values for a 24 hour simulation of EmChem19a, and external time-steps (dt) of between 1–600 s

Species	$c(1)^{(a)}$ (ppb)	$RRMS_k$ (in %) values for dt=							
		3	10	30	60	120	240	450	600
O3	3.473e+01	0.0021	0.0099	0.0326	0.0668	0.1362	0.2807	0.5570	0.7760
OH	2.077e-04	0.0043	0.0200	0.0600	0.1081	0.1793	0.2858	0.4957	0.6904
HO2	8.751e-03	0.0047	0.0235	0.0766	0.1504	0.2851	0.5360	0.9966	1.3778
NO	2.544e-01	0.0115	0.0556	0.1788	0.3463	0.6285	1.0537	1.7075	2.2788
NO2	1.911e+00	0.0027	0.0135	0.0452	0.0905	0.1758	0.3418	0.6613	0.9219
PAN	2.427e-01	0.0037	0.0175	0.0581	0.1194	0.2449	0.5112	1.0369	1.4660
NO3	5.235e-03	0.0041	0.0180	0.0571	0.1151	0.2298	0.4574	0.8673	1.1756
N2O5	1.046e-02	0.0028	0.0123	0.0396	0.0798	0.1572	0.2993	0.5334	0.7095
CPU$^{(b)}$:	10.17	3.38	2.02	1.00	0.52	0.25	0.13	0.07	0.05

Notes: (a) mean concentrations calculated for dt =1 s; (b) CPU time relative to dt =30 s run.

Tables S3–S4 illustrate the changes in $RRMS_k$ (in percent) with dt for some key species with EmChem19a and MCMv3.3Em systems. For our typical dt of 30 s, $RRMS_k$ are seen to be largest for NO and N2O5, but are ≤0.2% for those species in both schemes. Even with dt of 120 s $RRMS_k$ values don't exceed 1%. For dt=600 s the $RRMS_k$ are about 1–2% for EmChem19a, but much larger for the MCM scheme. Users may of course experiment with these choices, but the default dt=30 s seems adequate for normal boxChem usage.

Table S3 also illustrates the CPU required (relative to the dt=30 s case) for these runs (as calculated on a desktop PC, x86_64 linux processor, gfortran 5.4). The 30 s case is seen to require about 20 times more CPU than dt=600 s, but this corresponds to only 4 s real time. The dt=1 s case requires nearly 10 times more CPU than the 30 s, but for EmChem19a this is only 40 s real time. For MCMv3.3Em the CPU used is far greater, ca. 40 h for dt=1 s, ca. 4 h for dt=30 s, and ca. 1 hour for dt=120 s. (In future the MCM scheme might be made much more CPU efficient by omission of non-used VOC and halogen precursors.)

Table S4. Calculated $RRMS_k$ values for a 24 hour simulation with MCMv3.3Em, and external time-steps (dt) of between 1–600 s

Species	c(1) (ppb)	$RRMS_k$ (in %) values for dt=							
		3	10	30	60	120	240	450	600
Species	1	3	10	30	60	120	240	450	600
O3	3.488e+01	0.0021	0.0097	0.0324	0.0675	0.1436	1.7375	0.5737	0.7631
OH	1.938e-04	0.0041	0.0191	0.0570	0.1026	0.1730	2.8830	3.2349	3.6909
HO2	7.948e-03	0.0051	0.0246	0.0802	0.1608	0.3289	7.4843	7.0996	8.2689
NO	2.546e-01	0.0115	0.0552	0.1774	0.3445	0.6351	4.5669	1.6685	3.3062
NO2	1.902e+00	0.0026	0.0133	0.0445	0.0900	0.1818	1.5367	0.5717	0.7002
PAN	2.337e-01	0.0039	0.0185	0.0624	0.1348	0.3027	8.8077	4.7753	4.4970
NO3	4.568e-03	0.0038	0.0171	0.0542	0.1113	0.2277	0.5133	0.6467	0.6832
N2O5	9.001e-03	0.0027	0.0119	0.0385	0.0780	0.1544	0.3114	0.4404	0.5226
CPU:	10.03	3.34	2.00	1.00	0.50	0.25	0.12	0.07	0.05

Notes: as Table S3