

# Getting Started

The best way to get started with ModelBuilder is open one of the models included with the distribution and look at it . Yes, it's that simple.

So, if you have already intalled it, start Model-Builder by typing in the console:

```
$ PyMB
```

followed by <enter>.

This will start Model-Buider . From now on I will assume that you know what a system of differential equations is, otherwise you probably should not be using Model-Builder.

The larger box on the main frame, labeled "Differential Equations" is where we are going to start. On this text box you will write your system of equations (or a single equation) The syntax is that of python for mathematical [expressions](#) and [functions](#) and there are some conventions also, which I will explain below:

- First of all, Model-Builder expects only the right-hand-side (RHS) of you equations to be present one per line in the equations box. The LHS is assumed to be of the form  $dy(t)/dt$   $dy_i/dt$  where  $i$  is an index to the the number of equations in your model. This number  $i$  will be used to refer to the state variables of the model (  $y[i]$  ). This index,  $i$ , **MUST** start at 0, so if your model has 3 equations, their state variables will be  $y[0]$  ,  $y[1]$  , and  $y[2]$  .
- Another convention is the reference to model parameters. Any number of parameters may be included in the equations by the using this nomeclature:  $p[0]$  ,  $p[1]$  , and so on. These parameter must the be specified one per line and in ascending order in the "Parameters" box. So the first line would be the definition of  $p[0]$  , the second of  $p[1]$  and so forth.
- The mathematical expressions that make up the equations and the parameters may include any function of the [numpy](#) python module. This allows for the easy contruction of models with a level of sophistication higher than that of a simple algebraic expression. Time may be referenced in the equations or parameter expressions by the variable "t". So if you want a parameter that is a function of time, you can simply write in the parameter box.
- The user familiar with python will also be able to include more advanced structures in the model specification, such as [Lambda functions](#) for instance.

Well, with that out of the way, it remains to explain the rest of the interface which is pretty much self-explanatory:

- The initial values box should include one number for each equation (line) in the equations box. The numbers should be separated by spaces.
- The start time is the time value at which the equations begin to be evaluated. The values of the state variables at this point in time are those specified in the initial conditions.
- End time: You can figure this one out...😁

- Time step: this is the time-step used in reporting the output of the simulation. The actual time step used by the numeric integrator is variable and chosen on the fly. Its normally much finer than what you specify here.
- Critical time steps. Most users will leave this box empty.
- First Step: The size of the first step. Leave at 0 for automatic determination.
- Min Step Size and Max Step Size: Respectively the minimum and maximum value for step sizes as chosen by variable step size algorithm. Leave at zero for automatic determination.
- Full Output check box: If this box is checked a lot of useful information about the integration is included in the output. Check out the output spreadsheet to see what they are.
- Show convergence message: if this box is checked, ModelBuilder will print "Integration successful" to the console after the integration is completed. Useful for debugging purposes only. Uncheck if you are doing uncertainty analyses, as it will slow things down.
- Once you are done entering the necessary information for your model, just press the start button to calculate your model. Enjoy!