Computer Aided Modeling Tool – ModDev: Tutorial Examples. Part II

Rafiqul Gani, Marina Fedorova

CAPEC
Department of Chemical Engineering
Technical University of Denmark
DK-2800 Lyngby, Denmark

1. Introduction

This tutorial concentrates on the basic features of ModDev in order to guide the user through the different steps of the model generation procedure followed in ModDev. Simple examples are only considered here.

The model generation procedure followed in ModDev consists of providing descriptions of the process (to be modeled) in terms of shells, streams and connections. Based on the descriptions, ModDev generates the corresponding equations. In case the corresponding equations of a shell/stream/connection are not available, ModDev allows the user to define new equations.

The computer aided modeling tutorial is divided into two parts. Part I only deals with generation of the process model equations while part II deals with model analysis and code-generation.

2. Model Analysis and Code Generation in ModDev

2.1 Tutorial example 1 – Steady state tank-mixer

After a first version of the model equations has been generated, they can be analyzed with the available model analysis tools in ModDev. Let us start with the simple steady state tank-mixer model. Start ModDev and open the model1a.mdl file.

Before continuing it is necessary to make some changes with model equations. Enthalpy equation for steam 3 and connecting equation between the shell and steam 3 have different variables for temperature. So we need to rename one of the variables to make temperature variables identical.

Select from the tool-bar “model”, “mathematical analysis and manipulation”, “analysis and manipulation”, “model equations” as shown in figure 1.

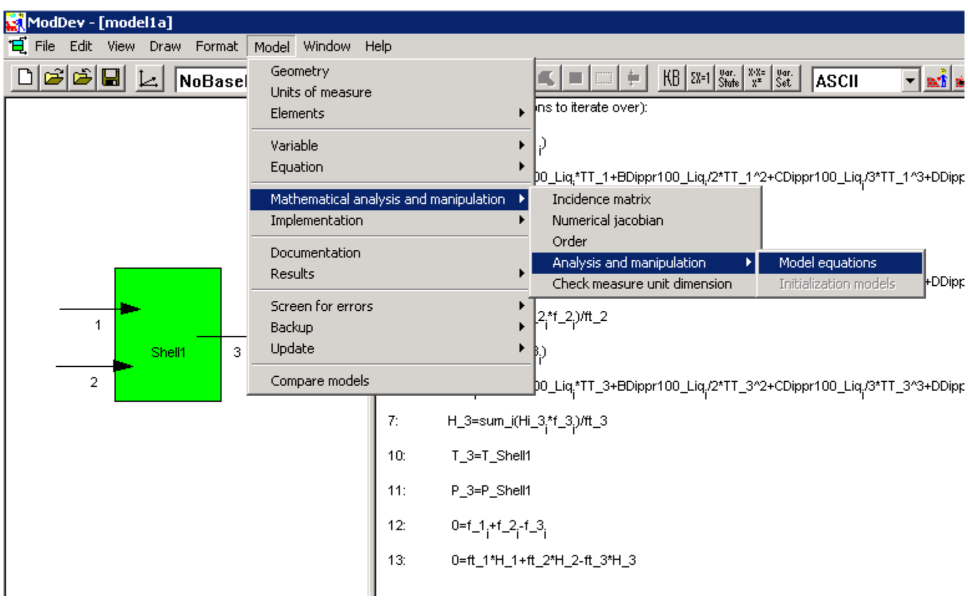


Figure 1: Selection of model manipulation from the main tool-bar in ModDev.

From the list choose the connecting equation between shell and steam 3 (fig. 2).

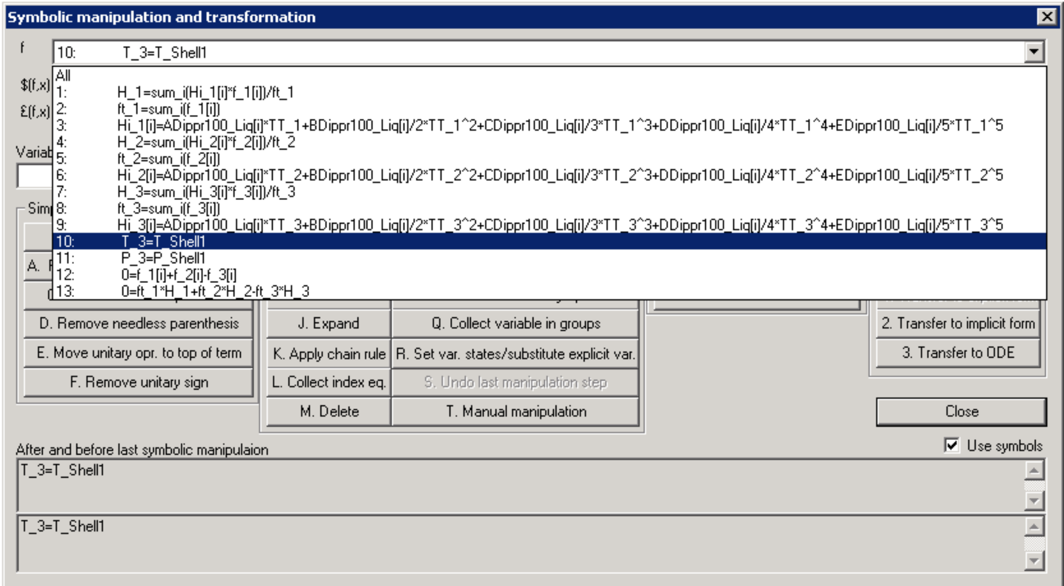


Figure 2: Selecting of equation for changes.

After that choose "T. Manual manipulation" and rename variable Temperature{3} to Temperature_Absolute{3} (see fig. 3 and fig. 4).

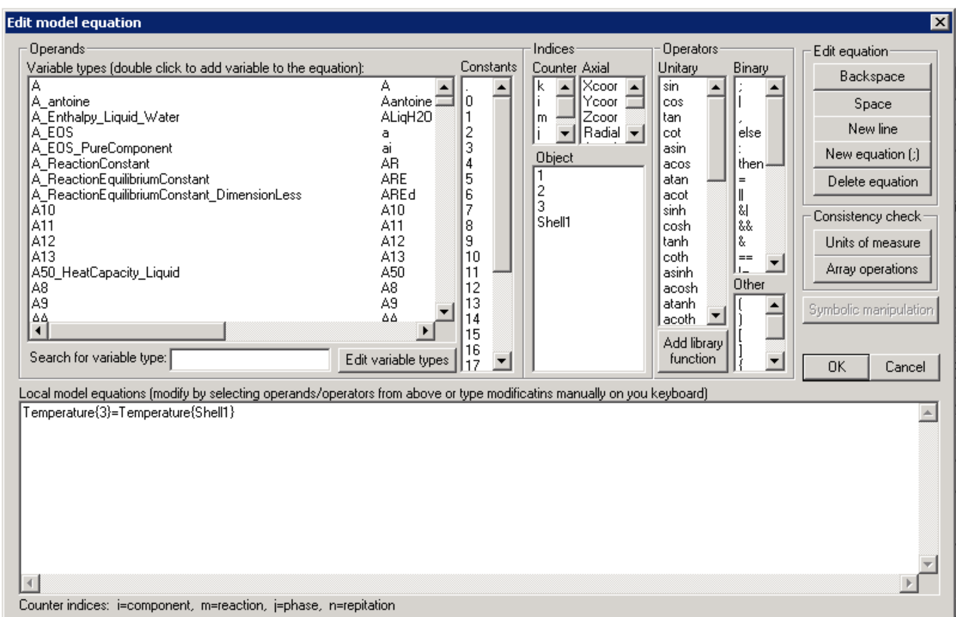


Figure 3: Manipulation with temperature variable (before).

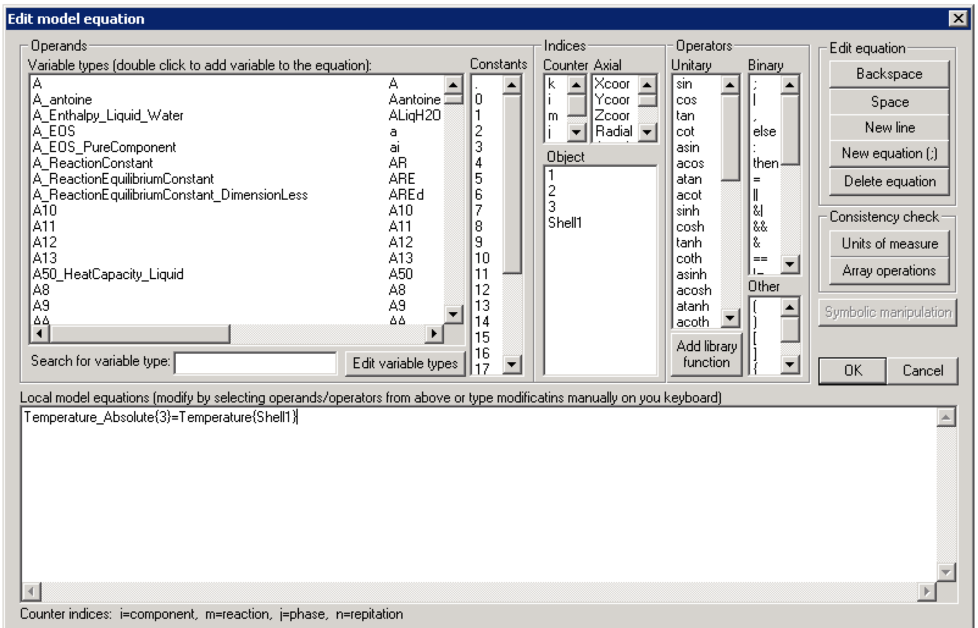


Figure 4: Manipulation with temperature variable (after).

Close the tool with “OK” and close the previous tool too.

Then select "model", "variable" and "states" from the tool-bar in ModDev as shown in figure 5.

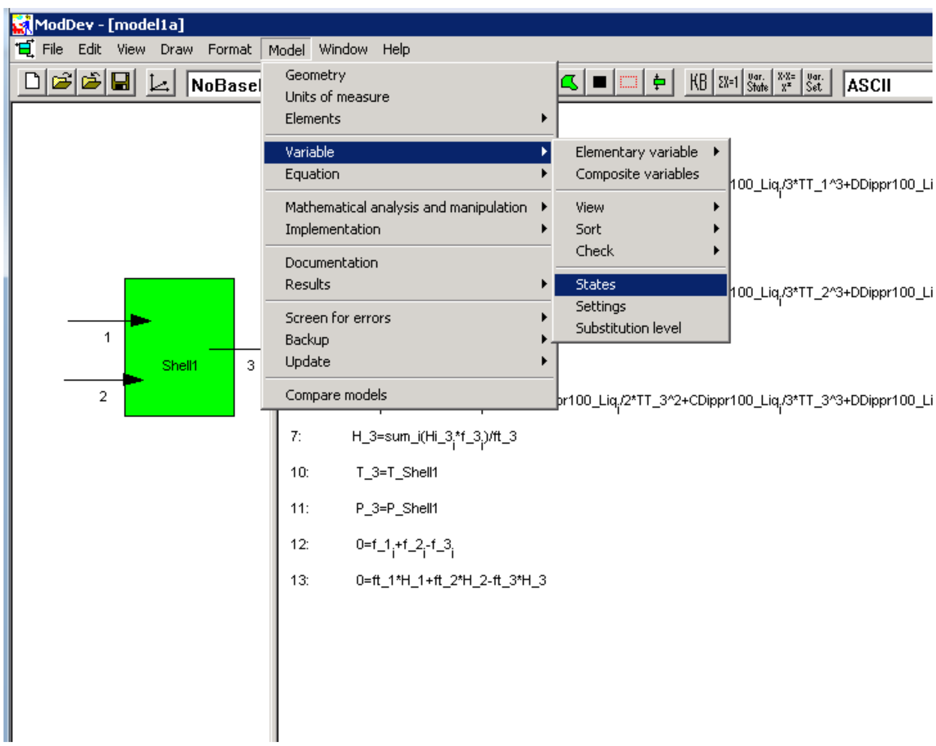


Figure 5: Selection of "variable states" from the main tool-bar in ModDev.

In the "variable states" tool, transfer the known variables listed under "unknown variables" to "known variables" as shown in figure 6. Note also the list of parameters whose values will have to be specified (when solving the model equations) and the degree of freedom analysis in terms of number of implicit equations and how many more variables need to be specified. When the number of variables to specify is zero (as shown in figure 6), the degree of freedom condition has been satisfied and we have equal number of variables and equations. This is also reflected by the number of variables remaining in the list of "unknown variables". On return to the main ModDev screen, the model equations are divided into two sets – "known model equations" and "model equations". The "known model equations" are only functions of the "known variables" and/or parameters. The "model equations" contain the set of "unknown variables" (see figure 7). The "model equations" are ordered in terms of "explicit" model equations (written first) and "implicit" model equations (equations with zero on the left hand sides).

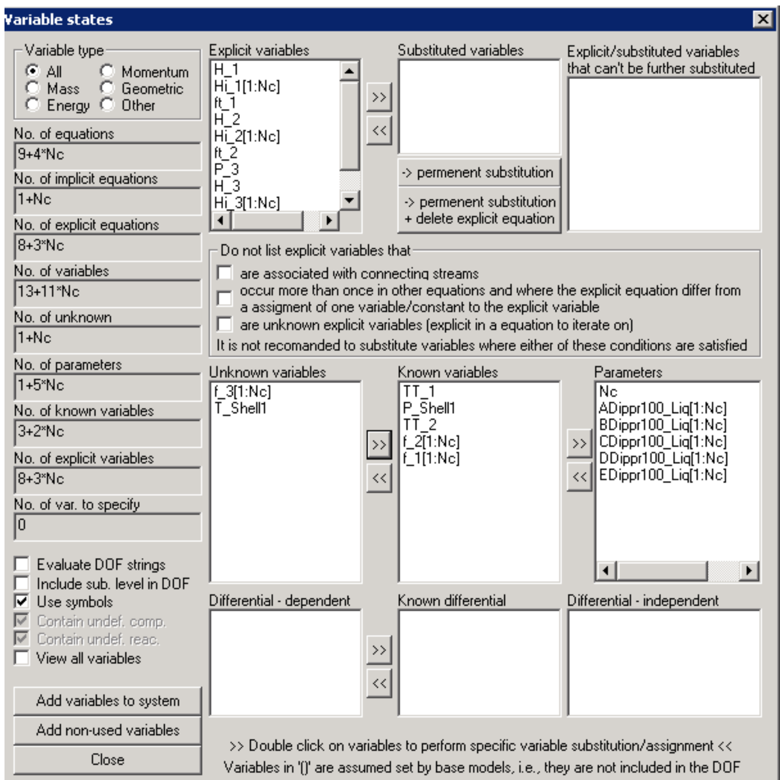


Figure 6: Selection of variables as "known" and "unknown" in the "variable states" tool.

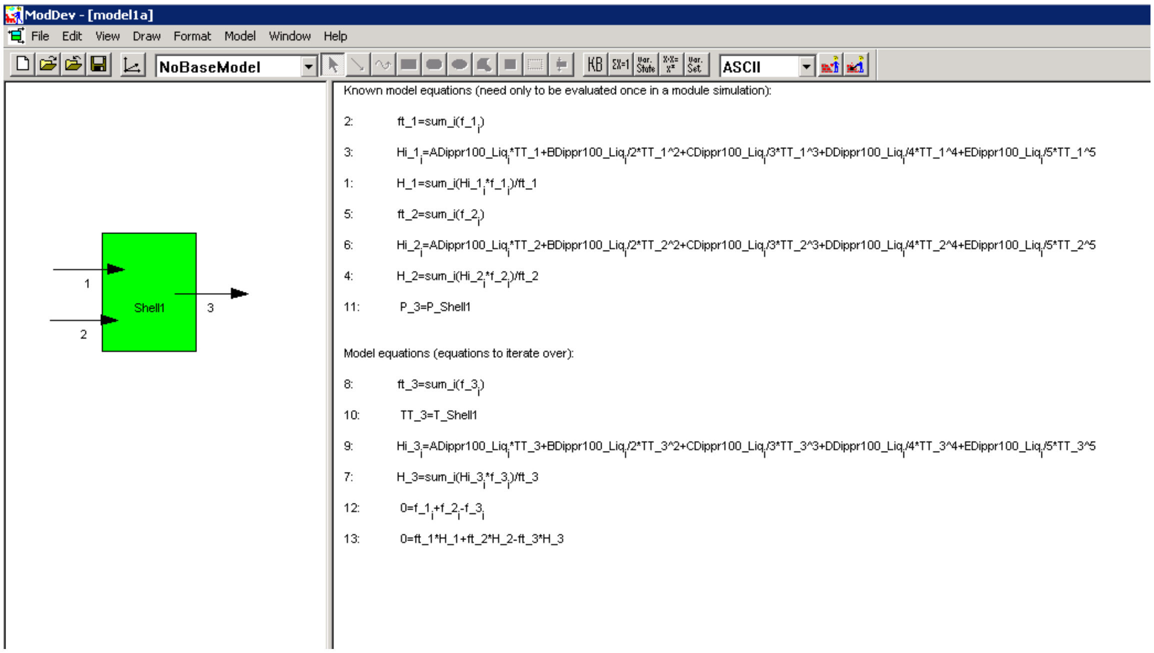


Figure 7: Representation of the process model equations after selection of variable states.

Now the model equations can be analyzed in terms of "incidence matrix". Figure 8 shows the way to find the "incidence matrix" analysis tool in the tool-bar

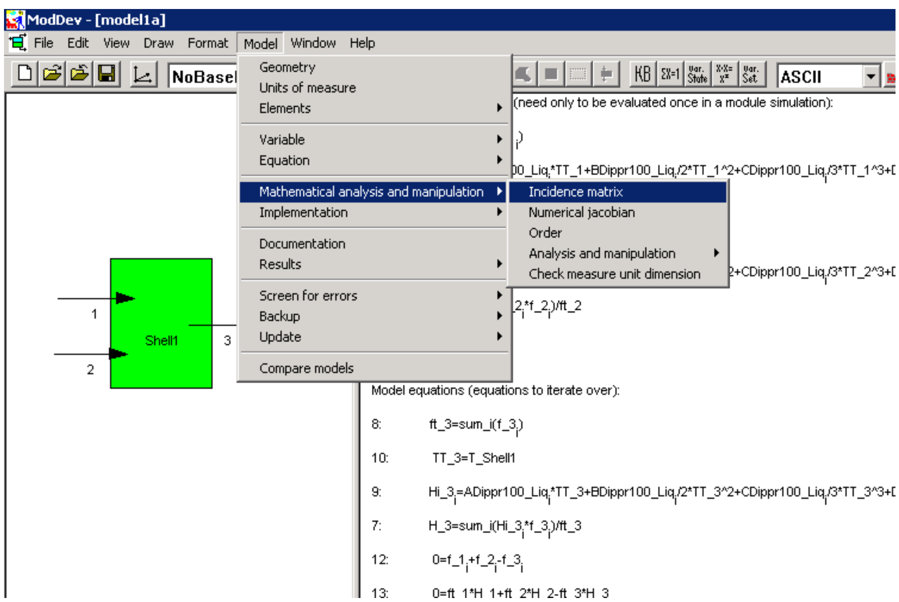
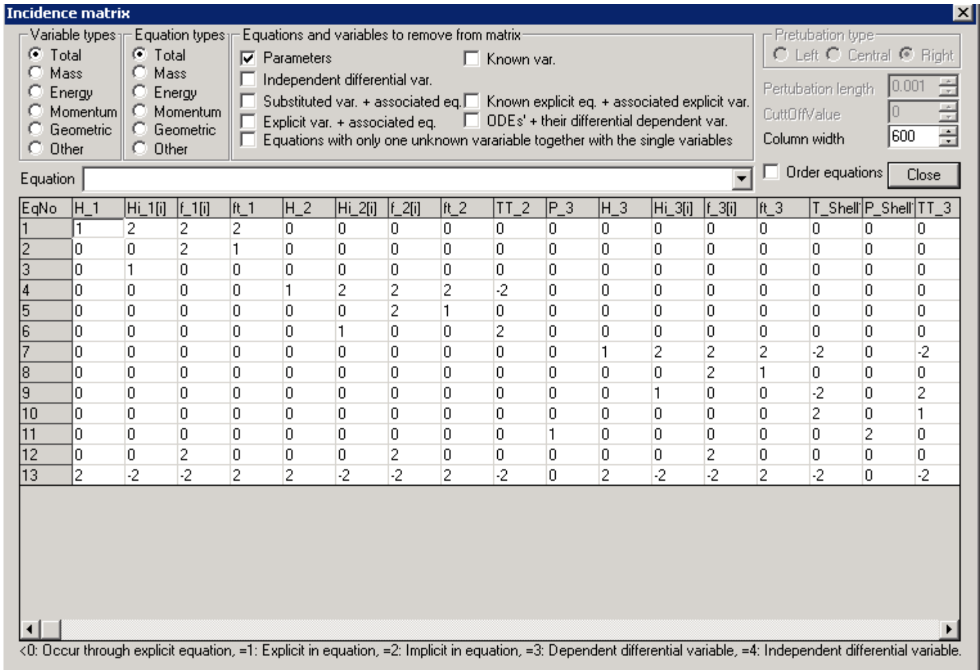


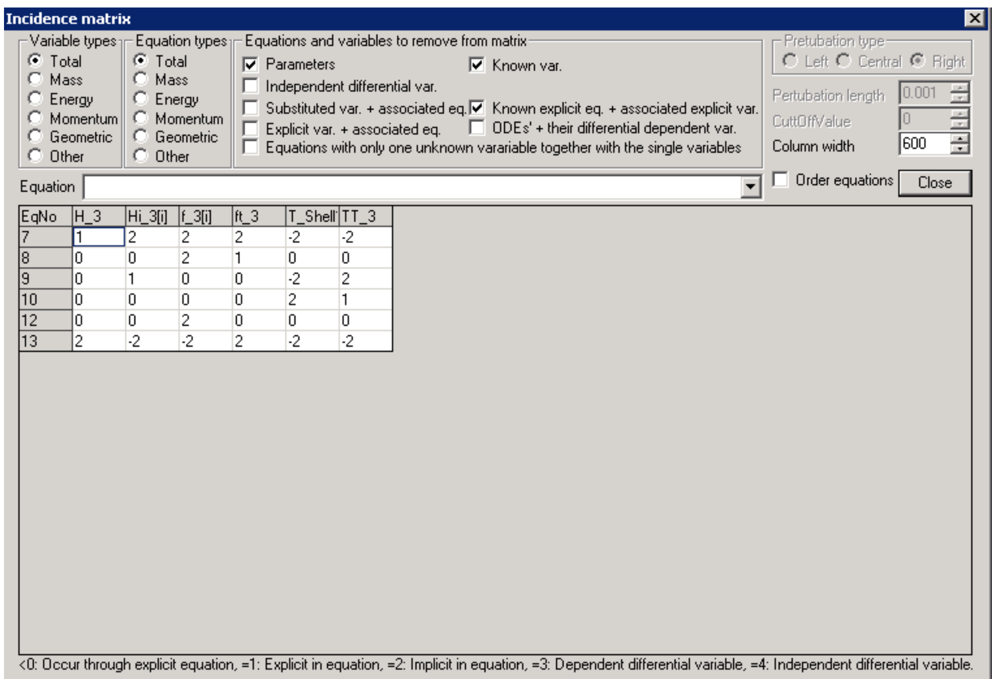
Figure 8: Selecting "incidence matrix" tool in tool-bar.

In figure 9a, the complete incidence matrix is shown. In figure 9b, the known variables and the known model equations have been removed. From figure 9b, it is clear that the equations can be arranged into three blocks – Eqs. 12 &8, Eqs. 9, 7 & 13 and Eq. 10. It can also be noted that Eqs. 7, 9 & 13 include the "implicit" model equations with respect to energy balance.



EqNo	H_1	Hi_1[i]	f_1[i]	ft_1	H_2	Hi_2[i]	f_2[i]	ft_2	TT_2	P_3	H_3	Hi_3[i]	f_3[i]	ft_3	T_Shell'	P_Shell'	TT_3
1	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	2	2	2	-2	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	2	2	2	-2	0	-2
8	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	2
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0
12	0	0	2	0	0	0	2	0	0	0	0	0	2	0	0	0	0
13	2	-2	-2	2	2	-2	-2	2	-2	0	2	-2	-2	2	-2	0	-2

Figure 9a: Incidence matrix of the process model equations and variables.



EqNo	H_3	Hi_3[i]	f_3[i]	ft_3	T_Shell'	TT_3
7	1	2	2	2	-2	-2
8	0	0	2	1	0	0
9	0	1	0	0	-2	2
10	0	0	0	0	2	1
12	0	0	2	0	0	0
13	2	-2	-2	2	-2	-2

Figure 9b: Incidence matrix after removal of known model equations and variables.

Now, if we are satisfied that the model is correct, we can generate a code for the model in the available programming languages in ICAS. It is possible to obtain the model in ASCII, it is possible to generate a code for use in gPROMS, it is possible to generate a code in RPN for direct use in the ICAS simulation engine and it is possible to generate a code as a subroutine in FORTRAN. Note that if the model parameters and the known variables are not defined or specified, the simulator will expect values of these parameters and variables to be passed to the model. Otherwise, the model equations would not be solved correctly. A sample RPN-code is given in appendix. The steps for model generation are given below.

Step 1: Select the model type from the list on the tool-bar (fig. 10).

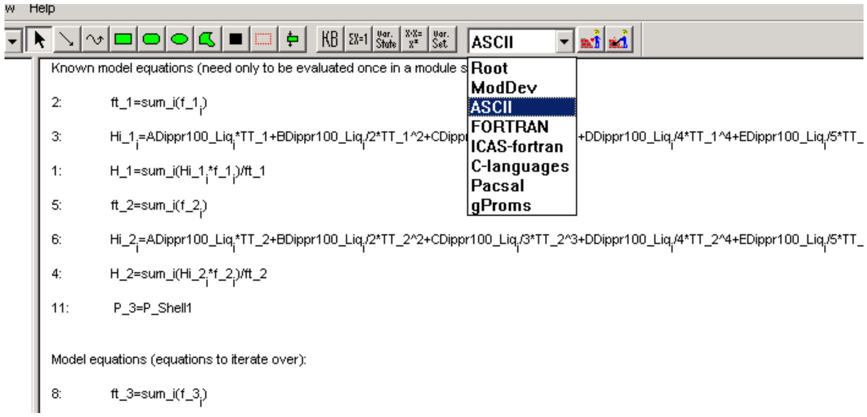


Figure 10: Selecting of the model implementation type.

Step 2: Note that before the model will be generated, it is necessary to save the current model. Save model1a.mdl as model1a-a.mdl (after model analysis). Press the button "implement the model equations" on the tool-bar (fig. 11).

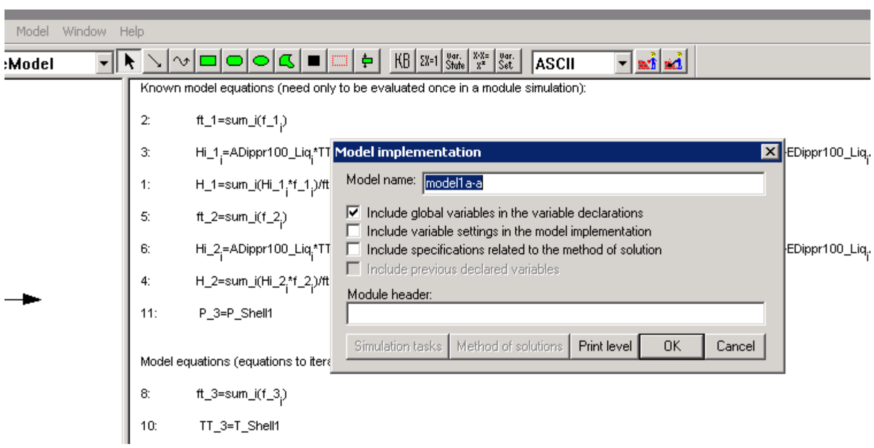


Figure 11: Model implementation tool.

Step 3: The generated model can be viewed through the "view/delete/modify developed models" tool as shown in figure 12.

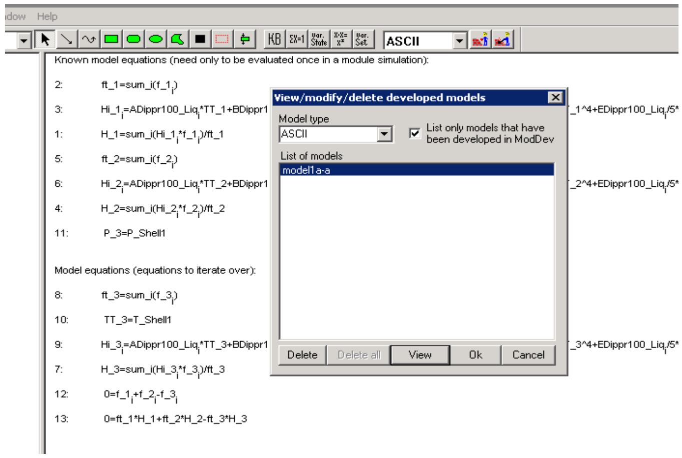


Figure 12: View/delete/modify developed models

2.2 Tutorial example 2 – Dynamic tank-mixer

We will now use model1b.mdl from part I of the tutorial on ModDev. It is necessary to make the same rename of the temperature variable as in example 2.1. Figures 13a-13c show the "variable states", the "incidence matrix" and the final model form before code generation.

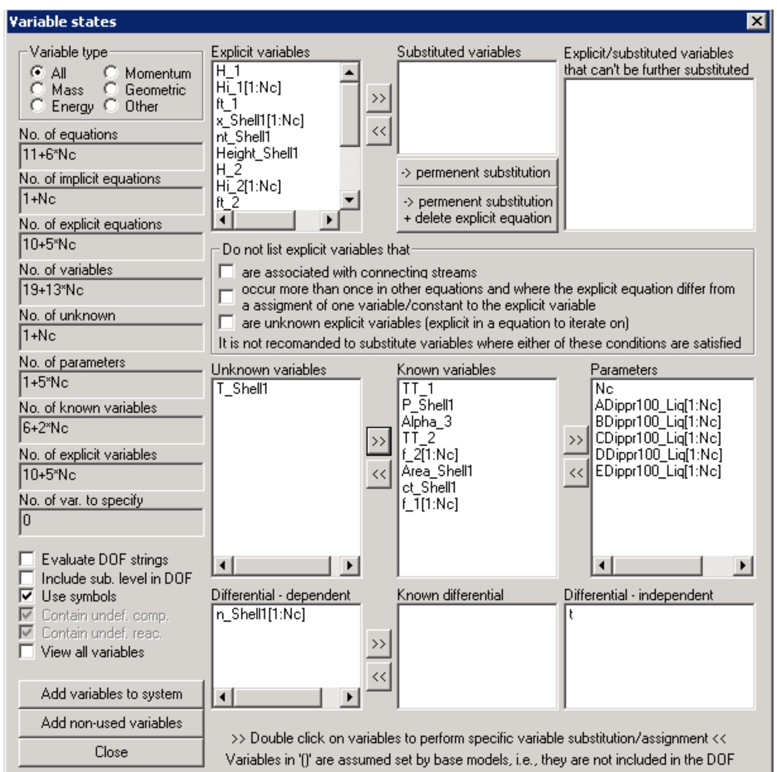
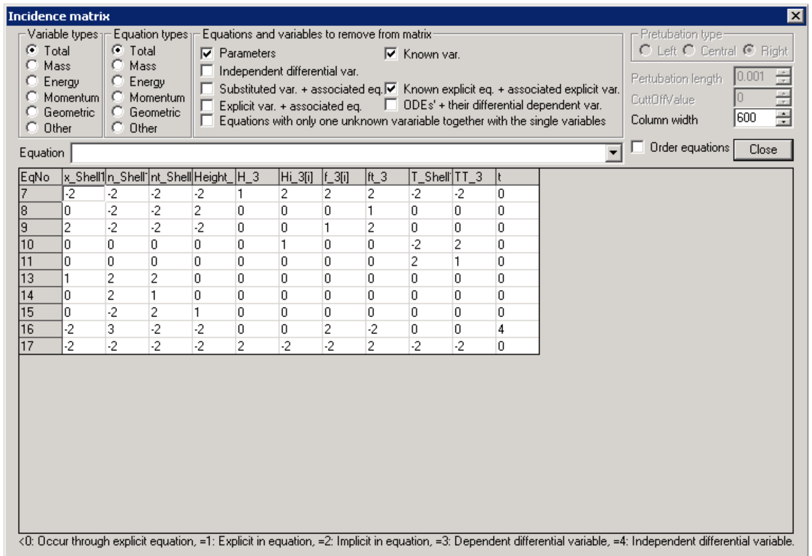


Figure 13a: Selection of variable states for dynamic tank-mixer model.

Note the assignment of the "differential variable". In comparison to the corresponding steady state model, it can be seen that additional variables such as "Area_Shell1" and "Alpha_3" now need to be specified. Also, either the volume or the height of the mixing tank would have been needed if the liquid height (volume) were to be checked against the tank height (volume).



EqNo	x_Shell1	n_Shell	nt_Shell	Height_	H_3	Hi_3[i]	f_3[i]	ft_3	T_Shell	TT_3	t
7	-2	-2	-2	-2	1	2	2	2	-2	-2	0
8	0	-2	-2	2	0	0	0	1	0	0	0
9	2	-2	-2	-2	0	0	1	2	0	0	0
10	0	0	0	0	0	1	0	0	-2	2	0
11	0	0	0	0	0	0	0	0	2	1	0
13	1	2	2	0	0	0	0	0	0	0	0
14	0	2	1	0	0	0	0	0	0	0	0
15	0	-2	2	1	0	0	0	0	0	0	0
16	-2	3	-2	-2	0	0	2	-2	0	0	4
17	-2	-2	-2	-2	2	-2	-2	2	-2	-2	0

Figure 13b: Incidence matrix analysis for the dynamic tank-mixer model.

The above incidence matrix has not order the equations into differential and algebraic and in a block triangular form. It can be noted, however, that the algebraic equations are not singular. Therefore, this dynamic model represented by a set of differential and algebraic equations (DAEs) represents an index 1 problem.

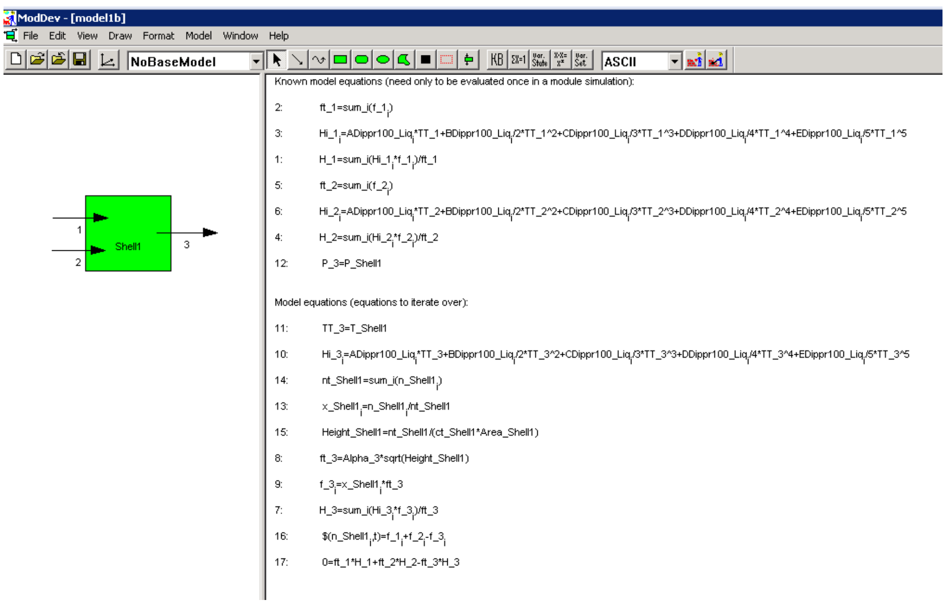


Figure 13c: Final model form.