ION and Charged Aerosol Growth Enhancement (ION-CAGE) model

Jacob Svensmark

September 11, 2019

1 Introduction

The ION-Charged Aerosol Growth Enhancement (ION-CAGE) v0.9 code constitutes a time-integration 0-dimensional model that evolves an aerosol distribution through time while taking charge into account. The model takes its initial and environmental conditions from an input-file, and outputs snapshots with specified time intervals.

2 File overview

Files for running the model, all coded in FORTRAN include:

- `main.f90`: The top level program i.e. the main program for the model. It is here the Runge Kutta time integration loop of the General Dynamics Equation is found.
- `dNdt_module.f90`: Module containing subroutines for the calculation of coagulation, condensation, ion-induced condensation, nucleation, charge exchange and loss terms as controlled in the `controlfile.txt`. Called from main.f90.
- `io_ioncage_module.f90`: Subroutines related to I/O for the model. `read_coef_matrix` and `read_coef_twocolumn` are used to read interaction coefficients from a file (see instructions in the source code to use your own tables). `read_controlfile` reads all parameters set in controlfile and converts to SI units. `print_controlfile_to_output` is used to print the contents of the used controlfile into the output. `print_snapshot` prints a snapshot of the simulation to the output. Finally `load_state` loads the latest snapshot of an output data from a previous run, and continues simulation from that state if corresponding FLAG is set in `controlfile.txt`.
- `math_module.f90`: This modile contains two interpolation subroutines `linear_interpolation` and `cubic_interpolation_2d` used in the process of loading interaction coefficients from the `read_coef_twocolumn` and

`read_coef_matrix` subroutines. Both will return an error if grid is out of the interpolation range. Therefore, remember to check if diameter range of the model is compatible with (contained in) the range of diameters provided in these files.

- `io_general.f90`: Contains `file_lines` which counts the number of lines in a text file, and `findnl` which locates newline character in a string.
- `precision_type.f90`: FORTRAN module specifying precision of floats.
- `controlfile.txt` Inputfile controlling parameters and other types of setup for the model at runtime. See below for in-depth explanation of the controlfile contents.

Also included are a number of interaction coefficient tables for condensation and coagulation. Coagulation coefficient tables are provided for aerosols of density 1200 $\mathrm{kg\,m^3}$. Condensation coefficients are provided for four different masses of monomers, namely 100 AMU, 150 AMU, 225 AMU and 350 AMU, which can be selected in the controlfile.

3 Compiling and running the model

In order to run the ION-CAGE model the source code inside the `f90` subdirectory should be compiled using the `Makefile`:

```
> make
```

This produces the executable `ioncage.out`. Note that this has only been tested using the g95 FORTRAN compiler from OS X. If other compilers are desired, the `Makefile` should be adjusted accordingly. If successful the program may be run by typing the following

```
> ./ioncage.out > outputfilename.dat
```

where `outputfilename.dat` is the data output file of the run (the compiler may produce some warnings that can be ignored). Note that the run is based on the input given in the `controlfile.txt` at runtime.

4 Input parameters and inputfile

Most of the input for the model can be controlled by the `controlfile.txt` file, and it must be placed in the same directory as the compiled executable file produced by the `Makefile`. The `controlfile.txt` is split up into a number of paragraphs:

- In the **Total number of particle nodes** top paragraph of the `controlfile.txt`, the number of volume nodes can be set, i.e. the number of log-spaced aerosol sizes.

- The second paragraph of `controlfile.txt` it is possible to switch on (FLAG=1) and off (FLAG=0) different modules i.e. different terms of the general dynamic equation that the code runs. Each module corresponds to contributions to dN/dt via different mechanisms (loss, coagulation etc), and the code for each module can be found in the `dNdt_module.f90` file.
- The **Continue from previous run** paragraph is concerned with the possibility of taking an end-state of a previous run as initial conditions for the current run. The name of the previous run data-file should be provided, and the FLAG set to 1 to actually use it.
- In the **Parameters** paragraph several parameters may be set for the course of the run. First the nucleation rate of small aerosols in neutral or charged state. Note that the nucleation of of a charged aerosol removes an ion of the same charge. Then the production rate for monomers (neutral and charged), the ion-ion recombination coefficient, neutral monomer density and ion densities, loss terms, masses (should match the used interaction coefficients).
- In the **Initial concentrations** paragraph initial ion and sulfuric acid concentrations are set. The `fix_initials` keyword has three flags separated by comma, controlling whether the initial concentrations of the positive ions, negative ions and sulfuric acid are kept constant throughout the entire run (in that order). Initial values are overwritten if `Load_file_flag`=1, however if the fix `fix_initials` are set to 1, the choice of fixed ions and sulfuric acid are not overwritten. Also the initial values may be kept constant by setting the `fix_initials` to 1.
- The **Time related parameters** control start time (usually 0), `End time` i.e. length of run, step length Δt for the integration and the time between each snapshot printed to the output file.
- The **Size related parameters** paragraph contains parameters controlling the size range of the nodes in the simulation, as well as critical diameters for neutral and charged nucleated aerosols.
- Finally, the **Interaction Coefficients** paragraph contains a list of paths to datafiles with interaction coefficients that are loaded for each type of interaction used in the GDE. (See subroutines `read_coeff_twocolumn` and `read_coeff_matrix` in file `io_ioncage_module.f90` for an explanation on the format assumed for the coefficient data files). For each file, a number of header lines to skip before reading data can be specified. **NOTE:** It is important to be aware exactly how these are used in the `main.f90` and `dNdt_module.f90`, as many different approaches are possible here.

Each parameter has a dedicated keyword followed by a value, which should be self explanatory. Lines that do not start with one of these keywords are ignored. Note that the executable does **not** need to be recompiled to register changes in `controlfile.txt`.

5 Units

Unless otherwise stated, all calculations internally and all output are in standard SI units. Note that several input parameters in the `controlfile.txt` are not necessarily SI units, but converted to SI by the `read_controlfile` subroutine.

6 CPU-Time vs. Precision

The model run speed varies significantly depending on the calculation. The number of nodes and temporal resolution can be adjusted to decrease computation time at the cost of precision inside the `controlfile.txt`. Especially the coagulation subroutine is sensitive to the number of nodes as it is an $N \times N$ process. The coagulation subroutine may be disabled by setting the corresponding FLAG inside `controlfile.f90`.

7 Visualizing output

A couple of python scripts inside of the `py` subdirectory have been included for reading and visualizing model output.