libreMCM documentation

pitkajuh

April 30, 2023

Contents

1 Introduction

libreMCM (libre Multi Compartment Modelling) is a free software for carrying out deterministic and probabilistic modelling.

2 Features

2.1 Supported numerical formats

LibreMCM supports the following numerical formats (the number of decimals is not restricted)

```
- 1
- 1.0
- 1e1
- 1E1
```

2.2 Supported mathematical functions/operators

LibreMCM supports the following mathematical functions in table 1.

Table 1: Supported mathematical functions and operators.

Function/Operator	Command
x^y	mcm.power(x, y)
sin(x)	mcm.sin(x)
cos(x)	mcm.cos(x)
sqrt(x)	mcm.sqrt(x)
log(x)	mcm.log(x)
exp(x)	mcm.exp(x)

The mathematical functions/operators can be used when creating transfer equations (see section 3.3).

2.3 Supported numerical solvers for time dependent ODEs

At the moment only 4th Order Runge-Kutta method is supported.

2.4 Supported modelling methods

At the moment the program can only solve models deterministically.

3 Creating a model

3.1 How to create model files

Creaing a model requires four files, which called "bins", "compartments", "sim_params" and "compartment.csv". All the files must be located in the same directory.

All the model files are saved in text form (without extension) so they can be created, viewed and edited on any device supporting text files. The user can use any text editor of their choosing. The file syntax of libreMCM's files is somewhat similar to the syntax of C++ code e.g. text blocks are separated with curly brackets ({ and }).

Let us create an example of a text block called "compartments".

```
1  compartments
2    {
3     // Insert the content between the curly brackets.
4    }
```

The name of the text block is on line 1 and the curly brackets on lines 2 and 4 enclose the content of this block. The placement of curly brackets and empty spaces in the text block is flexible and it is up to the user how to place them. For example all examples below are fine

```
compartments{
   // Insert the content between the curly brackets.
}
```

```
compartments
{
// Insert the content between the curly brackets.
}
```

and

```
compartments{
// Insert the content between the curly brackets.
}
```

3.2 Creating compartments

3.2.1 Compartment types

1. origin

   This is a compartment type which is used as an "infinite source of quantity" of which the differential equations can draw values. This type of compartment can only be subtracted from. Section 3.2.3 does not apply to this compartment type.

2. void

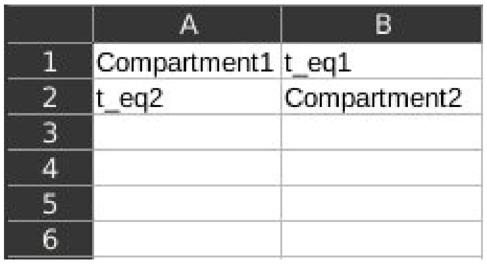
   This is a compartment type which is an opposite to the origin compartment type (see section 1), i.e. it can receive an infinite amount of "quantity". This type of compartment can only be added to. Section 3.2.3 does not apply to this compartment type.

3. compartment

   This is a regular compartment type of which the quantity can be subtracted from and added to. The initial values of this compartment type must be defined separately (see section 3.2.3).

3.2.2 Creating model matrix

The compartments used in the model are defined in a csv file called compartment.csv which uses the semicolon (;) as the delimiter. The compartments must be located on the diagonal axis and the transfer equations can be located on any cells around the diagonal axis. An easy way to create and edit the file is to use a spreadsheet editor. An example file is presented in figure 1, which has been created using LibreOffice Calc.

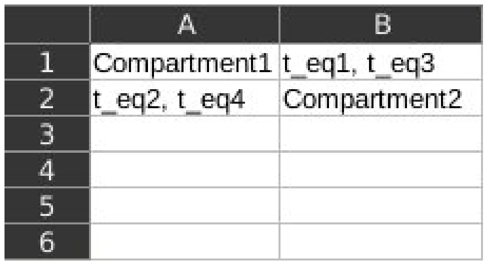


	A	B
1	Compartment1	t_eq1
2	t_eq2	Compartment2
3		
4		
5		
6		

Figure 1: A simple compartment definition csv file using LibreOffice Calc.

In figure 1 there are two compartments on the diagonal axis (Compartment1 and Compartment2) and the transfer equations t_eq1 and t_eq2 (cells A2 and B1) are acting on these compartments.

If there are more than one transfer equation acting on the same compartment, multiple transfer equations can be added by separating them with a comma ",", see figure 2.



	A	B
1	Compartment1	t_eq1, t_eq3
2	t_eq2, t_eq4	Compartment2
3		
4		
5		
6		

Figure 2: A simple compartment definition csv file with multiple transfer equations in one cell using LibreOffice Calc.

3.2.3 Creating compartment definitions

The definitions of the compartments are described in file called "compartments". Let us continue from where we were left on in section 3.1. Now the compartments text block has been created and the next step is to create the individual compartments, which were presented in section 3.2.2. Individual compartment can be created by first creating a text block with the compartment name (see below).

```
Compartment1{

}
```

All compartments must have some initial values which are used for calculating. Initial values can be defined by first creating a text block called "initial_values" inside the compartment text block. This text block must always be called "initial_values".

```
Compartment1{
  initial_values{
    iv1=5;
  }
}
```

Now the text block can be copied to the text block with the compartment name and the result will be

```
compartments
{
  Compartment1{
    initial_values{
      iv1=5;
    }
  }
}
```

The second compartment can be created in similar manner, thus

```
compartments
{
  Compartment1{
    initial_values{
      iv1=5;
    }
  }
  Compartment2{
    initial_values{
      iv2=8;
    }
  }
}
```

3.3 Creating transfer equations

A file called "bin" is used to define the transfer equations of the model. In order to run the model, the file is required to have at least equation defining text block i.e.

```
equations
{
  dydt_1=-2*x;
  dydt_2=3*x;
}
```

If the equations contain many constant values, which one does not want to substitute into the equations, a text block containing definitions of the constant values can be defined.

```
constants
{
  a=-2;
  b=3;
}
```

The section containing the constant values is not necessary to run the model. A file having both blocks could look something like this.

```
constants
{
  a=-2;
  b=3;
}

equations
{
    dydt_1=a*x;
    dydt_2=b*x;
}
```

3.3.1 Specific constant values

If the equations have constants which depend of the initial value (for example the equations describe some chemical reaction but the constant values depend on the element), the constants text block can be written in the following form

```
1  constants
2  {
3    a=-2;
4    (
5     iv1=5;
6     iv2=16;
7    )
8    b=3;
9  }
```

The values defined between parentheses (line 4 and 7) are so called specific values of constant a. The values between parentheses are names of the initial values, which are defined in the "compartments" file. When the differential equations are formed, the differential equation describing the initial value "iv1" will use the constant value $a = 1$ instead of $a = -2$. If the specific value has not been defined for the initial value, the default value will be used, which is -2 for a.

3.4 Simulation settings

The simulation parameters (settings) are described in file called sim_params. Below is an example of simulation settings.

```
simulation_settings
{
  time_start=0;
  time_end=300;
  step_size=5;
  num_method=rk4;
}
```

In order to run the model parameters "time_start", "time_end" and "step_-size" must be defined. These parameters are used to defined the start and end time of the simulation and the step size used in the numerical calculation. The parameter "num_method" defines the used numerical method, but it is optional (i.e. it is not necessary to have the line) at the moment due to the fact, that at the moment only 4th Order Runge-Kutta method is supported.

4 Running simulations

In order to run simulation, a file called "models_cfg" must be created to the same directory where the executable is located. By default the content of the file is the following.

```
model_path{
  tutorials/lotka-volterra/;
  tutorials/simple-chemical-reaction/;
  tutorials/simple-chemical-reaction-v2/;
}
```

In other words this file is used to list the locations of the models the user wants to run. By default libreMCM runs all the tutorial simulations. The user can add the path to the model they want to run. The tutorial models can be commented out or removed from the file anytime.