

User Manual – ODE MATLAB Models

This document enlists steps required to create and run ODE models with the MATLAB code. MATLAB (2019a or higher) is required to run the scripts.

Step 1:

The first step after opening the project in MATLAB is to add the path via command window:

For example:

```
addpath('F:\MATLAB Backups\platelet-pals-master')
```

Step 2:

In order to create a model, duplicate the script file named *Model_template.m* found in the subdirectory 'Models' and save the duplicate in the same subdirectory i.e. Models with an appropriate name for the model.

Rename the function in this duplicated script as 'ModelX', with X being the model number that you assign.

Also, save the model name in the list at index X, *Model_names(X)* the name of your model with a space like so:

```
Model_names(1) = "Model 1";
```

Return to main directory, if applicable.

Step 3:

The next step is to define the variables within the newly created model file accurately. This script file will take input related to:

- n = the total number of species
- *unit* = global unit of concentration in Molar (M) used for the model. Options include: ["y", "z", "a", "f", "p", "n", "u", "m", "", "K", "M", "G", "T", "P", "E", "Z", "Y"].

Table 1: **Options for specifying a unit.** All possible units for species, forward rates of reactions and time can be chosen from the following list of tabulated options:

Option	Unit
y	yocto
z	zepto
a	atto
f	femto
p	pico
n	nano
u	micro
m	milli
K	kilo
M	mega
G	giga
T	tera
P	peta
E	exa
Z	zetta
Y	yotta

If kept empty, it defaults to “u” or μM .

- Vars = a cell containing rows of species, with each row containing:
 - name of species in the first index,
 - initial concentration (named as IVs in the file) of species in the second index and
 - unit of IV in the third index.

For example:

```
Vars{1} = 'L';          IVs(1) = 1000;      Var_unit(1) = "";
Vars{2} = 'R';          IVs(2) = 10;       Var_unit(2) = "";
Vars{3} = 'L_R';        IVs(3) = 0;       Var_unit(3) = "";
```

- PlotVars = define names of species as they should appear on plots, without underscores.

For example:

```
PlotVars{1} = 'L';
PlotVars{2} = 'R';
PlotVars{3} = 'LR';
```

- eqns = equations involving species. This is used to define reactions involving species.

For example:

```
%           in           out      k value numbers
eqns{1} = {["L", "R"], "L_R", 1}; % {"R", "LL_R"], "LL_RR",
2}
```

In the above example, the first index is an array of substrates i.e. L and R, the second index corresponds to the product i.e. L_R, and the third index identifies the row number of the cell containing K values of all reactions, to be defined in the next point.

- *K* = a cell containing rows of rate constants in the index order matching that of the equations they relate to, with each row containing the forward rate constant in the first index and the backward rate constant in the second index.
- *K_unit* = unit of in Molar (M) used for the forward rate constant. Options include: ["y", "z", "a", "f", "p", "n", "u", "m", "", "K", "M", "G", "T", "P", "E", "Z", "Y"].
- *multiples* = a cell to be populated with the reactions that have more than one possibility to occur (for example, if the ligand has two un occupied epitopes, then the possibility for a receptor to bind at any one of the epitopes increases as it has two options to choose from for attachment).
- *catalysts* = a cell to be populated with the identity enzymes or catalysts, if there are any such species present in the model.
- *constants* = this is a variable that can be used to identify any species whose concentration remains constant, because it is in vast excess and therefore assumed to be constant.

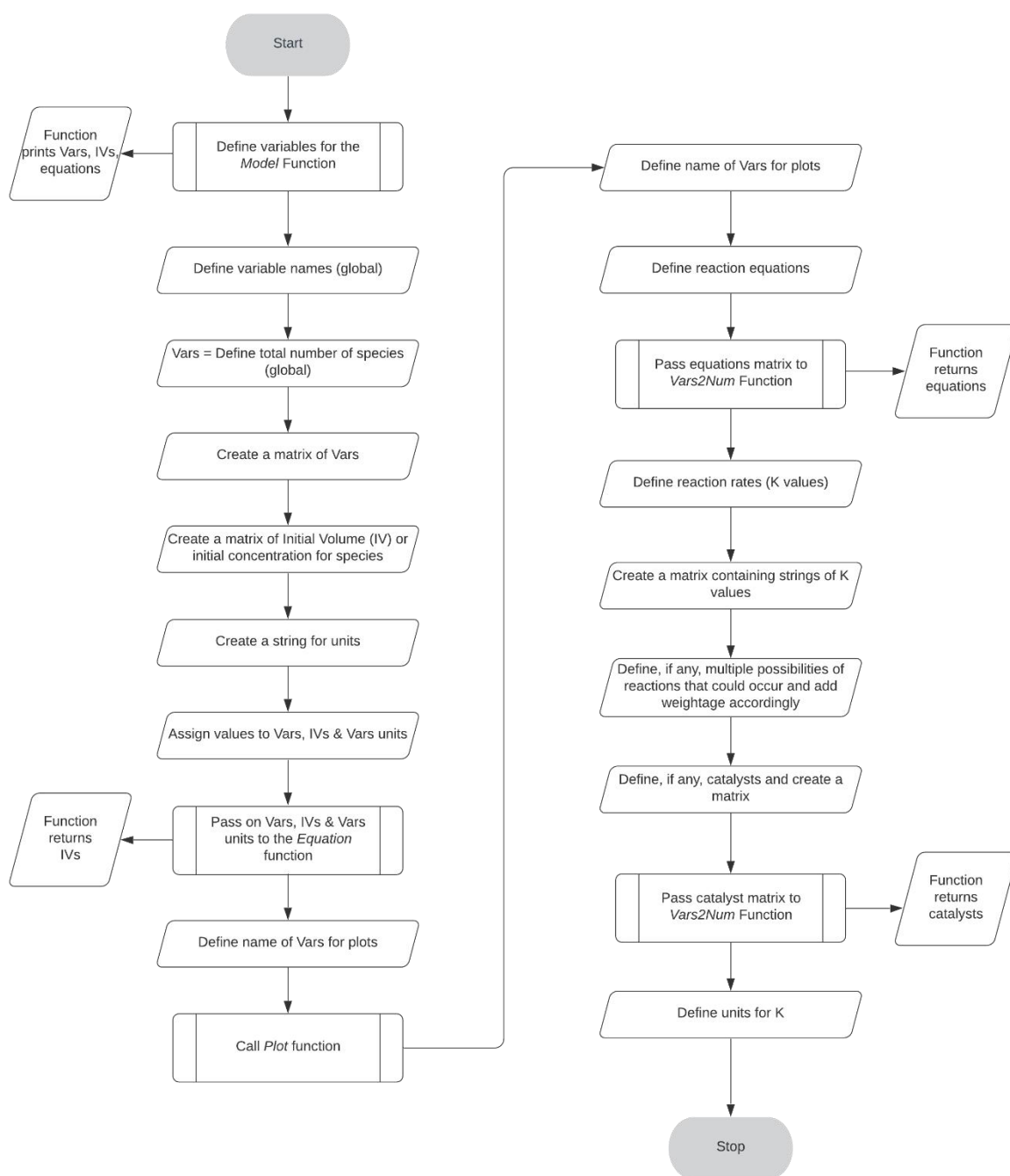


Figure 1. **Summary of process flow in order to create a model.** The flowchart summarises the steps taken in order to create an ODE model in MATLAB.

Commands

Once the model is created and saved, various commands can be run via the command window to perform the following:

Model initialisation

In order to initialise the model, run the following command:

```
Models (X)
```

With X being the model number that was defined in Step 1.

This will return a written output in the command window with a list of equation(s), initial concentrations (IVs) of all species and a basic network diagram the model.

Time course

To plot time course to equilibrium, run any of the following commands:

```
Plot (X, Y)
```

```
Plot (X, Z)
```

```
Plot (X, Y:Z)
```

where X is the model number, and Y and Z are the numbers corresponding to the species for which time course is to be plotted. This will generate on the y-axis the absolute numbers of species plotted.

In order to represent the species in percentages of their maximum value, use the keyword 'Proportion' after specifying which species are to be plotted:

```
Plot (X, Y:Z, "Proportion")
```

Also, the plots can be run for a specific amount of time that can be defined like so:

```
Plot (X, Y:Z, "T", A)
```

where A corresponds to time in seconds, if no unit is specified. In order to keep the time scale within a specific unit range, the unit may also be specified:

```
Plot(X, Y:Z, "T", A, "unit")
```

where unit is any of the units enlisted in Table 1.

Stoichiometric Analysis

The effect of varying initial concentration of a single species at a time vis a vis the system can be simulated via the following command:

```
Plot_Change("IV", X, "J", [<range of IVs to be plotted>], Q)
```

where "IV" is used to specify that the initial concentration of a species is to be modulated, X is the model number, J is the name of species to be modulated with the range of initial concentrations to be plotted given as an array, followed by Q as the species on which the effect is under investigation.

The same function can also be called in order to modulate the forward or backward rate of a specific reaction in order to see how the system changes:

```
Plot_Change("KD", X, J, [<range of rate constant value to be plotted>], Q)
```

where "KD" is used to specify that a rate constant is to be modulated, X is the model number, J is the number corresponding to the reaction for which the forward rate constant is to be modulated (numerical order as defined in Step 3 in the cell named K), with the range of the rate constant to be plotted given as an array, followed by Q as the species on which the effect is under investigation. If the value specified for J is negative, that will lead to changing the backward rate constant for that reaction instead.

Steady state analysis

To plot steady state values on a semi-log x axis while varying the initial concentration or IV of one of the species in the model, the following command can be used:

```
Plot_SS_Change_Log("IV", Y:Z, M, [<range of values to be plotted>])
```

where Y and Z are the species to be plotted, M identifies the species for which the initial concentration (IV value) is modulated and at the end the range of values to be plotted is specified as an array.

As before, percentages can be plotted on y axis with the inclusion of the keyword “Proportion” in the command.

Stochastic analysis

Addition of the keyword “Stochastic” to a basic time course plot generates the stochastic version, for example:

```
Plot(X, Y:Z, "Stochastic", "Proportion")
```

As an example, a model for a soluble monovalent ligand and a monomeric brane receptor is already created and named as ‘Model 1’ in the subdirectory named Models.