Supplement of Geosci. Model Dev., 14, 4555–4572, 2021
https://doi.org/10.5194/gmd-14-4555-2021-supplement

Supplement of

urbanChemFoam 1.0: large-eddy simulation of non-stationary chemical transport of traffic emissions in an idealized street canyon

Edward C. Chan and Timothy M. Butler

Correspondence to: Edward C. Chan (edward.chan@iass-potsdam.de)

Contents

S1: Compilation Instructions for `urbanChemFoam` and related components

`urbanChemFoam` is a weakly compressible, reactive solver developed based on the open-source finite-volume computational continuum mechanics framework OpenFOAM (Weller et al., 1998). Additional components have been introduced in conjunction to provide characteristics of a chemical transport model for an infinite street canyon. These are:

`dateTime` :	Associating simulation time with physical time based on Julian Day
`solarUtils` :	Calculating solar state based on geographical information and physical time

`irreversiblephotolysisReaction` :	Implementation of photolysis reactions for the OpenFOAM core thermophysics module
`simpleEmission` :	Enables emission boundary conditions to be prescribed in units of mass instead of mass mixing ratio
`cyclicFixedValue` :	Allows scalars to be prescribed as a Dirichlet condition on a cyclic boundary
`cyclicZeroGradient` :	Allows scalars to be prescribed as a Neumann condition on a cyclic boundary
`initCanyon` :	Utility to initialize velocity field in an infinite street canyon domain with perturbation

`urbanChemFoam` and all related components, described in this document or otherwise, are licensed under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or, at your option, any later version. A copy of the GNU General Public License can be found along with the distribution of `urbanChemFoam`. If not, please refer to the URL `http://www.gnu.org/licenses/`.

These modules can be built as user applications and libraries upon an existing functional installation of OpenFOAM version 7, maintained by CFD Direct Ltd.. Instructions to install OpenFOAM version 7, `urbanChemFoam`, and all related components are presented below. It is assumed that the user will compile said source locally on a Linux system using GCC version 4.8 and above. In addition to the GCC version requirement, a version of MPI library compiled with the same version of GCC toolchain for OpenFOAM is also requisite. In the case of this study, it is version 4.0.1 of OpenMPI (default MPI library for OpenFOAM) compiled with GCC 9.2.0:

1) Download the source packages for OpenFOAM version 7 and accompanying third-party tools from the OpenFOAM Foundation website

    a. The URL at the time of writing is `https://openfoam.org/download/7-source/`

    b. Some `urbanChemFoam` components have been known to work only with OpenFOAM version 7. The reaction class interface has been modified in this version and is incompatible with previous versions of OpenFOAM. In addition, there are irreconcilable implementation differences in basic data structures between OpenFOAM from CFD Direct (the ".org" URL) and that from ESI (the ".com" URL) that `urbanChemFoam` will fail to compile in the ESI release

2) Follow the instructions for installing the OpenFOAM 7 framework, first by installing the third-party tools, followed by the OpenFOAM core, with the following notes

    a. Compilation of third-party library CGAL and its dependencies is not required

    b. Compilation of third-party application Paraview and its dependencies is not necessary as Paraview version 5.4.0 or higher can also be installed independently, for instance, from a binary package

3) Upon successful installation of OpenFOAM (this will take a while), inspect the paths reported by following environment variables to ensure they are correct

    a. `WM_PROJECT_INST` (location of the OpenFOAM framework)
    b. `WM_PROJECT_DIR` (location of the OpenFOAM core)
    c. `WM_PROJECT_USER` (location of the OpenFOAM user-defined modules)
    d. `WM_THIRD_PARTY_DIR` (location of the OpenFOAM third-party components)

4) Source the script `${WM_PROJECT_DIR}/etc/bashrc` before using any OpenFOAM solvers or components

5) Unpack the tarball `urbanChemFoam-1.0.tar.gz` and component source codes into the location indicated by `${WM_PROJECT_USER}`. Create the directory if it does not exist. Adjust any contents in the directory as necessary should the path in question be not empty

6) Link or copy the following files or directory from directory `${WM_PROJECT_USER}/src/` to corresponding locations in directory `${WM_PROJECT_DIR}/src/`

   a. `makePhotolysisReactions.C` in `thermophysicalModels/specie/reaction/reactions`
   b. `photolysisReactionRate` in `thermophysicalModels/specie/reaction/reactionRate`

7) Link the files below in `${WM_PROJECT_DIR}/src/thermopyhsicalModels/specie/lnInclude`

   a. `${WM_PROJECT_USER}/src/common/commonGlobals.H`
   b. `${WM_PROJECT_DIR}/src/ … /photolysisReactionRate/photolysisReactionRate.H`
   c. `${WM_PROJECT_DIR}/src/ … /photolysisReactionRate/photolysisReactionRateI.H`

8) Execute the script `Allwmake` in directory `${WM_PROJECT_DIR}/src/thermophysicalModels` to update the OpenFOAM core thermophysics module with photolysis reactions

9) Execute the script `Allwmake` in directory `${WM_PROJECT_USER}` to build the remaining components and utilities for `urbanChemFoam`

10) Verify that a `platform` directory has been created in `${WM_PROJECT_USER}` after the build, and `urbanChemFoam`, `initCanyon`, as well as all other components are located in the sub-directories (e.g., `platform/linux64GccDPInt32Opt/bin` and `platform/linux64GccDPInt32Opt/lib`)

S2: Workflow for executing accompanied cases

The following accompanied cases for a core mesh size of 0.5 m (fine mesh) are provided:

1) `stationary-57600` : Stationary model run for 16:00 UTC (57600s)
2) `transient` : Transient model run for a 24-hour period

Both cases begin with a six-hour (21600 s) spin-up period; they are contained in the `spinup` directory for the stationary case, and the `day00` directory in the transient case. In the stationary case, the model rub for the sampling period is contained in the `sample` directory, while, in the transient case, this is in the `day01` directory. The transient run has an additional `day02` directory, which performs an additional hour of simulation (from 00:00 to 01:00 UTC) to cover a full 24-hour period of central hourly average with a 3600 s window.

Grid generation takes place prior to the spin-up run, and the same grid is used throughout the remaining stages of each model run. On the other hand, domain decomposition (almost certainly necessary given the size of the problem) is required for each stage of the run, as field data are also decomposed along with mesh data. Therefore, field data obtained at the end of each stage are recombined from the parallel domains and served as initial conditions for the proceeding stage. Each parallel run could be executed from the console or submitted to a queue manager.

Each directory represents a separate run following the same general workflow:

1) Go to the spin-up directory (`spinup` or `day00`)

2) Generate mesh by executing the OpenFOAM application `blockMesh`

3) Initialize perturbed flow field with the application `initCanyon`

4) Perform domain decomposition using the OpenFOAM utility `decomposePar`

5) Execute model run with the command `mpirun –np [NP] urbanChemFoam –parallel`

   a. The number of processors, `NP`, must match the number of decomposed domains specified in OpenFOAM dictionary file `system/decomposeParDict`

6) Prepare field data for the proceeding run phase by using the script `prepNextDay.sh`

    a. A numerical directory will be created for the end-of-spin-up field data (`21600` for the stationary run, and `86400` for the transient run). Due to double precision storage of time the field data directory might not be written exactly as shown; simply rename the directory to remove the fractional part.

7) Switch to the sampling run directory (`sample` or `day01`)

8) Check that the `0` directory (initial conditions) is linked correctly to the previous end-of-run field data (`../sample/21600` or `../day00/86400`)

    a. Note step 6a, if the previous end-of-run field data is not exactly as shown

9) Repeat Steps (4) and (5) for the sampling run.

The stationary model run should be complete at the conclusion of this step. The following steps apply to the transient model run.

10) Repeat Step (6) to generate end-of-run field data

11) Switch to trailing run directory `day02`

12) Repeat Steps (8) and (9) to complete the trailing data

Field data for all output time steps in the sampling run can be reconstructed using the OpenFOAM utility `reconstructPar`, which must be converted into VTK format using the OpenFOAM utility `foamToVTK` in order to be accessed in Paraview. Line sample data (for instance, for the vertical and spanwise horizontal stations defined in Section 4) can be found stored in text format in the `postProcessing` directory.

S3: Thermophysical Properties of Gas Phase Species

Thermophysical properties, such as specific heat (c_p) and dynamic viscosity (μ), can be specified in OpenFOAM as temperature-dependent parameterizations, which are tabulated as polynomial coefficients – a_i in Equation (23) – in McBride et al. (1993). Additional coefficients are provided as integration constants for enthalpy and entropy are not required in this study. Sutherland coefficients – A_S and T_S in Equation (24) – are calculated by fitting the Sutherland law expression on the polynomial expression over the temperature range between 250 K and 350 K. These values for the species used in the present study are presented in Table S1 to five significant digits.

Table S1: Coefficients of specific heat and dynamic viscosity for c_p and μ (McBride et al, 1993).

	a_0	a_1	a_2	a_3	a_4	A_S	T_S
N_2	3.2987	1.4082E-3	-3.9632E-6	5.6415E-9	-2.4449E-12	1.4217E-6	111.76
NO	4.2185	-4.6390E-3	1.1041E-5	-9.3361E-9	2.8036E-12	1.5797E-6	127.27
NO_2	3.9440	-1.5854E-3	1.6658E-5	-2.0475E-8	7.8351E-12	1.6909E-6	232.18
$O^3(P)$	3.1683	-3.2793E-3	6.6431E-6	-6.1281E-9	2.1127E-12	1.3967E-6	52.494
O_2	3.7824	-2.9967E-3	9.8473E-6	-9.6813E-9	3.2437E-12	1.6993E-6	128.23
O_3	3.4074	2.0538E-3	1.3849E-5	-2.2331E-8	9.7607E-12	7.7760E-7	2.9479

References

McBride et al. (1993) NASA Tech Memo 4513.

Weller et al. (1998) Comput Phys 12:620-631.