# 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).

Symbolic manip	Symbolic manipulation and transformation						
f 10:	T_3=T_Shell1				•		
10:       T_3=T_Shell1         \$(fix)       11:         P       All         \$(fix)       11:         P       10:         T_3=T_Shell       \$         \$(fix)       11:         P       10:         \$(fix)       1:         P       10:         \$(fix)       1:         P       10:         \$(fix)       1:         P       10:         \$(fix)       1:         P       10:         P       10:         \$(fix)       1:         P       10:         \$(fix)       1:         P       1:         P       1:         P       1:         P       3:         \$(fix)       1:         P       3:         \$(fix)       1:         P       3:         P       3:         \$(fix)       1:         P       3:         P       3:         P       3:         P       3:         P       3:         P       3: </th							
( <u>13:</u> (	)=ft_1*H_1+ft_2*H_2-ft	_3*H_3	0. Collect variable in groups	L]	2. Transfer to implicit form		
E. Move unitar	y opr. to top of term	K. Apply chain rule	R. Set var. states/substitute explicit var.		3. Transfer to ODE		
F. Remov	/e unitary sign	L. Collect index eq.	S. Undo last manipulation step				
[		M. Delete	T. Manual manipulation		Close		
After and before la	ust symbolic manipulaio	n			🔽 Use symbols		
T_3=T_Shell1 T_3=T_Shell1					× ×		

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).

dit model equation					×
Operands Variable types (double click to add variable to the equat A A Aantoine A_E0ts AE05_PureComponent A_ReactionEquilibriumConstant A_ReactionEquilibriumConstant AI10 A11 A12 A13 Search for variable type:	ion): A Aantoine ALigH2D a AR ARE ARE ARE ARE ARE ARE ARE	Constants 0 1 2 3 4 5 6 6 7 8 9 9 10 11 12 13 14 15 16 17 •	Indices Counter Axial K Coor Y Yeoor Z Zcoor Badial V Diject 1 2 3 Shell1	Unitary     Binary       sin     Image       sin     Image       cos     Image       tarn     Image       asin     Image       acos     Image	Edit equation Backspace Space New line New equation (;) Delete equation Consistency check Units of measure Array operations Symbolic manipulation
Local model equations (modify by selecting operands/ope	Hators from above or ty	pe modific	atms manually on you i	keyboard)	× ×

Figure 3: Manipulation with temperature variable (before).

it model equation				×
Operands Operands Variable types (double click to add variable to the equation A _antoine A_Enthalpy_Liquid_Water A_EOS A_EOS_PureComponent A_ReactionEquilibriumConstant A_ReactionEquilibriumConstant A_ReactionEquilibriumConstant A12 A12 A13 A50 HeatCapacity_Liquid A8 A9 A9	): Constants A Aantoine A ALiqH2D 1 a ai 3 AR 4 ARE 5 ARE 6 AID 7 AID	Indices Counter Axial K Xcoor Y YCoor i Radial V Diject 1 2 3 Shell1	Operators Unitary Sin Cos Itan Cot Satis S	Edit equation Backspace Space New line New equation (;) Delete equation Consistency check Units of measure Array operations Symbolic manipulation
Search for variable type: 	it variable types 15	catins manually on you	Add library function keyboard)	OK Cancel

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.

ModDev - [model1a]		
Image: Contract of the second seco	Model Window Help Geometry Units of measure Elements	KB 2x=1 Stars x <sup>24</sup> Set ASCII
1 Shell 3	Variable           Equation           Mathematical analysis and manipulation Implementation           Documentation           Results           Screen for errors           Backup           Update           Compare models           7:         H_3=sum_i(HL_3*f_3)/fL_3           10:         T_3=T_Shell1           11:         P_3=P_Shell1           12:         0=f_1+f_2+f_3;           13:         0=ft_1*H_1+ft_2*H_2-ft_3*H	Elementary variable Composite variable Composite variables 100_Liq,/3*TT_1^3+DDippr100_Li Sort Check Settings Substitution level pr100_Liq,/2*TT_3^2+CDippr100_Liq,/3*TT_3^3+DDippr100_Li

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 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.

ncidence	e matri	ж															>
Variabl	le types	Equat	tion types	Equal	tions and	variable	s to remo	ove from	matrix					Preta	ubation ty	ре —	
						I O L	eft 🔿 🛛	ientral (	🖲 Right								
O Ma	SS	O Ma	ass	🗌 🗖 Ine	depende	nt differe	ntial var.							Destude	- 6	-a. In	001 🖂
C Ene	ergy	🛛 🖸 En	iergy	🗆 🗆 Su	ubstituted	lvar. + a	ssociate	d ea. 🗖	Known e	xplicit eq	1. + assoi	ciated exp	olicit var.	Fertub	ation len	gin ju	<u><u> </u></u>
	mentum		omentum	E Ex	plicit var.	. + assoc	iated eq		ODEs' +	their diffe	erential d	ependen	t var.	CuttOf	fValue	U	-
	ometric		sometric her	E E E	quations (	with only	one unk	nown va	arariable t	ogether v	with the s	single vari	iables	Colum	n width	6	00 🚊
Equation													-	   🗖 On	der equa	tions 🗌	Close
c quatior	<u>' </u>																
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_She	l' P_She	ell'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	U	1	U	U	U	U	U	U	U	U	U	U	U	U	U	U	0
4	U	U	U	U	1	2	2	2	-2	U	U	U	U	U	U	U	U
5	U	U	U	U	U	U	2	1	U	U	U	U	U	U	U	U	U
<u>ь</u>	U	U	U	U	U	1	U	U	2	U	U	U	U	U	U	U	0
/	U	U	U	U	U	U	U	U	U	U	1	2	2	2	-2	U	-2
8	U	U	U	U	U	U	U	U	U	U	U	0	2	1	U	0	0
9	0	U	0	U	U	U	U	0	U	U	0	1	U	U	-2	0	2
10	0	0	0	0	U	U	0	0	0	1	0	0	U	U	2	0	-
10	0	0	0	0	0	0	0	0	0		0	0	0	0	0	2	0
12	0	0	2	0	0	0	2	0	0	0	0	0	2	0	0	0	2
13	2	-2	-2	2	2	-2	-2	2	-2	U	2	-2	-2	2	-2	U	-2
-																	

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

Incidence matrix	×
Variable types       Equation types       Equations and variables to remove from matrix            • Total         • Total         • Mass         • Mass         • Mass         • Lenergy         • Energy         • Momentum         • Geometric         • Geometric         • Other         • Other         • Other         • Other         • Equations and variables to remove from matrix         • Known var.         • Known var.         • Known var.         • Independent differential var.         • Substituted var. + associated eq.         • Momentum         • Geometric         • Geometric         • Other         • Other	Pretubation type       C Lett     C Central       Pertubation length     0.001       CuttOffValue     0       Column width     600
Equation	Close
EqNo       H_3       H_3(j)       F_3(j)       H_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       1         12       0       0       2       0       0       0         13       2       -2       -2       -2       -2         9       0       0       2       0       0       0         13       2       -2       -2       -2       -2       -2         9       0       1       0       0       0       0       0         13       2       -2       -2       -2       -2       -2       -2         9       0       1       0       0       0       0       0       0       -2       -2       -2         13       2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2       -2<	
C: Occur through explicit equation, =1: Explicit in equation, =2: Implicit in equation, =3: Dependent differential variable, =4	: Independent differential variable.

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.

▼ 📐 🔷 🗖 🗢 🕰 🔳 🖾 🛊 KB 58=1 9ar. 8xx= 9ar. ASCII - 🖬 🖬 Known model equations (need only to be evaluated once in a module s Root ModDev 2:  $ft_1=sum_i(f_1)$ ASCII FORTRAN Hi\_1;=ADippr100\_Liq;\*TT\_1+BDippr100\_Liq;/2\*TT\_1^2+CDippr +DDippr100\_Liq,/4\*TT\_1^4+EDippr100\_Liq,/5\*TT\_ 3: ICAS-fortran C-languages 1: H\_1=sum\_i(Hi\_1;\*f\_1;)/ft\_1 Pacsal 5: ft\_2=sum\_i(f\_2;) gProms Hi\_2;=ADippr100\_Liq;\*TT\_2+BDippr100\_Liq;/2\*TT\_2^2+CDippr100\_Liq;/3\*TT\_2^3+DDippr100\_Liq;/4\*TT\_2^4+EDippr100\_Liq;/5\*TT\_ H\_2=sum\_i(Hi\_2;\*f\_2;)/ft\_2 11: P\_3=P\_Shell1 Model equations (equations to iterate over): ft\_3=sum\_i(f\_3;)

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).

Model Window H	lelp		
:Model 🔻		ッ□੦੦ᢏ∎	HB EX=1 Barrie XX= Barrie ASCII 💌 🐋
	Known	model equations (need only	to be evaluated once in a module simulation):
	2:	ft_1=sum_i(f_1 <sub>i</sub> )	
	3:	Hi_1;=ADippr100_Liq;*TT	Model implementation
	1:	H_1=sum_i(Hi_1*f_1;)/ft	Model name: model1a-a
	5:	ft_2=sum_i(f_2 <sub>i</sub> )	Include global variables in the variable declarations Include variable settings in the model implementation
	6:	Hi_2=ADippr100_Liq.*TT	Include specifications related to the method of solution     EDippr100_
	4:	H_2=sum_i(Hi_2*f_2;)/ft	I include previous declared variables Module header:
	11:	P_3=P_Shell1	
	Model e	quations (equations to itera	Simulation tasks Method of solutions Print level OK Cancel
	8:	ft_3=sum_i(f_3 <sub>j</sub> )	
	10:	TT_3=T_Shell1	

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.

idow H	lelp		
-		៴៙៙៙៹៙	(B EX#1 Store 25 St. ASCII 💌 🐋
	Knowr	n model equations (need only to be evalu	ated once in a module simulation):
	2:	ft_1=sum_i(f_1)	View/modify/delate developed models
	3:	Hi_1,=ADippr100_Liq_1TT_1+BDippr1	Model type
	1:	H_1=sum_i(Hi_1^f_1)/ft_1	ASCII List only models that have been developed in ModDev
	5:	ft_2=sum_i(f_2)	List of models
	6:	Hi_2 <sub>j</sub> =ADippr100_Liq <sub>j</sub> *TT_2+BDippr1	_2^4+EDippr100_Liq/5*
	4:	H_2=sum_i(Hi_2*f_2)/ft_2	
	11:	P_3=P_Shell1	
	Model	equations (equations to iterate over):	
	8:	ft_3=sum_i(f_3)	
	10:	TT_3=T_Shell1	
	9:	Hi_3 <sub>i</sub> =ADippr100_Liq <sub>i</sub> *TT_3+BDippr1	Delete Delete all View Ok Cancel _3^4+EDippr100_Liq/5*
	7:	H_3=sum_i(Hi_3,*f_3,)/ft_3	
	12:	0=f_1 <sub>i</sub> +f_2 <sub>i</sub> -f_3 <sub>i</sub>	
	13:	0=ft_1*H_1+ft_2*H_2-ft_3*H_3	

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.

¥ariable states			×
Variable type All C Momentum C Mass C Geometric C Energy C Other No. of equations 11+6*Nc No. of implicit equations 1+Nc No. of explicit equations 10+5*Nc No. of variables 19+13*Nc No. of unknown 1+Nc	Explicit variables H_1 H_1[1:Nc] H_1 H_1[1:Nc] H_1 H_2 H_2 Do not list explicit variables are associated with com a cassigment of one varia are unknown explicit variables It is not recomanded to subs	Substituted variables     Substituted variables     Permenent substitution     Permenent substitution     delete explicit equation that     that     streams     other equations and where t     ble/constant to the explicit v     iables (explicit in a equation t	Explicit/substituted variables that can't be further substituted he explicit equation differ from ariable o iterate on) of these conditions are satisfied
No. of parameters           1+5"Nc           No. of known variables           6+2"Nc           No. of explicit variables           10+5"Nc           No. of var. to specify           0	Unknown variables T_Shell1	Known variables TT_1 P_Shell1 Alpha_3 TT_2 f_2[1:Nc] Area_Shell1 ct_Shell1 f_1[1:Nc]	Parameters Nc ADippr100_Liq[1:Nc] BDippr100_Liq[1:Nc] CDippr100_Liq[1:Nc] CDippr100_Liq[1:Nc] EDippr100_Liq[1:Nc]
E Valuate DUP strings     Include sub, level in DOF     Use symbols     Contain undef, comp.     Contain undef, reac.     View all variables	Differential - dependent	Known differential	Differential - independent
Add non-used variables Close	>> Double click on variable Variables in '()' are assumed	es to perform specific variable d set by base models, i.e., the	substitution/assignment << y are not included in the DOF

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).

Catable types         Equation types           © Total         © Total           © Mass         © Mass           © Momentum         © Energy           © Momentum         © Geometric           © Other         © Other	Equations and variables to Parameters Independent differential Substituted var. + assoc Explicit var. + associate Equations with only one	remove from matrix— Ivar. ciated eq.IV Known deq. ☐ 0DEs' s unknown varariable	var. explicit eq. + asso + their differential c together with the :	ciated explicit var. lependent var. single variables	Pretubation type- C Left C Centre Pertubation length CuttOffValue Column width	al © Right 0.001 = 0 = 600 =
Equation            Equation         x_shell [n_shell n_shell n_	Height         H_3         Hi_3()         f.2           2         1         2         2           2         0         0         0           -2         0         0         1           0         0         1         0           0         0         0         0         0           0         0         0         0         0           0         0         0         0         0           1         0         0         0         1           -2         0         0         2         -2         -2           -2         2         -2         -2         -2         -2	30    t_3   T_Sh 2 2 1 0 2 0 0 -2 0 2 0 0 0 0 0 0 0 0 -2 0 2 2 2	ell TT_3 t -2 0 0 0 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • •	Urdel equations	Close

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.

Image: Internet         Normal Model         Window         Help           Image: Imag	ModDev - [model1b]	
Image: Shart         NoBaseModel         Image: Shart         Image: Shart </th <th>🕻 File Edit View Draw Format Model Window</th> <th>Help</th>	🕻 File Edit View Draw Format Model Window	Help
1       2       ft_1=sum_(t_1)       3         1       3       H_1_rAD(ppr100_Liq_2TT_1*2+CD(ppr100_Liq_3*TT_1*3+DD(ppr100_Liq_4*TT_1*4+ED(ppr100_Liq_5*TT_1*5)         1       3       H_1_rAD(ppr100_Liq_2*TT_1*2+CD(ppr100_Liq_3*TT_1*3+DD(ppr100_Liq_4*TT_1*4+ED(ppr100_Liq_5*TT_1*5)         1       5       ft_2=sum_(t_2)         6       H_2_sAD(ppr100_Liq_2*TT_2*2+CD(ppr100_Liq_3*TT_2*3+DD(ppr100_Liq_4*TT_2*4+ED(ppr100_Liq_5*TT_2*5)         1       H_2=sum_(t_2)       6         6       H_2=sum_(t_2)_7       7         7       H_3=sum_(t_2)_ft_3       7         8       ft_3=sum_(t_2)_sOft       7         9       ft_3=sum_(t_1)_sOft       7         10       H_3=sum_(t_3)_ft_3       7         11       TT_3=T_Sheft       10         12       P_3=P_Sheft       13         13       x_Sheft_en_Sheft       13         14       rt_3=sheft_et_Sheft       14         15       Height_Sheft_et_Sheft       14         16       % (n_Sheft_et_Sheft)       13         17       H_3=sum_(t_3,t_3)/ft_3       16       % (n_Sheft_et_2,t_3)         16       % (n_Sheft_et_2,t_2,t_3)H_3       16       % (n_Sheft_et_2,t_2,t_3)H_3	🗅 🗃 🛱 🛃 🔟 NoBaseModel 🛛 🝷	
		Non-With and the state over):         Important product (not provide the state over):           11         TT_3=T_Shell         TT_3=T_Shell           10         H_3=ADippr100_Liq/STT_3=EDippr100_Liq/STT_3=Shell         Liq/STT_3=Shell           11         H_3=ADippr100_Liq/STT_3=EDippr100_Liq/STT_3=Shell         Liq/STT_3=Shell           12         P_3=P_Shell         Model equations (requarks to reate over):           11         TT_3=T_Shell         Liq/STT_3=Shell           13         X_Shell=n_Shell         Liq/STT_3=Shell           14         H_3=ADippr100_Liq/STT_3=EDippr100_Liq/STT_3=Shell         Liq/STT_3=Shell           15         H_3=ADippr100_Liq/STT_3=EDippr100_Liq/STT_3=Shell         Liq/STT_3=Shell           13         X_Shell = n_Shell (cl_Shell*Area_Shell)         Liq/STT_3=Shell           14         H_3=ADippr100_Liq/SHT_3=Shell         Liq/SHT_3=Shell           15         Height_Shell=nt_Shell (cl_Shell*Area_Shell)         Liq/SHT_3=Shell           16         R_3=ADipa_3=agt(Height_Shell)         Liq/SHE_3           17         H_3=um_[H_3^1_4]/H_3/H_3         Liq/SHE_3           18         R_3=ADipa_3=agt(Height_Shell)         Liq/SHE_3           19         Liq/SHE_3         Liq/SHE_3           19         Liq/SHE_3         Liq/SHE_3           19

Figure 13c: Final model form.