TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Masterarbeit

Development of thermal separation units and a tutorial for flowsheeting applications in gPROMS

ausgeführt zum Zwecke der Erlangung des akademischen Grades einer

Diplom- Ingenieurin

am Institut für Thermische Verfahrenstechnik der TU Wien,

unter der Leitung von

Ao. Univ. Prof. Univ. Prof. Dipl.-Ing. Dr. techn. Anton FRIEDL,

und

Dipl. Dipl.-Ing. Orestis ALMPANIS LEKKAS,

ausgeführt durch

Simone Christa

0625751

1180 Wien

Wien, am 12/01/2015

Abstract

The target of this work is the development of a distillation column model for the separation of binary mixtures in gPROMS.

gPROMS is an equation-oriented modelling and simulation software. It combines modelling capabilities with a process flowsheeting environment and is applicable for both steady-state and dynamic simulations. The model was implemented in gPROMS due to its flexibility regarding the programming structure and its capability to perform flowsheeting simulations.

To determine the initial operating conditions for a given separation problem, a shortcut distillation model was developed. The WUG-model is based on the Winn-Underwood-Gilliland shortcut correlations. The results obtained from this model serve as initial values for the precise column model.

The steady state column model is based on mass and energy balances. It is designed as an n-equilibrium stage model including a total condenser, a splitter and a reboiler. For the calculation of the equilibrium state in each stage, the thermodynamic properties (e.g. enthalpies, activities) of Multiflash are used. For the consideration of the deviation from the equilibrium state, the Murphree efficiency can be defined. Additionally, there is the opportunity to specify a column pressure drop. The challenging part in modelling such processes is the complex system of equations, since the balances and equilibrium conditions form coupled non-linear matrices. For handling this numerical challenge, an initialisation procedure with regard to both, model robustness and flexibility in user defined specifications, was implemented.

For the validation of the models two binary systems, water/methanol and water/ethanol, were investigated. The developed models are used in flowsheet simulations and the simulation results are validated and cross-referenced with literature data. The data obtained by Multiflash and both the WUG and the column model show satisfactory correspondence with literature data.

Additionally, a tutorial for gPROMS was developed. This should help beginners to get started and to gain basic knowledge about programming simple unit operations for flowsheeting applications within the gPROMS environment.

Kurzfassung

Das Ziel dieser Arbeit ist die Entwicklung eines Destillations-Kolonnenmodells zur Trennung von binären Gemischen in gPROMS.

gPROMS ist eine gleichungsorientierte Modellierungs- und Simulationssoftware, in der die Modellierungsebene mit der Flowsheeting- Umgebung verknüpft ist. gPROMS ist sowohl für stationäre, als auch für dynamische Simulationen einsetzbar. Das Modell wurde aufgrund der flexiblen Programmstruktur und seiner Fähigkeit zur Durchführung von Fließschema-simulationen in gPROMS implementiert.

Um die Anfangsprozessbedingungen für ein gegebenes Trennungsproblem ermitteln zu können, wurde ein Short-Cut Kolonnen-Modell entwickelt. Das WUG-Modell basiert auf den Korrelationen von Winn, Underwood und Gilliland. Die Ergebnisse aus den Short-Cut Berechnungen dienen als Anfangswerte für das Kolonnenmodell.

Das stationäre Kolonnenmodell basiert auf Massen- und Energiebilanzen. Das n-stufige Gleichgewichtsmodell beinhaltet einen Kondensator (vollständige Verflüssigung), einen Stromteiler und einen Reboiler. Die Stoffdaten (z.B. Enthalpien, Aktivitäten) für die Gleichgewichtsberechnungen in jeder Stufe werden dabei aus Multiflash abgerufen. Durch das Spezifizieren des Stufenwirkungsgrades (Murphree Efficiency) können Abweichungen vom Gleichgewichtszustand mitberücksichtigt werden. Zusätzlich gibt es die Möglichkeit einen Druckverlust zu definieren. Da die Bilanzen und Gleichgewichtsbedingungen zusammenhängende, nicht-lineare Matrizen bilden, ist die herausfordernde Aufgabe bei der Modellierung von solchen Prozessen, die Lösung der komplexen Gleichungssysteme. Um diese numerische Herausforderung zu bewältigen, wurde eine Initialisierungsstrategie entwickelt, die sowohl für die Stabilität, als auch für die Flexibilität für benutzerdefinierte Eingaben sorgen soll.

Für die Validierung der Modelle wurden zwei binäre Gemische, Wasser/Methanol und Wasser/Ethanol, untersucht. Dafür wurden die entwickelten Modelle in einer Fließschema-Simulation angewendet und die Ergebnisse mit Literaturdaten verglichen. Es zeigt sich, dass sowohl die Stoffdaten von Multiflash, als auch die Ergebnisse beider Modelle, WUG und Kolonnenmodell, gut mit den Literaturdaten übereinstimmen.

Zusätzlich wurde ein Tutorial für gPROMS entwickelt. Dies soll als Einstiegshilfe für Anfänger_innen dienen und das Basiswissen zum Programmieren von einfachen Grundoperationen für Flowsheeting-Anwendungen in der gPROMS Umgebung vermitteln.

Acknowledgements

First of all, I would like to thank Prof. Anton Friedl for giving me the opportunity to realize this work.

I would like to thank Dr. Walter Wukovits for his scientific support and giving me the opportunity to work on my thesis at their location at TU Vienna. I also want to mention the support I got from the whole working group for process simulation and for helping me with Aspen Plus.

I'm especially grateful for my supervisor Orestis Almpanis Lekkas, for all the time he sacrificed for me, for his scientific and personal support, for the nice discussions and his patience while developing this thesis.

I want to acknowledge the Land Vorarlberg for the financial support I received for my studies abroad. In retrospect, I would like to warmly thank my friends and flatmates in Berlin, who made this great time possible.

Furthermore I would like to thank all my friends, especially Pez, who were on hand with help and advice and supported me whenever needed.

Finally, I'm especially grateful for my parents, Josef and Renate, who always supported me personally and as well financially and who gave me the opportunity to acquire a broad education. Not forgetting my sister Babsi for her steady support during my studies.

Table of Contents

1 Introduction

Distillation is one of the most important thermal separation technologies. The process is closely connected to the production of alcoholic beverages. Besides different forms of alcohol production, the Egyptians for instance prepared essential oils by heating up plants (pure or in combination with water) in an earthenware vessel. The rising vapour was condensed and collected in a packing of wool. The produced oil could be won easily by pressing out the wool package. Due to limitations concerning the product purity by the application of a simple distillation process, it was necessary to develop multistage processes with as little apparatus and energy demand as possible. The rectification concept with counter-current flow of the liquid and vapour phase within one column emerged later and became public in 1817. Rectification leads to high distillate purities, where only minor instrumentation is required. Additionally the heat demand can be kept as low as possible [Behr, 2010], [Forbes, 1970] and [Stephan, 2010]. Today, nearly every product includes components which went through a distillation process. The increase of the global energy demand and the application of distillation in petrochemical industry led to enormous development and tens of thousands operating columns worldwide. Besides the fuel and oil industry it is applied for producing alcoholic beverages, to isolate air components or for the desalination of water. Furthermore the trend to renewable fuels leads to aqueous, mainly ethanol and methanol mixtures that require distillations [Kiss, 2013]. Due to increasing requirements on yields, product purity and economical feasibility efficient models to analyse and predict system performances are required. As investigations with real processes are time consuming and complex, process simulation is used for thorough analysis.

The development and availability of high-speed computers that provide the possibility to design and optimise physical, chemical and biological processes led to an extreme extension of the application of process simulation in the last decades. High performance processors make it possible to calculate large, nonlinear problems in a fast, convenient and cheap way. New advanced software packages and efficient numerical algorithms serve as basis for application-oriented simulations and for fast solutions of complex differential equation systems. Process simulation can be applied multidisciplinary and has become state-of-the-art. Beneath the possibility to analyse the behaviour of process-variations and developing existing processes, it can be used for the prediction of system performances and the detailed design of apparatus for new processes. Additionally it is a tool for ensuring safety and can be applied for the automation and control as an instrument for quality assurance. In regard to economic factors it is a tool for the optimisation of energy and material efficiency as well as for cost estimations and minimisations. Process simulation has proved to be very successful and has therefore become an indispensible engineering tool [Friedl et al., 2010] and [Ramirez, 1997].

The target of this thesis is to develop a distillation column model for binary mixtures within the environment of gPROMS. gPROMS was selected due to the flexibility in the programming structure and the flowsheeting capability that makes it possible to build processes with other custom-made unit operation models.

Following this introduction (chapter 1), the general principles of distillation are described in section 2.1. The section 2.2 gives a brief overview about the vapour-liquid phase equilibrium of ideal and real mixtures, about its calculation and graphical representation methods. Additionally, the activity coefficient models and the Murphree stage efficiency are discussed. Section 2.3 proceeds with estimation methods for initial process parameters. Therefore the graphical McCabe-Thiele and the short-cut calculation methods are described. Chapter 3 treats the topic of process simulation. In the general part in section 3.1 the subdivisions depending on time behaviour and solution strategies of simulation software are itemized. A brief discussion about numerical solvers and the degrees of freedom follows. Section 3.2 continues with introducing gPROMS with a focus on the hierarchical structure of submodels in 3.2.1. Following the description of the steady state solvers, section 3.2.3 gives an overview of providing physical properties and Multiflash. General aspects of the process initialisation procedure are discussed in 3.2.4. Chapter 4 describes the models implemented in gPROMS. Section 4.1 depicts all the models needed for the flowsheeting applications developed and implemented in gPROMS. The main models, WUG and the column model, are described in the sections 4.1.10 and 4.1.11. In chapter 5 the designed process initialisation procedure is discussed. It includes the implementation of the initial state correlations (section 5.1), the final state consideration (section 5.2), the variable preset and the adjustment for the application of the components ethanol and water (section 5.4). Chapter 6 summarises the simulation results. First, the physical and thermodynamic properties obtained by Multiflash are cross referenced with literature data. A second investigation shows the comparison of the shortcut models WUG and Aspen Plus' DSTWU. To validate the column model, flowsheeting applications with two different binary mixtures are cross referenced with simulation results of Aspen Plus. Finally, chapter 7 gives a summary of the current work and a short outlook.

A tutorial for gPROMS users has been developed as a second part of this thesis and is attached in the appendix. It should help beginners to get started and to familiarise themselves with gPROMS. A detailed guide and simple examples have the intention to provide an overview of the capabilities and functionalities of the gPROMS environment. The user should gain a basic knowledge about programming simple unit operations, creating a flowsheet in gPROMS, executing a simulation and viewing results without any previous knowledge or additional support.

2 Theoretical background of distillation

2.1 General principles

Distillation is a thermal process for separating liquid mixtures based on boiling points.

Applying heat to a liquid mixture leads to the formation of a vapour phase in addition to the liquid one. During distillation the liquid and vapour phases come into close contact and due to differences in the chemical potentials of the components, mass transfer takes place until the equilibrium state is reached.

The result of the equilibrium state is that the vapour phase gets richer with lower boiling components (light key components), while the higher boiling components (heavy key components) concentrate in the liquid phase. The separation is only possible if the boiling temperatures and the relative volatilities of the components are different. In a simple distillation the vapour is separated from the boiling mixture and condensed afterwards. In order to reach an effective separation, countercurrent distillation or rectification is used instead.

In rectification the two phases, liquid and vapour, are conveyed in counter-flow within columns with a defined number of stages. The liquid phase gets richer in heavy key components by running down the column whereas the vapour phase becomes richer in light key components as it rises up. The vapour leaves the column at the top where it is condensed and split into the distillate product and the reflux. The reflux is returned to the column. The bottom flow is partly evaporated. The produced vapour is recirculated while the liquid is withdrawn as bottom product. In order to carry out this separation process, heat has to be provided for the partial evaporation of the bottom stream where the temperature is highest. To condense the vapour distillate stream, heat has to be dissipated. The temperature in that case is lowest. Figure 1 shows the basic design of a continually operating distillation column with one feed stream.

Through the counter-flow within the column, both streams are in intense contact on all separation stages throughout the column. After the mass and energy transfer on each stage, the liquid and vapour phases reach the state of equilibrium [Friedl, 2000], [Perry, 1997] and [Schönbucher, 2002].

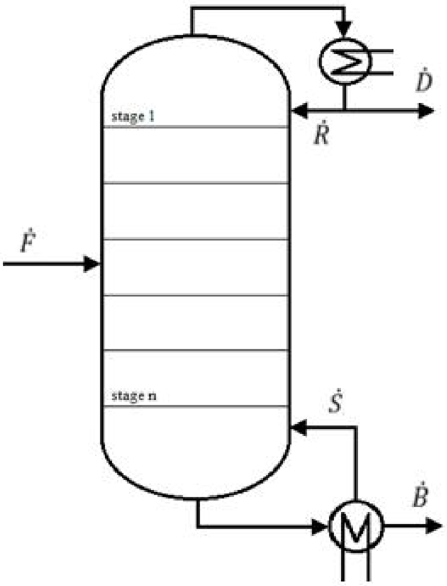


Figure 1: Design of a distillation unit

For modelling the energy and mass transfer within a column and to determine the number of the required stages to reach a defined separation the equilibrium stage concept is used. It assumes that the phase contact between the vapour and liquid streams in a column stage is long enough to establish phase equilibrium. Furthermore, there is no entrainment of liquid into the vapour phase and the condensed phase is always separated after reaching the equilibrium. With the equilibrium stage concept it is possible to use thermodynamic relations for the determination of the compositions. Those relations are described in the following subchapters [Friedl, 2000], [Perry, 1997] and [Schönbucher, 2002].

2.2 Phase equilibrium

The VLE (vapour-liquid equilibrium) characterizes the phase equilibrium in every single stage of a distillation column. There are two approaches the one for ideal and the one for real mixtures. Both enable the calculation of the equilibrium compositions at defined pressure and temperature [Schönbucher, 2002]. The pressure needs to be assigned and the temperature is obtained from the equilibrium conditions.

2.2.1 Ideal mixtures

Ideal mixtures are described by the interaction of their components. The forces of attraction between heterogenous or homogenous molecules are of the same magnitude. Since there are no volume effects or heats of mixing, the mixing enthalpy and volume are zero. For the assumptions above, the following equation applies for the VLE:

$$f_{iL} = f_{iV} \qquad 2.1$$

The fugacities of the liquid and vapour phases are the same for every component i. In the case of ideal mixtures, Raoult's law can be used:

$$y_i \cdot p = p_i = p_{0,i} \cdot x_i \qquad 2.2$$

In the equilibrium state, the partial pressure p_i of a component i is proportional to the saturated pure component vapour pressure $p_{0,i}$ multiplied by the liquid molar fraction x_i. Whereby, the vapour pressure depends on the equilibrium temperature [Friedl, 2000], [Sattler, 1988] and [Schönbucher, 2002].

2.2.2 Real mixtures

Real mixtures are characterized by the deviations of the interacting forces between heterogenous and homogenous molecules. If the interacting forces between heterogenous molecules are higher than the ones between the homogenous, the vapour pressure is smaller than the one from the ideal mixture. Otherwise, if these forces are smaller, the vapour pressure gets higher. To describe real mixtures, Raoult's law is extended by the vapour fugacity φ_i^V, for the adaption of the vapour phase, and by the liquid activity γ_i, for adjusting the liquid phase:

$$\varphi_i^V \cdot y_i \cdot p = \gamma_i \cdot x_i \cdot p_{0,i} \qquad 2.3$$

This equation suits pretty well for VLE calculations at low pressures, but is inapplicable for supercritical ranges and calculations near the critical point. For calculations at higher pressures and near the critical point, the equation can be adjusted with fugacities for both, the liquid and vapour phases [Friedl, 2000], [Sattler, 1988] and [Schönbucher, 2002].

$$\varphi_i^V \cdot y_i = x_i \cdot \varphi_i^L. \qquad 2.4$$

The fugacity and activity coefficients depend on equilibrium temperatures, pressures and vapour/ liquid molar fractions.

The activity coefficient is a measure for the deviations of the ideal behaviour and depends strongly on the concentrations. There are different approaches to describe the correlations of the activity with the temperature, concentration and pressure. Those activity models are discussed in chapter 2.2.5 [Lüdecke et al., 2000].

2.2.3 Relative volatility

The volatility α is a measure of the selectivity of a separating stage. It is defined as the ratio of the *K*-values of two species i and j:

$$\alpha_{i,j} = \frac{K_i}{K_j} \qquad 2.5$$

The distribution coefficient *K* is the ratio of the vapour and appendant liquid composition of a component i:

$$K_i = \frac{y_i}{x_i} \;. \qquad 2.6$$

For ideal mixtures, the *K*-values correspond to the vapour pressures of the pure components. For the calculation of the distribution coefficients and the volatility, thermodynamic data for pure components can be used [Friedl, 2000] and [Schönbucher, 2002].

To separate a mixture the volatility has to be larger than one. The separation is getting better the higher the value of α. If one composition with $\alpha_{i,j} = 1$ appears, it is called an azeotropic point. Azeotropic mixtures can’t be separated within simple rectification. However, most of the mixtures are non-azeotropic. That means that one component is more volatile than the other over the whole composition range. Those mixtures can be separated into nearly pure components by simple separation processes. In the phase diagrams for binary systems (shown in 2.2.4) it is clearly visible, which component is more volatile and whether the mixture is azeotropic or not [Perry, 1997].

2.2.4 Equilibrium data and phase diagrams for binary systems

The following considerations only concern binary systems. Experimentally determined equilibrium data for binary systems are available in tables. The most common tables present the liquid and vapour mole fractions, x and y, of one component at a fixed pressure over a temperature range [Perry, 1997].

Table 1 shows the liquid-vapour equilibrium data of the methanol/water mixture at a fixed pressure.

Table 1: Constant pressure liquid-vapour equilibrium data for methanol/water [Perry, 1997]

Temperature [K]	Mole fraction methanol in		Total pressure [kPa]
	Liquid	Vapour	
373.15	0.000	0.000	101.3
369.55	0.020	0.134	
366.65	0.040	0.230	
364.35	0.060	0.304	
362.45	0.080	0.365	
360.85	0.100	0.418	
357.55	0.150	0.517	
354.85	0.200	0.579	
351.15	0.300	0.665	
348.45	0.400	0.729	
346.25	0.500	0.779	
344.35	0.600	0.825	
342.45	0.700	0.870	
340.65	0.800	0.915	
339.15	0.900	0.958	
338.15	0.950	0.979	
337.65	1.000	1.000	

There are various possibilities for the graphical representation of phase equilibrium data. Two of them are shown and described below.

Phase equilibrium diagram

In the phase equilibrium diagram, the vapour equilibrium compositions y, of the light key component, are plotted in dependence of their liquid equilibrium compositions x at a defined pressure. As a reference, the 45° line is shown. The phase diagram for the system methanol/water on the left is a non-azeotropic mixture. The other two illustrations, 1-propanol/water in the middle and aceton/chloroform on the right, are examples for azeotropic mixtures. Due to the temperature minimum of the 1-propanol/water mixture (see figure 3), it is called minimum azeotrope. The aceton/chloroform mixture is an example for a maximum azeotrope. The azeotropic point is where the equilibrium line crosses the diagonal one. In that point, the vapour and liquid compositions are equal. In the mixture 1-propanol/water, 1-propanol is more volatile until a concentration of approximately 42 mole percent, after that point, the volatility is reversed. The illustration on the right hand shows, that aceton is less volatile until 36 mole percent and afterwards more volatile than chloroform.

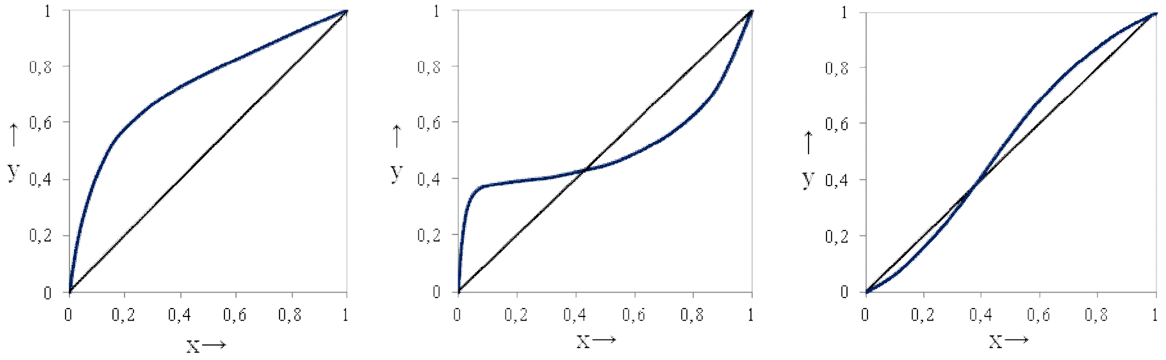


Figure 2: Phase equilibrium diagrams for binary mixtures at 1.103 kPa: water/methanol (left), 1-propanol/water (middle), aceton/chloroform (right handside); data from [Perry, 1997]

Temperature diagram

The temperature diagram shows the boiling and dew temperatures dependent on the compositions of the light key component at a fixed pressure. In the 1-porpanol/water figure in the middle it is visible that the boiling- and dew temperatures reach a minimum (minimum azeotrope). Contrary to this, the mixture aceton/chloroform reaches a temperature maximum (maximum azeotrope) [Schönbucher, 2002].

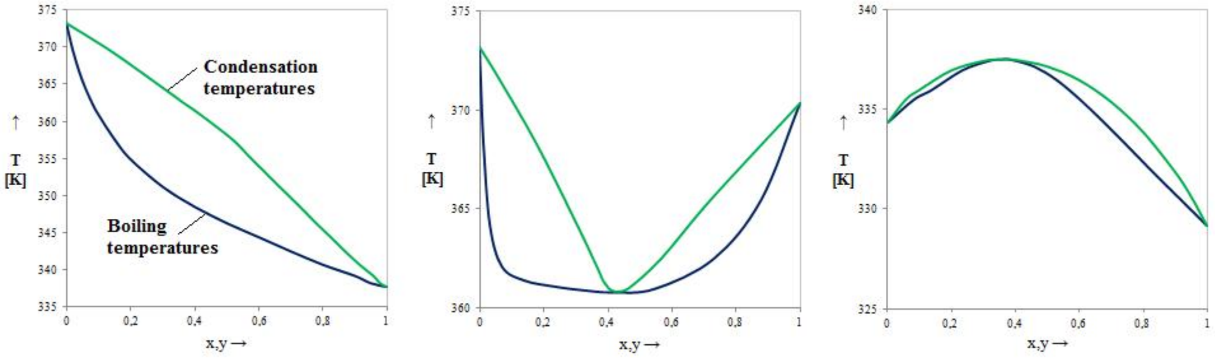


Figure 3: Temperature diagrams for binary mixtures at 1.103 kPa: water/methanol (left), 1-propanol/water (middle), aceton/chloroform (right handside); data from [Perry, 1997]

2.2.5 Activity coefficient models

Activity coefficient models are equations which describe the dependence of the activity coefficient on the composition, temperature and pressure. Since the deviations from the ideal mixture are different for every compound type, multiple models were developed on the basis of experimental data. A few of them are discussed in this chapter with a focus on their suitability for different compound systems [Lüdecke et al., 2000].

Margules

The Margules exposition is available with two or three parameters. It brings good results for systems, where the molecules of the components are approximately the same size and do not associate with each other. It is not applicable for strongly non-ideal systems and for multi-component mixtures.

Van Laar

The Van Laar equations are suitable for simple, non-polar as well as for more complex binary mixtures and therefore often applied in praxis.

Wilson

Wilson's model contains only two adjustable parameters and is applicable for strongly non-ideal mixtures. It can be used for binary or multi component VLE calculations at small or medium pressures, but not for electrolyte systems or systems with miscibility gaps (LL or VLL phase equilibriums).

NRTL

NRTL is the short form for "Non-Random-Two-Liquid". The model describes two hypothetical liquids A and B. Liquid A is built up with central-molecules from the component A and surrounded by molecules of both types A and B. The same applies for the second hypothetical liquid B, just reversed. The theory is based on the assumption that molecules prefer their own or the one from the second liquid as direct neighbours because of different pair interactions. That means that the neighbour molecules are not randomly arranged. The NRTL equation for binary systems contains three adjustable parameters. With regard to the theories of Margules and Van Laar, the NRTL approach has the advantage that it is suitable for strongly non-ideal mixtures and in contrast to the Wilson equation it describes as well mixtures with miscibility gaps. It allows therefore the calculation of VLE and LLE at small and medium pressures of strongly non ideal mixtures and can be used for multi component mixtures.

UNIQUAC

The "Universal Quasi-Chemical Theory" is based on the Guggenheim theory for mixtures of molecules of the same size. The molecules have a certain order and their arrangement is described as a quasi-crystal grid, as the density properties of liquids are more identical to solids than to gases. Since the molecules of the components have different sizes and shapes, they are divided into equal segments which occupy a lattice site where their freedom of movement is limited. The model consists of two parts, one that describes the order and structure of the molecules, depending on their composition, shape and size, the second one describes their intermolecular interactions. The UNIQUAC model needs compared to the NRTL model, only two parameters. It can be used for systems with polar and non-polar components, for mixtures that strongly differ from ideality and as well for mixtures with miscibility gaps. The UNIQUAC equation is valid for calculations at pressures that are not too high and can be used for multi component mixtures.

UNIFAC

UNIFAC (“UNIversal Functional group Activity coefficient”) is one of the group contribution methods that allow describing non electrolyte systems without any experimental data. Those methods are based on the principle to split molecules into structure groups. The mixture is then characterized through the existing structure groups and the interactions between them. Whereas the interactions are determined by the structure groups only and not from the molecules surrounding them. The following illustration shows an example.

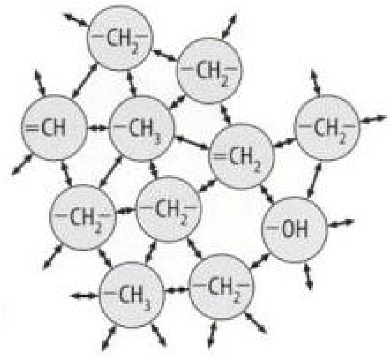


Figure 4: Mixture of the structure groups for a 1-hexanol/buten-(1)-mixture [Lüdecke et al., 2000]

The mixture 1-hexanol/buten-(1) consists of saturated CH_2- and CH_3- groups, unsaturated CH- and CH_2- groups (the ones with double bonds) and one OH-group. According to the theory, the interactions between the OH-group and the saturated CH_2-group are the same for 1-hexanol and buten-(1).

The UNIFAC model is limited for a special temperature range (room temperature to approximately 130 degrees Celsius) and is used for VLE calculations of multi-component mixtures up to 15 bar. Otherwise the temperature dependent interaction parameters need to be determined from experimental data [Lüdecke et al., 2000].

The parameters for the activity models are available in tables (e.g. Dechema Data Series).

2.2.6 Deviation from the equilibrium state

In reality the equilibrium compositions are rarely reached. To describe non-equilibrium behavior the Murphree stage efficiency E is introduced. For the vapour phase the Murphree efficiency is defined as follows:

$$E = \frac{y_{i,n} - y_{i,n+1}}{y_{i,n}^{*} - y_{i,n+1}}. \qquad 2.7$$

The correlation shows that the stage efficiency is defined as the ratio of the actual enrichment of the light key component in the vapour phase to the theoretical (equilibrium) enrichment at a fixed liquid composition. Within the equation, $y_{i,n}^{*}$ is the vapour composition of the component i that is in equilibrium with the liquid molar fraction $x_{i,n}$ (calculated with the equilibrium correlation 2.3 and read from the y-x equilibrium phase diagram). $y_{i,n+1}$ and $y_{i,n}$ are the actual values for the incoming and leaving vapour fractions at stage n. Figure 5 shows a part of the McCabe Thiele diagram (will be discussed in the next chapter) which makes the actual and equilibrium compositions visible [Friedl, 2010].

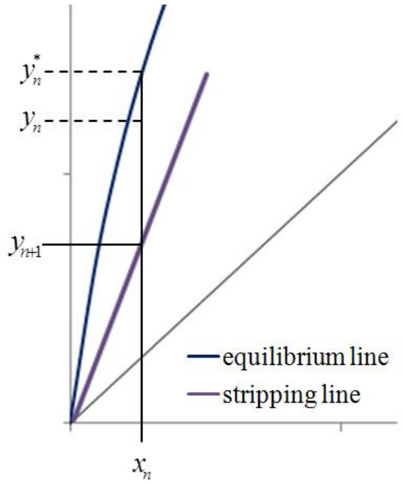


Figure 5: Illustration of the equilibrium and non-equilibrium vapour fractions

2.3 Distillation column design

In general separation processes such as the distillation, the goal is to reach predefined specifications of the distillate product stream, whereas the feed stream is known. Therefore suitable initial values for the basic design are needed, which can be determined with different estimation and design methods. Compared to the short-cut methods for the process parameter estimation, the McCabe Thiele diagram is a detailed method to obtain design specifications.

2.3.1 McCabe-Thiele diagram

The McCabe-Thiele diagram is a graphical method to determine the minimum number of equilibrium stages and the minimum reflux ratio. Additionally it can be applied to obtain the number of stages and the feed stage location for a given separating task. It is valid for binary mixtures and is based on material balances being represented by operating lines in the y-x phase equilibrium diagram (see 2.2.4). The most important assumptions for the use of the McCabe-Thiele method are:

- constant molar overflow
- equal molar heats of vaporization for the two components (Trouton rule)
- no heat loss
- no heat of mixing [Friedl, 2000] and [Perry, 1997].

The description of the McCabe-Thiele method is presented with an example. Table 2 shows the specifications for the separation of the binary methanol/water mixture:

Table 2: Process parameters of the water/methanol separation

Parameter	unit	value
Concentration methanol in feed	[wt%]	5
Concentration methanol in distillate	[wt%]	95
Recovery methanol	[%]	90
Feedflow	[kg/h]	10000
Temperature feed after preheating	[K]	313

Determination of the minimum number of equilibrium stages

The minimum number of stages, with infinite reflux ratio, can graphically be determined by drawing the equilibrium stages as steps between the vapour liquid equilibrium curve of the more volatile component and the 45 degrees line. Figure 6 shows the procedure for the system methanol/water. The horizontal lines represent the equilibrium stages. To reach the required specifications (molar compositions in the distillate and bottom), as many horizontal lines have to be drawn till they cut with the vertical composition lines [Friedl, 2000] and [Perry, 1997].

In this example the minimum number of stages is five.

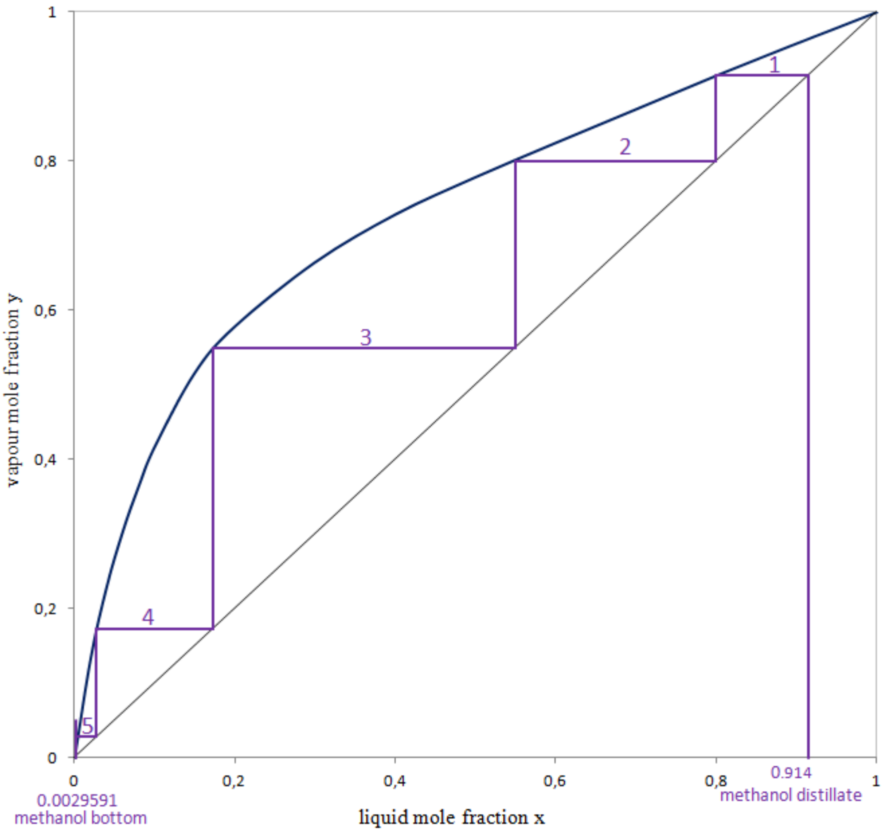


Figure 6: McCabe Thiele method to find minimum number of stages

Determination of the number of theoretical stages and the feed stage location

The required number of theoretical stages and the feed stage location can be specified through the operating lines of the enriching and stripping section and the intersection line. For the rectification part (enriching section), illustrated in figure 7 the overall balance is:

$$\dot{G}_n = \dot{D} + \dot{L}_{n-1} \qquad 2.8$$

and for any component i:

$$\dot{G}_n \cdot y_n = \dot{D} \cdot x_D + \dot{L}_{n-1} \cdot x_{n-1} \qquad 2.9$$

If the reflux Ṙ is sent back as boiling liquid:

$$\dot{R} = \dot{L}_{n-1} \qquad 2.10$$

The reflux ratio *v* is defined as

$$v = \frac{\dot{R}}{\dot{D}} = \frac{\dot{L}_{n-1}}{\dot{D}} \qquad 2.11$$

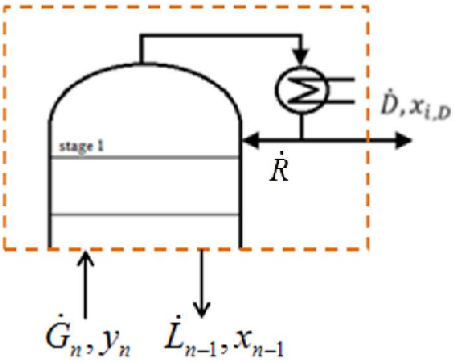


Figure 7: Balance of the rectification part

With equations 2.8- 2.11 the enriching line gets:

$$y = \frac{x_D}{v+1} + \frac{v}{v+1} \cdot x \qquad 2.12$$

x_D is the distillate molar fraction of the light key component. For the stripping part, shown in figure 8 the balance is:

$$\dot{G}_n^+ + \dot{B} = \dot{L}_{n-1}^+ \qquad 2.13$$

and for any component i:

$$\dot{G}_n^+ \cdot y_n + \dot{B} \cdot x_B = \dot{L}_{n-1}^+ \cdot x_{n-1} \qquad 2.14$$

The stripping reflux ratio v^+ is defined as

$$v^+ = \frac{\dot{L}_{n-1}^+}{\dot{B}} \qquad 2.15$$

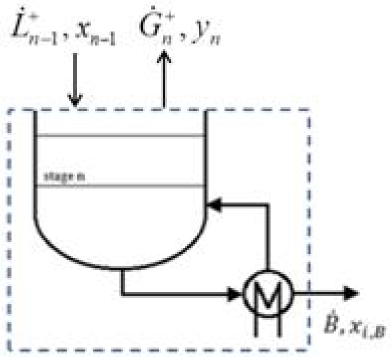


Figure 8: Balance of the stripping part

The equations 2.13- 2.15 lead to the stripping line:

$$y = \frac{v^+}{v^+ - 1} \cdot x - \frac{x_B}{v^+ - 1} \qquad 2.16$$

Since the liquid molar flow $\dot{L}_{n-1}^+$ of the stripping part and therefore its reflux ratio v^+ are not known yet, the intersection line is needed. It can be determined by balancing the feed stage:

$$\dot{F} + \dot{G}^+ + \dot{L} = \dot{G} + \dot{L}^+ \qquad 2.17$$

$$\dot{F} \cdot x_F + \dot{G}^+ \cdot y_{n+1} + \dot{L} \cdot x_{n-1} = \dot{G} \cdot y_n + \dot{L}^+ \cdot x_n \qquad 2.18$$

Additionally, the caloric factor q, which describes the thermal condition of the inlet mixture is defined as

$$q = \frac{\dot{L}^+ - \dot{L}}{\dot{F}} \qquad 2.19$$

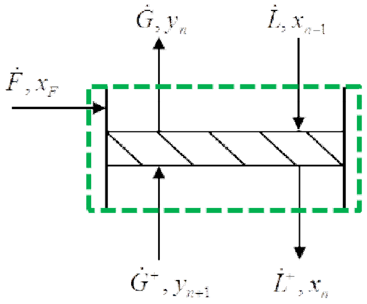


Figure 9: Feed stage balance

The relations 2.17-2.19 lead to the intersection line with x_F as feed molar fraction of the lower boiling component:

$$y = \frac{q}{q-1} \cdot x - \frac{x_F}{q-1} \qquad 2.20$$

The intersection, stripping and enriching line have one common intersection point. The operating part of the column is specified as the area limited through the phase equilibrium curve, the enriching and stripping line. The theoretical number of stages can be determined by drawing stages step by step from the distillate fraction to the bottom fraction in the operating area [Schönbucher, 2002]. As it can be seen in figure 10, 12 theoretical stages are needed to reach the required specifications. The optimal location to add the feed is where the intersection line crosses one of the stages (horizontal line). In that example the feed stage is on stage six.

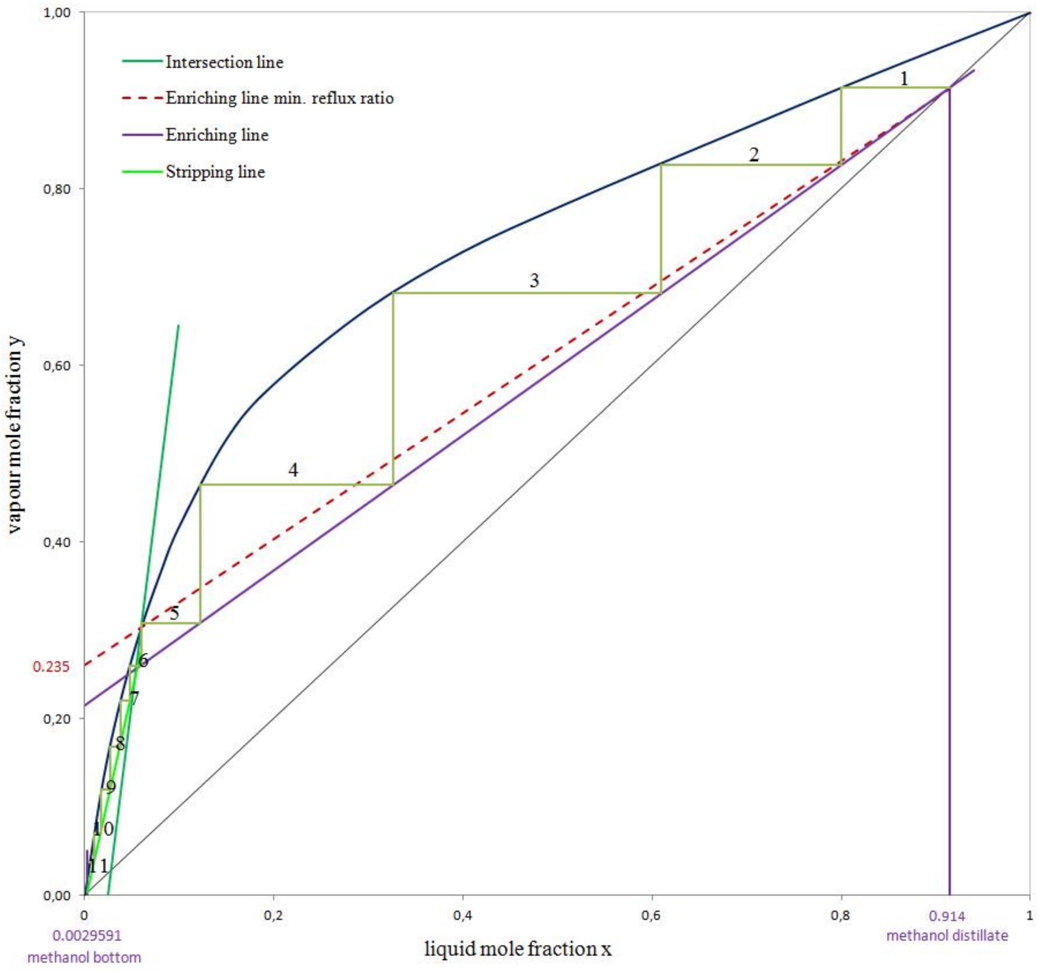


Figure 10: McCabe Thiele diagram

Determination of the minimum number of stages

The intersection line crosses the equilibrium curve once (see figure 10). By connecting this point with the point of intersection of the enriching line at $x = x_D$ with the diagonal, and extending it to the y-axis it is possible to calculate the minimum reflux ratio with the value at the ordinate axis.

$$y(0) = \frac{x_D}{v_{min} + 1} \tag{2.21}$$

With 0.235 as y-value the minimum reflux ratio gets 2.89. The minimum reflux ratio is a limited value with n_{th} → ∞. To make a separation process with a finite number of stages possible, the reflux ratio has to be chosen larger than the minimum.

The caloric factor

The caloric factor q describes the thermal condition of the feed mixture. This thermal condition influences the vapour and liquid flows of the stripping and enriching parts. Beneath the already mentioned equation (2.19) another typical correlation defines the factor.

With

$$q = \frac{\dot{L}^{+} - \dot{L}}{\dot{F}} \tag{2.19}$$

and the enthalpy balance for the feed stage

$$\dot{F} \cdot h_F + \dot{G}^{+} \cdot h_{n+1}^{v} + \dot{L} \cdot h_{n-1}^{l} = \dot{G} \cdot h_n^{v} + \dot{L}^{+} \cdot h_n^{l} \tag{2.22}$$

the second correlation can be derived

$$q = 1 + \frac{h_{F,b} - h_F}{\Delta h_{V,F}} \tag{2.23}$$

With $h_{F,b}$ as the liquid molar feed enthalpy at boiling temperature, h_F as actual present feed molar enthalpy and $\Delta h_{V,F}$ as the evaporation feed molar enthalpy. This relation assumes that the vapour respectively liquid molar enthalpies are the same on every stage ($h_{n+1}^{v} = h_n^{v}$, $h_{n-1}^{l} = h_n^{l}$).

The following table gives an overview about the different possible thermal conditions of the feed.

Table 3: Overview of the various thermal conditions of the feed

Thermal condition feed	caloric factor q	Slope of the intersection line
subcooled-liquid	q > 1	> 0
saturated-liquid feed	q = 1	∞
wet steam (partly liquid, partly vapour) feed	0 < q < 1	< 0
saturated-vapour feed	q = 0	0
superheated-vapour feed	q < 0	> 0

Adding the feed as subcooled liquid leads to a magnification of the stripping area whereby a superheated-vapour feed enlarges the enriching part of the column [Friedl, 2000] and [Schönbucher, 2002].

In the methanol/water example the feed is added as subcooled liquid and the caloric factor must therefore be larger than one.

2.3.2 Short-cut methods

Short-cut methods are correlations that allow preliminary estimations of initial values for the subsequent design of columns. The assumptions for the determination of the minimum number of stages, minimum reflux ratio, number of stages and feed stage location are constant molar flows and geometric average values of the relative volatilities for every separating stage. Short-cut methods may not be used for azeotropic mixtures due to the change of the component's volatilities [Schönbucher, 2002].

Determination of the minimum number of stages

Fenske-equation

With the Fenske-equation, the required minimum number of theoretical stages can be determined. Therefore the distillate and bottom molar fractions of the heavy and light key components have to be known.

$$n_{th,min} = \frac{log\ \frac{(x_{HK}/x_{LK})_D}{(x_{HK}/x_{LK})_B}}{log \bar{\alpha}_{HK,LK}} \qquad 2.24$$

The geometric average value for the relative volatility is calculated as the square root of the product from the bottom's and distillate's volatilities [Schönbucher, 2002]:

$$\bar{\alpha}_{HK,LK} = \sqrt{\left(\alpha_{HK,LK}\right)_B \cdot \left(\alpha_{HK,LK}\right)_D} \qquad 2.25$$

Winn-equation

The Winn-correlation is a modification of the Fenske-equation.

$$n_{th,min} = \frac{ln\ \left[\frac{x_{LK,D}}{x_{LK,B}} \cdot \left(\frac{x_{HK,B}}{x_{HK,D}}\right)^{\theta_{LK}}\right]}{\ln \beta_{\frac{LK}{HK}}} \qquad 2.26$$

The two constants θ_{LK} and $\beta_{\frac{LK}{HK}}$ which are evaluated at the top and bottom, at a fixed pressure, consider temperature dependency on volatilities [Aspen Technology, 2012] and [Coker, 2010].

$$ln\,\beta_{\frac{LK}{HK}} = \frac{K_{LK}}{{K_{HK}}^{\theta_{LK}}} \qquad 2.27$$

Determination of the minimum reflux ratio

The 1. Underwood-equation serves to calculate the Underwood-factor ϑ. With the average relative volatilities $\bar{\alpha}$, calculated with equation 2.25, specified molar feed fractions and the caloric factor *q*, calculated with equation 2.23, □ is clearly defined.

$$\sum_{i=1}^{nc} \frac{\bar{\alpha}_{i,HK} \cdot x_{i,F}}{\bar{\alpha}_{i,HK} - \vartheta} = 1 - q \qquad 2.28$$

With the Underwood-factor ϑ which must be in accordance with the following condition

$$1 < \vartheta < \alpha_{HK,LK} \qquad 2.29$$

and known distillate molar fractions, the minimum reflux ratio v_{min} can be determined with the 2. Underwood equation [Schönbucher, 2002]:

$$\sum_{i=1}^{nc} \frac{\bar{\alpha}_{i,HK} \cdot x_{i,D}}{\bar{\alpha}_{i,HK} - \vartheta} = 1 - v_{min} \qquad 2.30$$

Determination of the actual reflux ratio

For the estimation of either the actual reflux ratio *v*, or the number of theoretical stages n_{th} the Gilliland curve can be used. There are different approaches describing the curve, one is the equation of Molokanov et al [Schönbucher, 2002]:

$$\frac{n_{th} - n_{th,min}}{n_{th} + 1} = 1 - e^{\left[\frac{1 + 54.4 \cdot \psi}{11 + 117.2 \cdot \psi} \cdot \left(\frac{\psi - 1}{\psi^{0.5}}\right)\right]} \quad with \quad \psi = \frac{v - v_{min}}{v + 1} \qquad 2.31$$

With the previously calculated minimum number of theoretical stages and minimum reflux ratio, the actual reflux ratio can be determined with a given number of stages. Vice versa, for the definition of the theoretical number of stages, the actual reflux ratio needs to be defined.

In practice the reflux ratio is often selected as

$$v \approx 1.3 \cdot v_{min} \qquad 2.32$$

Determination of the feed stage location

The feed stage location can be estimated through the ratio of the number of rectification (stages above the feed stage) and stripping stages (below the feed stage). Therefore the Winn equation is adjusted between the distillate and feed compositions to obtain the number of rectification stages n_r and between the feed and bottom compositions to calculate the number of stripping stages n_s.

$$n_r = \frac{ln\left[\frac{x_{LK,D}}{x_{LK,F}} \cdot \left(\frac{x_{HK,F}}{x_{HK,D}}\right)^{\theta_{LK}}\right]}{\ln \beta_{\frac{LK}{HK}}} \qquad 2.33$$

$$n_s = \frac{ln\left[\frac{x_{LK,F}}{x_{LK,B}} \cdot \left(\frac{x_{HK,B}}{x_{HK,F}}\right)^{\theta_{LK}}\right]}{\ln \beta_{\frac{LK}{HK}}} \qquad 2.34$$

With the ratio of the equations 2.33 and 2.34 and a specified number of stages $n_{th,}$ which is defined as the sum of the stripping- and rectification part:

$$n_{th} = n_r + n_s \qquad 2.35$$

the feed stage location can be determined [Seider, 2010].

The short-cut correlations discussed in that chapter provide initial values for following adaptations to meet desired specifications. In regard to the graphical McCabe-Thiele design method which obtains detailed design parameters, the short-cut methods only provide estimated values. The advantage of short-cut methods is that they are capable of reducing the effort through computer-based calculations [Schönbucher, 2002].

3 Process simulation

3.1 Basics

Process simulation is the application of process models with the objective to determine parameters of a chemical process. The process model consists of a system of combined mathematical models (unit operations) [Friedl et al., 2010].

On the basis of time behaviour, process simulation can be subdivided into steady state and dynamic process simulation. The steady state process simulation is only valid for systems at defined operating points. To describe transition states between those operating points and also (time dependent) plant dynamics such as start-up and shut-down, dynamic simulation has to be applied. Steady state models are based on a system of algebraic equations whereas dynamic mathematical models are described with differential (partly discontinuous) algebraic equations.

Another classification of process simulation programs follows the solution strategies [Friedl et al., 2010]:

- modular-sequential
- equation oriented and
- mixed approaches.

3.1.1 Modular-sequential and equation oriented programs

In the modular-sequential approach every unit operation is calculated as a single unit and the blocks are solved successively in a certain order. The calculated output values of a unit serve as input values for the following unit operation. This means that the order follows the material flows in the flowsheet. Figure 11 shows a process with one recycle stream. The preheated feed is mixed with the recycle stream. After a reaction, the stream 4 is split into the recycle and the product stream. If the recycle stream is not known, it has to be initialised and calculated as a convergence loop. As a first iteration step, the recycled stream is assumed as zero. After the first calculation, it attains a new value that is added to stream 2. This iteration process and therefore the loop calculation is continued until the results satisfy the defined convergence criteria.

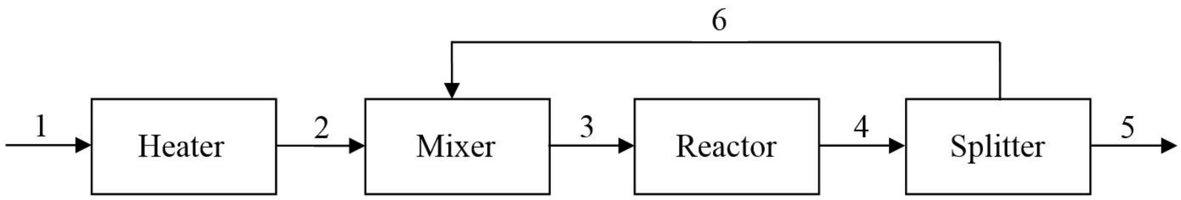


Figure 11: Process with one recycle stream

With the modular-sequential approach it is easy to follow the calculations and to locate errors. Low computing capacity and easy extension possibilities are as well advantages of this solution strategy. The disadvantages are long computing times for complex systems, convergence problems within iterative calculations of recycle streams and the inability to handle dynamic simulations [Friedl et al., 2010]. Examples for modular-sequential flowsheeting software are:

- Aspen Plus
- ChemCAD and
- HYSYS.

In the equation oriented approach a complete process is expressed in form of an *n* equations and *n* variables (*n x n*)-matrix. This system of equations is solved simultaneously.

The equation-solving architecture has the advantage of a very flexible environment for design specifications, where it doesn’t matter if the input- or output streams are specified. Complex systems and optimisation calculations are faster and the treatment of recycles is better due to less convergence problems. In contrast to modular-sequential programs, dynamic simulations are possible. The drawbacks of equation oriented programs are the high programming effort, especially in dynamic simulations and the needs of more computing capacity. Due to the mathematical complexity the handling and debugging are much more difficult. To find solutions for this complex system of equations good initialisation values are needed [Friedl et al., 2010]. Examples for equation oriented software are:

- Aspen Dynamics
- gPROMS
- IPSEpro
- MathLAB/Simulink.

In the mixed approach, the combination of sequential-modular and equation-oriented architectures, initial values for the single operating units are calculated sequentially. Those are used as starting values within the flowsheeting level, which is solved globally on the base of equation-oriented approach.

3.1.2 Structure of process simulation programs

The fundamental element of process simulation software is the executive program which manages computation and data exchange tasks. It is connected with different subroutines:

- Physical properties database
- Subroutines for the calculations of physical and thermodynamical properties
- Equipment subroutines that include database for unit-operations
- Subroutines to control the calculation sequences
- Numerical solvers and convergence follow-up

The (graphical) user interface serves for both, the input of simulation problems and presentation of the simulation results (output) [Friedl et al., 2010].

Numerical Solver and Initialisation

To solve complex systems of nonlinear equations and to converge recycle loops, there are different numerical solution options:

- Procedures based on direct substitution
    - Direct substitution: stable but slow and convergence is not guaranteed
    - Wegstein-method: fast and robust, designer is able to influence the convergence
- Newton and Quasi-Newton methods

Newton-method

The Newton method is based on the following correlation:

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}. \qquad 3.1$$

Figure 12 illustrates the procedure: The function *f(a)* is linearised in a given inital zero-point a_0. The resulting tangent intersects with the x-axis to a new point a_1 which is closer to the real root of the function and serves as new starting point. The process is repeated until a_n reaches a specified level of precision or the convergence criteria.

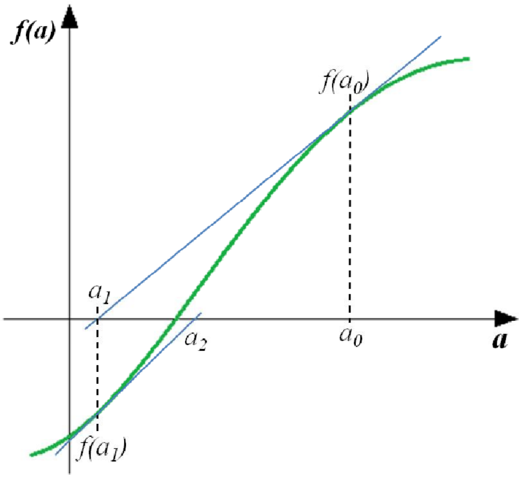


Figure 12: Newton iteration graphical illustration

Further extensions of the basis correlation with the Jacobian matrix are used for simultaneous solutions of nonlinear equation-systems. The Quasi-Newton methods Secant and Broyden apply different adaptations of the Jacobian matrix [Friedl et al., 2010]

Initialisation

The convergence of solution algorithms discussed above is better, faster and more robust if good initial values are estimated. Complex coupled non-linear matrix systems even require user intervention to achieve convergence. Therefore, realistic boundaries for the values of iterative variables and initial values need to be set.

Besides the initialisation values it should be concerned, that the sequence of the equations and the use of more and simpler unit operations can affect the solution behaviour additionally [Friedl et al., 2010].

3.1.3 Degrees of freedom

To execute a simulation, the number of degrees of freedom needs to be defined to completely specify a process and to guarantee that a solution is possible.

The degrees of freedom of a system are defined as the number of unknown independent variables [Lüdecke et al., 2000].

The calculation works as follows:

total number of variables (unknown)	+	
independent equations	-	3.2
degrees of freedom	∑	

A design solution is therefore possible when the total number of unknown variables equals the number of independent equations [Perry, 1997].

For a column stage, shown in figure 13, the degrees of freedom are:

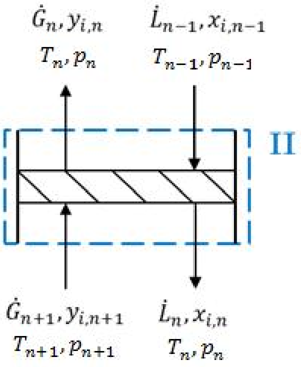


Figure 13: Balance II with stream variables

Table 4: Degrees of freedom calculation for one column stage

Stream variables (4 streams)	**Equations**	
4 x flow	nc x mass balances	
4 x nc molar fractions	nc x equilibrium equations	
4 x temperature	1 x heat balance	
4 x pressure	2 x summation conditions	
	2 x pressure	
	1 x temperature	
4 x (nc + 3)	-(6 + 2 x nc) =	**2 nc + 6**

There are four streams at one stage. Each of them has unknown composition, temperature and pressure. This means, for a binary system, there are 20 unknown variables. The equations for the system include two mass balances, two equilibrium correlations, one heat balance and two summation-conditions. According to the summation conditions, the sums of the liquid and vapour fractions of the components leaving one equilibrium stage *n* have to be one:

$$\sum_{i=1}^{nc} x_{i,n} = 1 \qquad 3.3$$

$$\sum_{i=1}^{nc} y_{i,n} = 1 \qquad 3.4$$

Since the outgoing streams are in equilibrium with each other, the temperatures and pressures are equal (one temperature and pressure equation). Furthermore one pressure correlation is added. Hence there are ten independent variables that must be defined to exactly determine the system. One proposal is to completely define the input streams:

- both input flows
- the compositions of the input streams
- input temperatures and pressures.

If the system is extended with another stream (for instance with one feed stream), the number of equations will stay the same. However there are more stream variables and as a consequence the degrees of freedom will rise. In order to determine all unknown quantities five more independent variables need to be defined (feed flow, compositions, temperature and pressure).

To design a distillation column, the overall balance, shown in figure 14, has to be considered.

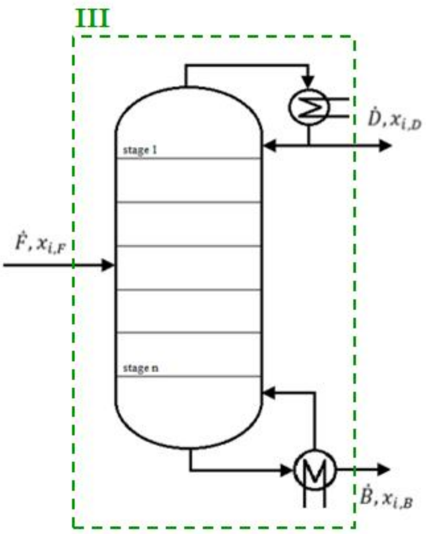


Figure 14: Overall balance distillation column

Taking into account that the feed stream is totally specified and the pressure in the column is known, the degrees of freedom are calculated as follows.

Table 5: Degrees of freedom calculation of the column

Variables (3 streams)	Independent equations	Known variables
3 x flow	nc x mass balance	1 x flow
3 x nc molar fractions	1 x heat balance	1 x nc molar fractions
3 x temperature	2 x summation conditions	1 x temperature
3 x pressure	1 x pressure	2 x pressure
3 x (nc + 3)	-(nc + 4)	-(nc + 4)
		nc + 1 = 3

There are three variables that have to be chosen. One opportunity is to define one output flow and the compositions of one stream.

3.2 gPROMS

gPROMS (**g**eneral **PRO**cess **M**odelling **S**ystem) is an equation-oriented modelling and simulation software developed by Process System Enterprise (PSE). The gPROMS ModelBuilder® combines modelling capabilities with a process flowsheeting environment and is applicable for both, steady-state and dynamic simulation. Other capabilities are advanced parameter estimation and steady-state, respectively dynamic optimisation [Process Systems Enterprise Limited, 1997-2012].

3.2.1 Submodels as base for superior models

In order to manage complex systems of processes, it is necessary to structure systems and divide them into a set of connected smaller and less extensive models. Small models have the advantages that they are easier to write, read and debug. Figure 15 illustrates the hierarchical model structure in gPROMS. The superior model flowsheet can be decomposed in the single unit operations pump, tank and the connections (pipe) between them. The unit operations can again contain one or more hierarchical sublevels. The interconnection of many primitive models in a set of intermediate hierarchical levels results in a complex process.

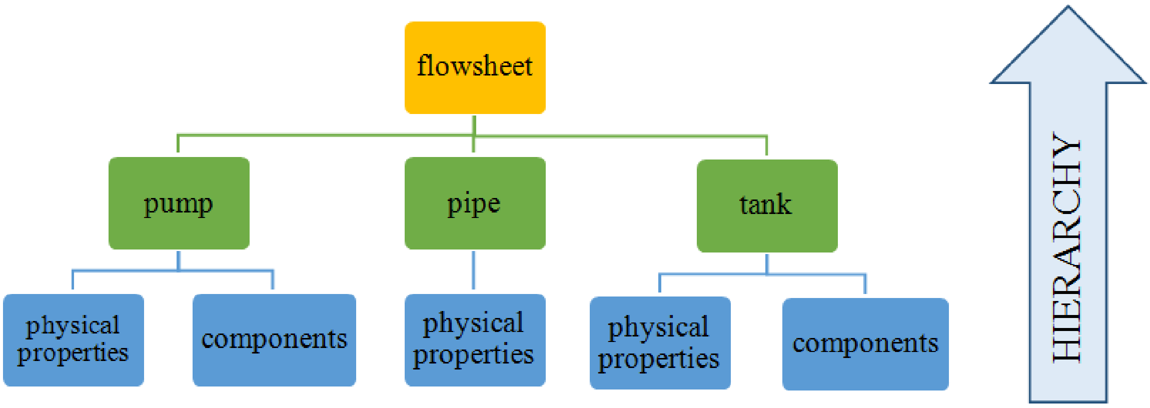


Figure 15: hierarchical structure of submodels

The models developed in gPROMS can be stored in the library and can therefore be used again. Many of the components and unit operations of complex systems are common to a wide range of processes and reusing the models avoids repeated modelling of identical components [Barton, 1992].

3.2.2 gPROMS standard steady state solvers

gPROMS uses two mathematical solvers for the solution of sets of nonlinear algebraic equations:

- BDNLSOL (**B**lock **D**ecomposition **N**on**L**inear **SOL**ver): The concept of the block decomposition is to rearrange the system of nonlinear equations. The columns and rows of the matrix are reordered in order to get a block triangular form.

- SPARSE: The SPARSE solver provides the implementation of the Newton-method. It is a solver for the solution of nonlinear algebraic systems without block decomposition [Process Systems Enterprise Limited, 2014b].

3.2.3 Supplying physical properties

Physical properties such as density, viscosity, enthalpy and phase equilibrium data can be implemented as foreign objects. gPROMS provides different packages designed for the use within the gPROMS ModelBuilder®. The Ideal Physical Properties Foreign Object (IPPFO) and Multiflash^{TM} are two standard packages:

IPPFO provides equilibrium and pure component properties whereas ideal mixing of the liquid and gas mixtures is assumed. It only supports mass based quantities.

Multiflash^{TM} is a physical properties package developed by InfoChem and supplied by gPROMS. It supports thermodynamic and transport properties with respect to temperature, pressure and mole compositions, specifically for equation-oriented modelling. Multiflash can calculate phase equilibria and fixed fraction of specified phases at fixed temperature or pressure. Multiflash^{TM} is a standalone software to generate files for the use in the gPROMS ModelBuilder® [KBC Advanced Technologies plc., 2014] and [Process Systems Enterprise Limited, 2014a].

Within the Multiflash GUI (graphical user interface) the user defines the components and activity coefficient model. This information is exported in an mfl-file, which is implemented in the gPROMS ModelBuilder environment. In order to choose other components, the user can directly change the mfl-file in the gPROMS “miscellaneous files”-folder. The modification of the mfl-file sometimes leads to problems in the first simulation. If that happens the user should safe the settings and restart the project.

Figure 16 illustrates the interdependency between the main and the physical properties data program.

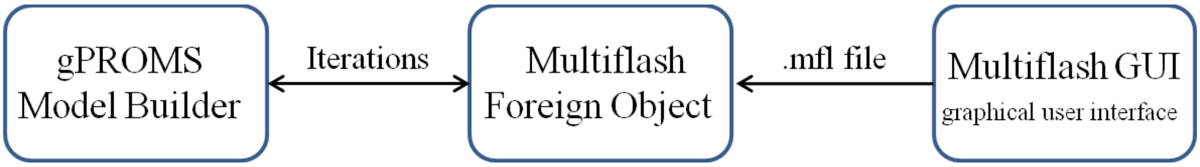


Figure 16: Overview of the interdependency between Multiflash and the gPROMS Model Builder

The (thermodynamic) properties provided for the following models are:

- dew- and bubble temperatures
- phase equilibrium data
- liquid activity, liquid and vapour fugacity coefficients
- vapour pressures
- enthalpies
- molecular weights.

Both packages IPPFO and Multiflash^{TM} offer a library with the most common components and properties. Nevertheless the number of properties of components is limited and it might be necessary to generate own files (e.g. in Excel with data from literature) and import them as foreign objects.

3.2.4 Process initialisation procedure

The initialisation procedure is a method to reach convergence of complex models and additionally guarantee higher model robustness. Since gPROMS is an equation-oriented simulation software, a user intervention to achieve convergence might be required. With initialisation procedures:

- temporary initial values can be assigned
- numerically “easier” versions of the systems are established

Practically the complex system of equations is divided into smaller blocks and those are simplified with defined initialisation procedures. By solving those step for step initial values for the precise models can be found. Afterwards the simplified models are sequentially reverted to the original problem.

4 Model implementation in gPROMS

gPROMS provides a library including common unit operations such as sources, sinks, heaters, pumps, columns, etc. These models often need special input specifications that require detailed knowledge of processes and even with the descriptions the models are not so easy to understand. In regard to the application in flowsheets those available unit operation models might cause problems. Besides the partly unusual input parameters, the structure of the library allows reverse flows. Due to this facts it is recommended to develop and implement own simple models and use the existing gPROMS models as support.

4.1 gPROMS models

The two mathematical models, WUG- and the column-model, were developed and implemented within the environment of the gPROMS ModelBuilder. In order to build up complete processes within flowsheets it is, besides those two main models, necessary to implement a series of other models:

- compounds
- source
- sink
- mixer
- splitter
- heater
- recycle breaker
- condenser
- reboiler
- **WUG (Winn-Underwood-Gilliland)**
- **column**.

Multiflash is used as physical properties database for all the models developed in gPROMS. They are designed for binary mixtures.

Figure 17 shows the hierarchical structure of the submodels that are integrated in the above listed models.

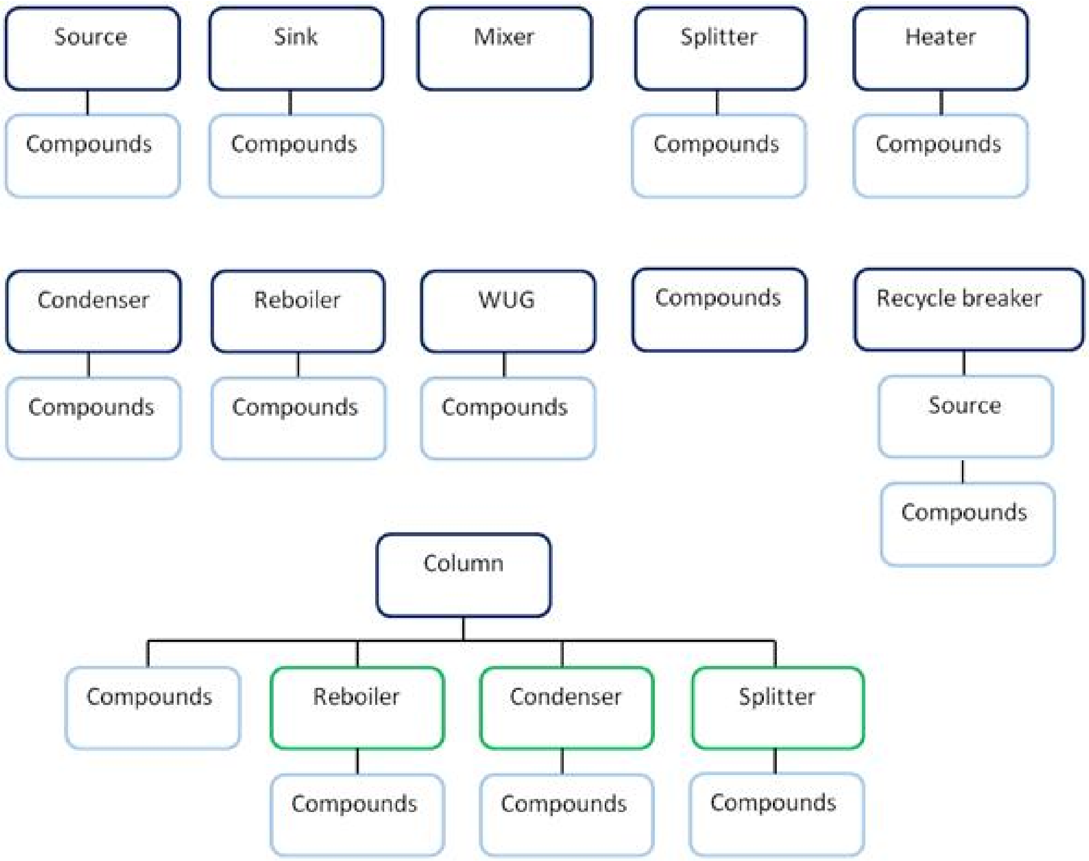


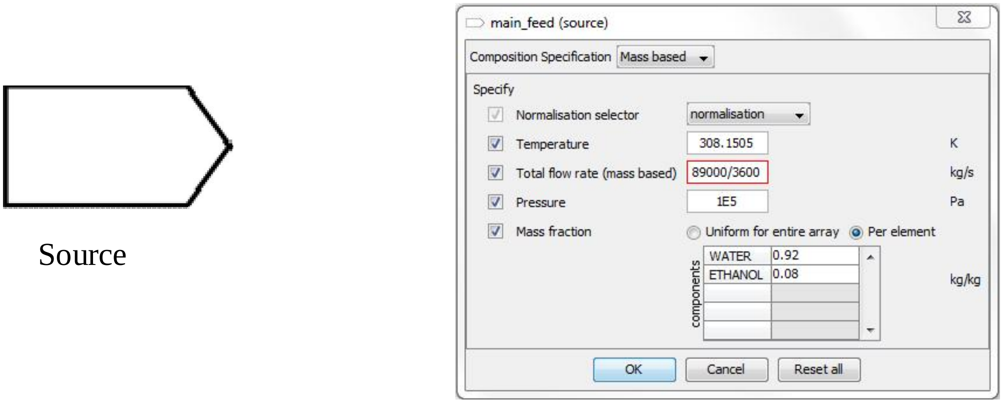
Figure 17: Hierarchical illustration of the models and their submodels

4.1.1 Compounds

As it can be seen in the illustration above, the model "compounds" is, apart from the mixer model, implemented in all the models. Its target is to retrieve the components defined in the Multiflash file.

4.1.2 Source

The source is a unit that defines an input stream to the flowsheet. Within this model the defined mass flow and, if chosen, the mass fractions are converted into mole based values. A selector allows the user to specify either mass based or mole based compositions. The conversion of the mass based values is necessary, because of the mole based unified communication between the models. Additionally the pressure and temperature need to be defined. Figure 18 shows the icon and specification window. There is a selector for the normalisation of the compositions. It applies, if the sum of the defined fractions is unequal to one.



main_feed (source)

Composition Specification: Mass based

Specify

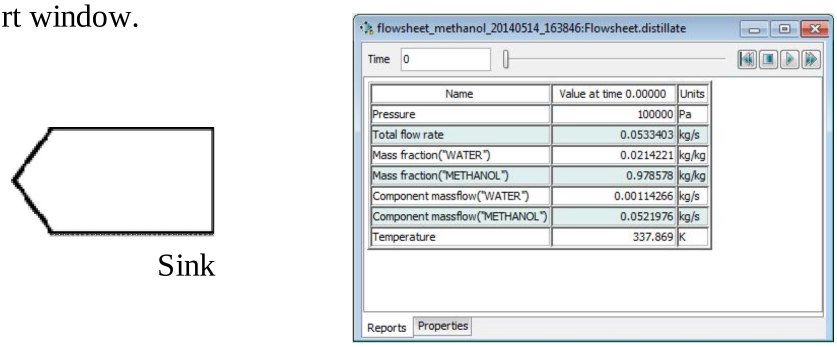
- ☑ Normalisation selector: normalisation
- ☑ Temperature: 308.1505 K
- ☑ Total flow rate (mass based): 89000/3600 kg/s
- ☑ Pressure: 1E5 Pa
- ☑ Mass fraction: ○ Uniform for entire array ● Per element

components		
WATER	0.92	kg/kg
ETHANOL	0.08	

Figure 18: Source icon and specification window

4.1.3 Sink

On the other hand, the output stream of the flowsheet is represented by the sink model. The incoming mole based stream values are converted to mass based values. After running the process the report is displayed by double clicking on the sink model. Figure 19 shows the icon and the report window.



flowsheet_methanol_20140514_163846:Flowsheet.distillate

Time 0

Name	Value at time 0.00000	Units
Pressure	100000	Pa
Total flow rate	0.0533403	kg/s
Mass fraction("WATER")	0.0214221	kg/kg
Mass fraction("METHANOL")	0.978578	kg/kg
Component massflow("WATER")	0.00114266	kg/s
Component massflow("METHANOL")	0.0521976	kg/s
Temperature	337.869	K

Figure 19: Sink icon and report window

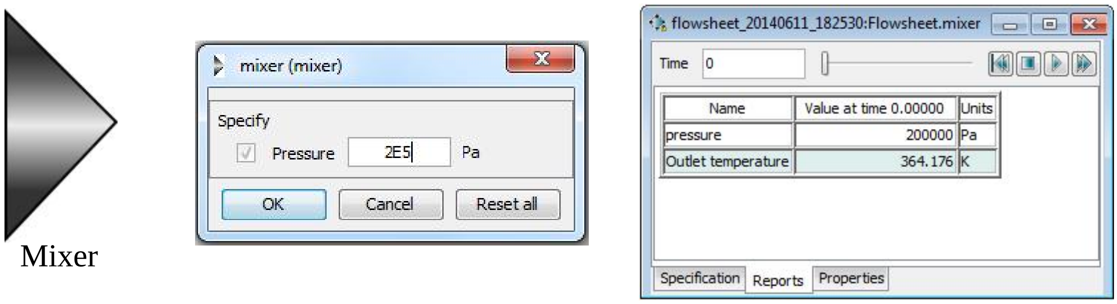
4.1.4 Mixer

The mixer includes the material and heat balance.

$$\sum_{m=1}^{ns} \dot{L}_{m,Mixer\_in} \cdot x_{i,m,Mixer\_in} = \dot{L}_{Mixer\_out} \cdot x_{i,Mixer\_out} \qquad 4.1$$

$$\sum_{m=1}^{ns} \dot{L}_{m,Mixer\_in} \cdot h_{i,m,Mixer\_in} = \dot{L}_{Mixer\_out} \cdot h_{m,Mixer\_out} \qquad 4.2$$

The outlet stream has to be equal the sum of the inlet streams m. The only variable that needs to be defined by the user is the outlet pressure. The following figure shows the model icon, specification and report window.



mixer (mixer)

Specify

☑ Pressure 2E5 Pa

flowsheet_20140611_182530:Flowsheet.mixer

Time 0

Name	Value at time 0.00000	Units
pressure	200000	Pa
Outlet temperature	364.176	K

Figure 20: Mixer icon, specification and report window

4.1.5 Splitter

The splitter is developed in order to be implemented as submodel in the column model. Its target there is to provide liquid overflow and thus counter current flow in the column. Therefore one part of the incoming product stream has to be returned to the column as reflux, whereas the other part is withdrawn as distillate product. The sum of the outgoing streams, reflux $\dot{R}$ and distillate $\dot{D}$, must therefore be equal the input stream:

$$\dot{L}_{Splitter\_in} = \dot{R} + \dot{D} \qquad 4.3$$

Additionally, the reflux ratio v is introduced. It is defined as the ratio between the reflux and distillate flow:

$$v = \frac{\dot{R}}{\dot{D}} \qquad 4.4$$

In order to calculate the outlet streams, either one stream or the reflux ratio need to be specified.

The compositions, temperature and pressure remain the same for both outgoing streams. They are equal the splitter's incoming stream.

4.1.6 Heater

The heater is used to raise or reduce the temperature of a stream by supplying or dissipating heat. The model is designed for liquid streams and doesn't consider phase transitions. Within the heater model, the material and heat balance are calculated.

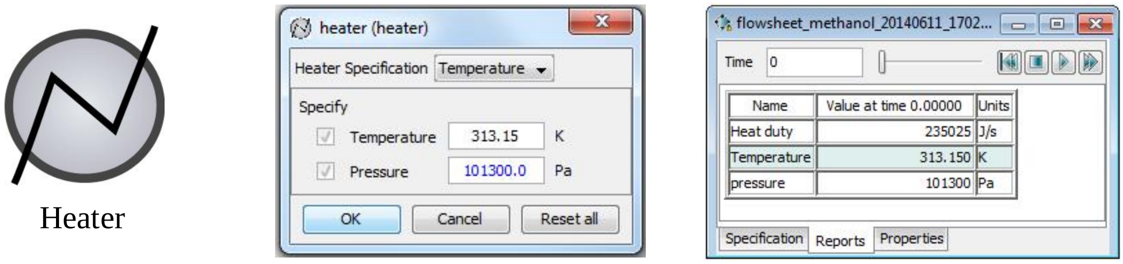
$$\dot{L}_{Heater\_in} \cdot x_{i,Heater\_in} = \dot{L}_{Heater\_out} \cdot x_{i,Heater\_out} \qquad 4.5$$

$$\dot{L}_{Heater\_in} \cdot h_{Heater\_in} + \dot{Q}_{Heater} = \dot{L}_{Heater\_out} \cdot h_{Heater\_out} \qquad 4.6$$

Thereby the flow rate of the outlet stream has to be equivalent to the inlet.

$$\dot{L}_{Heater\_in} = \dot{L}_{Heater\_out} \qquad 4.7$$

Besides the pressure, either the temperature or heat duty can be defined. The icon, specification and report window are illustrated in the figure below.



Heater Specification: Temperature

Specify

☑ Temperature 313.15 K

☑ Pressure 101300.0 Pa

Time 0

Name	Value at time 0.00000	Units
Heat duty	235025	J/s
Temperature	313.150	K
pressure	101300	Pa

Figure 21: Heater icon, specification and report window

4.1.7 Recycle breaker

The recycle breaker is an important element for the numerical stability. The definition of starting values helps the system to find results. First the calculation is carried out with user defined values, which are then reverted to the ones obtained during the calculation. The specifications of the recycle breaker are the same as in the source model, which is implemented as submodel.

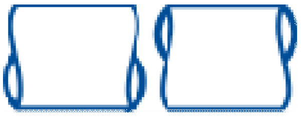


Figure 22: Recycle breaker icon

4.1.8 Condenser

The condenser is designed as total condenser. By dissipating heat, the incoming vapour stream is cooled down until it is totally liquefied. If the dew temperature is known, the model could be calculated using the mole and enthalpy balance and the equality equations for the incoming and leaving compositions. Since the case of using the dew temperature is numerically unstable, the equilibrium temperature is determined with the additionally defined equilibrium correlations and afterwards assumed to be equal to the dew temperature. This means that the model is designed with a (not existing) vapour output stream, which is then set zero. The deviations of the equilibrium state are considered with the Murphree efficiency.

The mole balance is:

$$\dot{G}_{C\_in} \cdot y_{i,C\_in} = \dot{G}_{C\_out} \cdot y_{i,C\_out} + \dot{L}_{C\_out} \cdot x_{i,C\_out} \qquad 4.8$$

Additionally, the compositions of the incoming vapour stream and the liquid ones leaving the condenser are the same. This is described with one summation correlation and the following equality equation:

$$y_{i,C\_in}(water) = x_{i,C\_out}(water) \qquad 4.9$$

Finally the dissipated heat duty $\dot{Q}_C$ to condense the vapour stream is determined with the heat balance:

$$\dot{G}_{C\_in} \cdot h^v_{C\_in} + \dot{Q}_C = \dot{L}_{C\_out} \cdot h^l_{C\_out} \qquad 4.10$$

4.1.9 Reboiler

Within the reboiler model the incoming liquid stream is partly evaporated. The reboiler's material balance is therefore:

$$\dot{L}_{R\_in} \cdot x_{i,R\_in} - \dot{G}_{R\_out} \cdot y_{i,R\_out} - \dot{L}_{R\_out} \cdot x_{i,R\_out} = 0 \qquad 4.11$$

The determination of the compositions is furthermore achieved by the use of the equilibrium correlations and summation conditions for the liquid and vapour phase. To reach the partly evaporation, external energy $\dot{Q}_R$ has to be supplied. Equation 4.12 shows the heat balance:

$$\dot{L}_{R\_in} \cdot h^l_{R\_in} + \dot{Q}_R = \dot{L}_{R\_out} \cdot h^l_{R\_out} + \dot{G}_{R\_out} \cdot h^v_{R\_out} \qquad 4.12$$

Furthermore the Murphree efficiency which is, as there is no vapour input, referred to the liquid phase, applies:

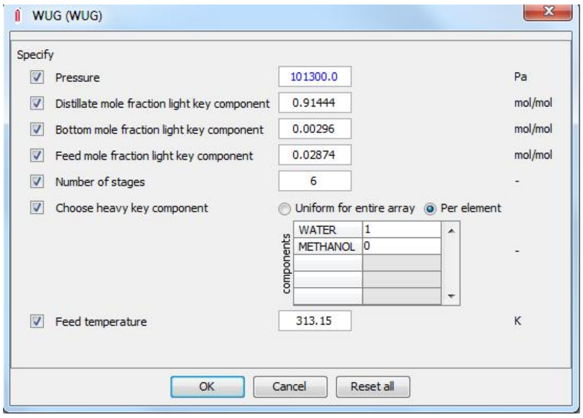
$$E = \frac{x_{i,n} - x_{i,n-1}}{x_{i,n} - x^*_{i,n-1}} \qquad 4.13$$

4.1.10 WUG (Winn-Underwood-Gilliland)-model

The WUG- model was developed for estimating operating parameters for a given separation problem. It is based on the shortcut correlations described in chapter 2.3.2. Within the model, the Winn correlation is used for the determination of the minimum number of theoretical stages and for the feed stage location. The Underwood equations provide a method to find the minimum reflux ratio and finally the Molokanov approach of the Gilliland curve defines the actual reflux ratio. In order to enable the comparison with Aspen Plus these short-cut correlations are chosen.

For the simulation, the feed, distillate and bottom molar compositions of the light key component need to be defined. In addition it is necessary to specify the pressure, the feed temperature, an estimated value for the number of theoretical stages and to declare the heavy key component.

The WUG-column icon (in flowsheet applications) and the appropriate specification window (appears with double click on the icon) are shown in the figure below.


WUG (WUG)

Specify

☑	Pressure	101300.0	Pa
☑	Distillate mole fraction light key component	0.91444	mol/mol
☑	Bottom mole fraction light key component	0.00296	mol/mol
☑	Feed mole fraction light key component	0.02874	mol/mol
☑	Number of stages	6	-
☑	Choose heavy key component	○ Uniform for entire array ● Per element	
		components: WATER 1 METHANOL 0	-
☑	Feed temperature	313.15	K

Figure 23: WUG-column icon and specification window

The process parameters obtained with the WUG-model serve as starting values for the column model and need to be further adapted to reach the desired specifications.

The short-cut methods are usually not valid for azeotropic mixtures. At and above the azeotropic point, the model will not find a positive number of stages-value anymore. However, cautiously it might be applied for compositions below the azeotropic point.

4.1.11 Column-model (n-stages model)

The steady state column model for the separation of binary mixtures is based on mass and energy balances. It is designed as an n-equilibrium stage model including a total condenser, a splitter and a reboiler as submodels.

For developing the column model the following model assumptions are made:

- no accumulation of material or energy on the stages (steady-state)
- no reaction terms
- consideration of equilibrium stages:
    - perfect mixing of the vapour and liquid phase
    - thermal equilibrium for every component on each stage
- no heat loss

The distillation column is set up of single separation stages. Apart from the equilibrium correlations for real mixtures (equation 2.3) described in chapter 2.2.2, two more equations are needed to entirely define one stage: the mass and heat balance.

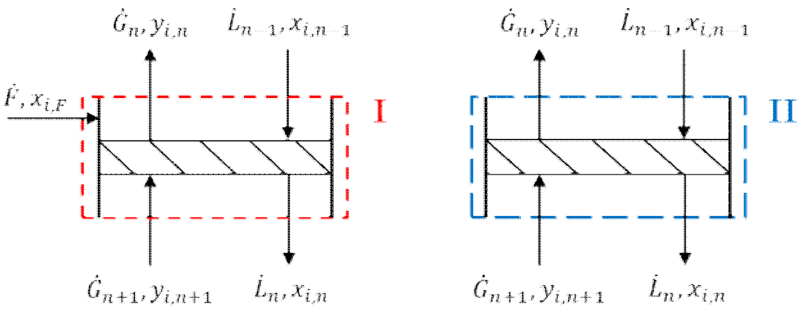
Mass balance

Figure 24 shows the boundary systems of the feed stage (*n*=*f*) and of any random separating stage *n* in the column. The overall flow is given by the two balances:

$$\text{(I)} \qquad \dot{F} + \dot{G}_{n+1} + \dot{L}_{n-1} = \dot{G}_n + \dot{L}_n \qquad 4.14$$

$$\text{(II)} \qquad \dot{G}_{n+1} + \dot{L}_{n-1} = \dot{G}_n + \dot{L}_n \qquad 4.15$$

The total input has to be equal to the output. $\dot{G}$ are the vapour and $\dot{L}$ the liquid flows. The only difference of the two balances is the input feed stream $\dot{F}$.



$\dot{G}_n, y_{i,n}$
$\dot{L}_{n-1}, x_{i,n-1}$
$\dot{F}, x_{i,F}$
I
$\dot{G}_{n+1}, y_{i,n+1}$
$\dot{L}_n, x_{i,n}$
$\dot{G}_n, y_{i,n}$
$\dot{L}_{n-1}, x_{i,n-1}$
II
$\dot{G}_{n+1}, y_{i,n+1}$
$\dot{L}_n, x_{i,n}$

Figure 24: Balance I: feed stage, balance II: any random stage n

For the component *i* and stage *n/f* applies:

$$\text{(I)} \qquad \dot{F} \cdot x_{i,F} + \dot{G}_{n+1} \cdot y_{i,n+1} + \dot{L}_{n-1} \cdot x_{i,n-1} = \dot{G}_n \cdot y_{i,n} + \dot{L}_n \cdot x_{i,n} \qquad 4.16$$

$$\text{(II)} \qquad \dot{G}_{n+1} \cdot y_{i,n+1} + \dot{L}_{n-1} \cdot x_{i,n-1} = \dot{G}_n \cdot y_{i,n} + \dot{L}_n \cdot x_{i,n} \qquad 4.17$$

The sum of the incoming liquid and vapour streams multiplied with their liquid and vapour composition *x* and *y* is equal to the sum of the outgoing streams multiplied with their compositions.

For the components leaving one equilibrium stage the summation conditions (equations 3.3 and 3.4) for the liquid and vapour compositions must apply.

Heat balance

The heat/ enthalpy balances for the systems are described as follows:

$$\text{(I)} \qquad \dot{F} \cdot h_F + \dot{G}_{n+1} \cdot h^v_{n+1} + \dot{L}_{n-1} \cdot h^l_{n-1} = \dot{G}_n \cdot h^v_n + \dot{L}_n \cdot h^l_n \qquad 4.18$$

$$\text{(II)} \qquad \dot{G}_{n+1} \cdot h^v_{n+1} + \dot{L}_{n-1} \cdot h^l_{n-1} = \dot{G}_n \cdot h^v_n + \dot{L}_n \cdot h^l_n \qquad 4.19$$

The sum of the incoming liquid and vapour flows multiplied with their liquid and vapour enthalpies h^l and h^v is equal to the sum of the outgoing streams multiplied with their enthalpies. External heat supply and heat loss are neglected in both equations (model assumptions).

Pressure drop

The introduction of a stage-pressure drop Δp_n is besides the Murphree efficiency a second development towards a more realistic model of a distillation column. It is assumed, that the pressure drop of every stage is identical. Therefore the total column pressure drop Δp is divided through the number of stages + 1 (including the reboiler and condenser):

$$\Delta p = p_{Reboiler} - p_{Condenser} \qquad 4.20$$

$$\Delta p_n = \frac{\Delta p}{NT + 1} \qquad 4.21$$

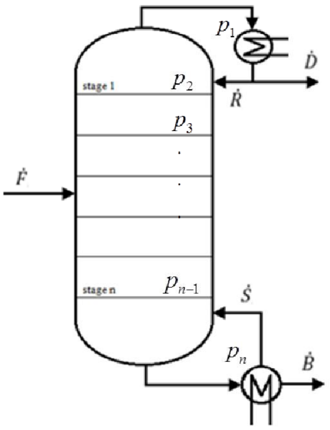


Figure 25: Illustration pressure distribution

Those basic mathematical correlations entirely describe the single stages and additionally deviations from the equilibrium state. As a further step linking conditions to specify the input of the stages are required to build up the n-stages column model. The stages are thereby counted from top to bottom and the variables transferred are the streams and fractions, the temperature and pressure.

Stage connectivity

For the component *i* and the stages *n=1* to *NT-1* applies:

$$liq\_in\_variables(i, n + 1) = liq\_out\_variables(i, n) \qquad 4.22$$

and

$$vap\_in\_variables(i, n) = vap\_out\_variables(i, n + 1) \qquad 4.23$$

The predefined value for the number of stages *n* determines thereby the number of linking conditions. In addition to the connections within the column, the feed input has to be considered.

Feed connectivity

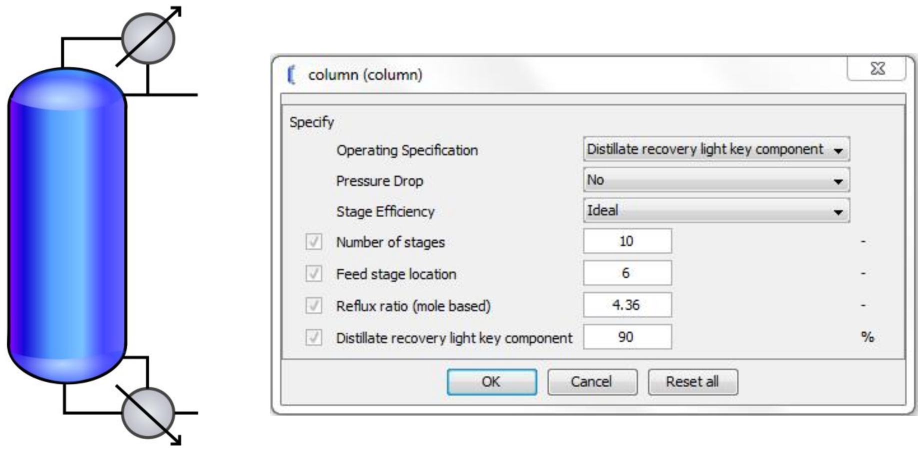
The column model is designed for one feed stream. Since the equations 4.16 and 4.18 describe the mass and heat balance for every stage n in the column, all of them would have a feed input. By defining the feed stage location k, the following correlations help to reduce the mass and heat balances to the equations 4.17 and 4.19 for the stages without feed inlet. Therefore the feed flow is multiplied with the obtained value x:

$$for\ n > k: \quad x = 0 \tag{4.24}$$

$$for\ n = k: \quad x = 1 \tag{4.25}$$

$$for\ n < k: \quad x = 0 \tag{4.26}$$

Figure 26 shows the column icon and the specification window that appears by double clicking on the icon within the flowsheet application.



Operating Specification: Distillate recovery light key component

Pressure Drop: No

Stage Efficiency: Ideal

☑ Number of stages: 10 -

☑ Feed stage location: 6 -

☑ Reflux ratio (mole based): 4.36 -

☑ Distillate recovery light key component: 90 %

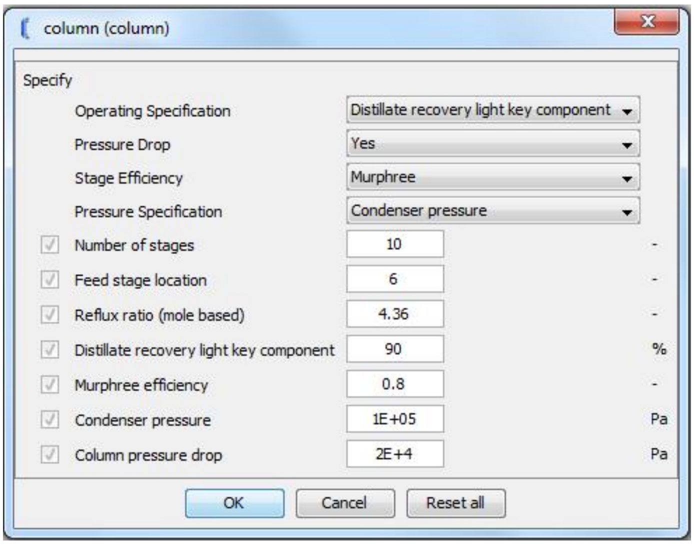
Figure 26: Column icon and specification window

In addition to the number of stages, the feed stage location and the reflux ratio, one more variable must be specified to carry out the simulation. The user can choose between:

- distillate recovery light key component
- reboiler heat duty and
- distillate rate (mass flow).

Furthermore, the Murphree efficiency and column pressure drop can be activated and defined. The introduction of a pressure drop additionally requires the specification of either the condenser or reboiler pressure. It should be noted, that the number of stages defined in the column model, do not include the condenser and reboiler.

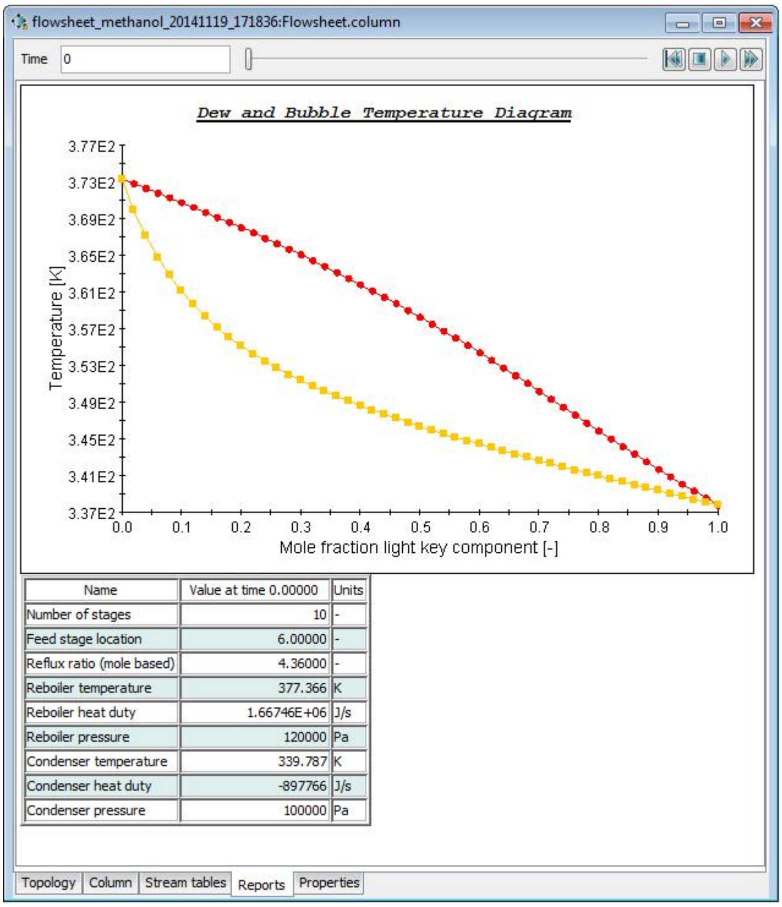
The specification window with all possible adjustable process parameters looks as follows:



	Operating Specification	Distillate recovery light key component	
	Pressure Drop	Yes	
	Stage Efficiency	Murphree	
	Pressure Specification	Condenser pressure	
☑	Number of stages	10	-
☑	Feed stage location	6	-
☑	Reflux ratio (mole based)	4.36	-
☑	Distillate recovery light key component	90	%
☑	Murphree efficiency	0.8	-
☑	Condenser pressure	1E+05	Pa
☑	Column pressure drop	2E+4	Pa

Figure 27: Column specification window with all process parameters

After the process model execution, the column report can be displayed by double clicking on the column icon. Besides the bubble and dew temperature diagram, it shows the column parameters and the most important reboiler and condenser data. Figure 28 illustrates the column report window of a separation problem.



Name	Value at time 0.00000	Units
Number of stages	10	-
Feed stage location	6.00000	-
Reflux ratio (mole based)	4.36000	-
Reboiler temperature	377.366	K
Reboiler heat duty	1.66746E+06	J/s
Reboiler pressure	120000	Pa
Condenser temperature	339.787	K
Condenser heat duty	-897766	J/s
Condenser pressure	100000	Pa

Figure 28: Column model report window

5 Process initialisation procedure

Considering the system equations, the mass/energy balances and equilibrium correlations are non-linear coupled correlations. The solver may fail to converge or converge to an infeasible or physically unrealistic solution. To overcome these difficulties the iteration-variables have to be preset initially. To enable convergence, the assumptions should be as close to the true values as possible.

Additionally, the conditions in the top, bottom and every single stage in the column are related and depend on each other. The difficulty is to implement a stage by stage calculation in the equation oriented approach, because the whole system of equations is solved simultaneously.

Therefore, in the process initialisation procedure the complex system is divided into smaller blocks. For those initial correlations or values are assumed. The simulation starts with the initial state to obtain initial values for the precise model, successively activates single blocks in an user defined order until reaching the original state.

5.1 Initial state

This section describes the developed simplified systems in order to determine initial values for the final state calculations.

5.1.1 Enthalpy initial state

Column

To find initial values for the enthalpies, the temperature profile of the column has to be estimated. The first approach was a linear profile between both boiling temperatures, which was not robust at all. The study of simulation results with different input specifications has shown that even small changes, for example another feed stage location, result in a completely different temperature profile. But it turned out, that, in contrast to the upper part of the column, the lower area is more sensitive regarding the temperature steps. For this purpose, the boiling temperature of methanol was not chosen as equilibrium temperature of stage one, because then the slope and thus the temperature steps between the stages in the stripping part would get too high.

The introduction of the equations 5.1-5.3 serves to approximate the equilibrium temperatures throughout the stages n in the column. Figure 29 shows the calculated values across the number of stages. This approach leads to good initial temperature values for the enthalpy calculations.

$$T_{eq}(1) = 350 \tag{5.1}$$

$$T_{eq}(NT) = 372 \tag{5.2}$$

$$T_{eq}(n) = T_{eq}(1) + \frac{T_{eq}(NT) - T_{eq}(1)}{(NT - 1)} \cdot (n - 1) \tag{5.3}$$

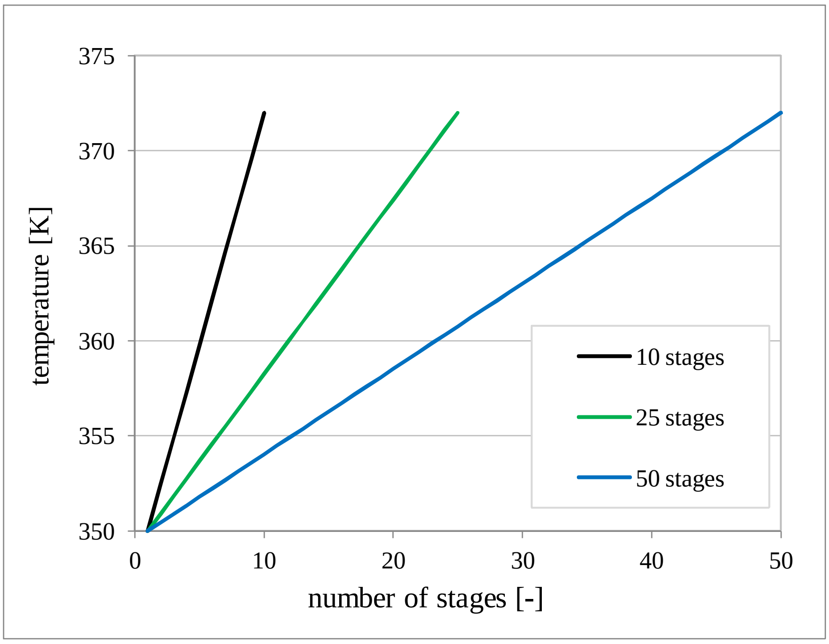


Figure 29: Equilibrium temperature initialisation

The problem of this temperature initialisation is that the starting and end value of the linear profile are fixed. If the components are changed, an adaptation of this profile will be required.

Reboiler

In the reboiler, depending on which input variable is defined, different initial approaches are activated. Therefore a case indicator (CA) is introduced:

- CA=1: distillate rate (distillate mass flow)
- CA=2: reboiler heat duty
- CA=3: distillate recovery of the light key component

By defining one of the three variables, the case indicator automatically takes the right value. With if cases within the model equations, the right initial correlation is recalled.

For CA=1 and CA=3 the variable enthalpy check h_{check} is introduced and set zero:

$$h_{check} = 0. \tag{5.4}$$

Afterwards, the sums of the incoming and leaving heat flows are calculated. They have to be equal to the value of the enthalpy check:

$$\sum \dot{Q}_{in} - \sum \dot{Q}_{out} = h_{check} \tag{5.5}$$

For the defined reboiler heat duty (CA=2) an approach for the calculation of the equilibrium temperature has proved to be effective. Therefore, the first approach calculates the equilibrium temperature as the temperature of the incoming stream plus 1 Kelvin:

$$T_{eq} = T_{Reboiler\_in} + 1 \tag{5.6}$$

Additionally, the heat streams are calculated and equation 5.5, which is used as well for the other case indicators, applies.

Condenser

The enthalpy initial state of the condenser equals the one of the reboiler with the case indicators 1 and 3.

The correlations selected for the reboiler and condenser have the advantage of being applicable for various binary feed mixtures.

5.1.2 Equilibrium initial state

Column

The very first attempt of an equilibrium initialisation was the estimation of the vapour and liquid composition profiles throughout the column. Several separation problems were simulated with Aspen Plus. The evaluation of the results has shown that the feed stage location has a significant influence on the profile. The try to divide the column into two sections, one above and one below the feed stream, and find a universally valid and suitable profile that can be applied for different feed stage locations worked out for a fixed number of stages. In regard to stay flexible up to 50 stages, every profile approach that was tested failed at some point. It turned out that a well chosen enthalpy initialisation provides initial equilibrium temperatures that are sufficient for a rather generally formulated equilibrium initialisation.

Finally, the equilibrium equations for high pressures and real mixtures, described in chapter 2.2.2, are used as an approach of the equilibrium compositions in the column stages.

$$\varphi_i^V \cdot y_i = x_i \cdot \varphi_i^L. \qquad 2.4$$

Besides the column stages, this correlation is further used in the condenser.

Reboiler

The trial of different pressure specifications has shown that the solver became unstable by defining special pressure values (for instance 100000 Pascal). Only small changes of the values led in turn to convergence and stability. This behaviour might occur because of discontinuities in the iteration-functions. Therefore it is probably a problem of Multiflash. Since Multiflash is a black-box application the background of this behaviour can not exactly be classified.

To avoid these instabilities, equation 2.4 is adapted as follows:

$$\varphi_i^V \cdot y_i = x_i \cdot (\varphi_i^L - 0.002). \qquad 5.7$$

This small adjustment of the liquid fugacitiy coefficient proved to be effective. The initialisation correlation 5.7 is valid for the case indicators 1 and 2.

For a defined distillate recovery of the light key component (CA=3) it is provisionally assumed that the water vapour composition leaving the reboiler is 0.95.

$$y_{R\_out,water} = 0.95 \qquad 5.8$$

Furthermore, the equilibrium temperature is set as bubble temperature of the incoming liquid composition.

$$T_{R\_eq} = T_{bubble}(x_{R\_in}) \qquad 5.9$$

The exact assignment to the component “water” in equation 5.8 has the consequence that a change of the components will lead to initialisation problems, if water is not one of the components and furthermore not the heavy key component. This problem could be eliminated through an algorithm that automatically recognizes the heavy and light key component. Attempts regarding this have already been made, but were not longer pursued due to the time and work effort.

5.1.3 Reflux ratio and Murphree efficiency initial state

Since the solver has problems to converge by the introduction of the Murphree efficiency, the initialisation procedure is designed on the base of equilibrium behaviour throughout the column. Therefore, the Murphree efficiency is set to one at the beginning and is then reverted to the specified value.

The initialisation procedure for the reflux ratio is necessary if the number of stages increases. Since a higher number of stages require a lower reflux ratio, it is set 0.01 at the beginning. It is adjusted slowly before it is completely reverted to the user specified value.

5.2 Final State

After the process model calculations with the initial correlations, the final state equations are activated step by step. The final state equations are the ones that are defined in the models (described in chapter 4). Additionally the initial values for the Murphree efficiency and reflux ratio are reverted to the ones defined by the user. The following sequence of initialisation proved to be effective [Christa et al., 2014]:

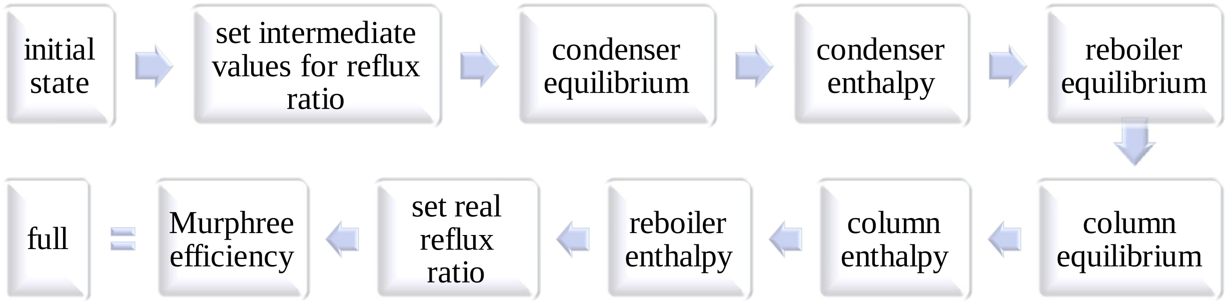


Figure 30: Sequence of initialisation steps

5.3 Preset

Besides the initialisation procedure it is possible to preset variables. The preset command is a local redefinition of bounds and starting values. Therefore a value for the variable is assumed and additionally lower and upper boundaries set. The PRESET command in the gPROMS language looks like follows:

```
PRESET
variable := preset value: lower boundary : upper boundary
```

Table 6 shows the model’s preset values and boundaries.

Table 6: Model's preset values and boundaries

(Sub-) Model	Variable	Preset value	Lower boundary	Upper boundary	Unit
n-stages Column Model	$y_{i,n}$	0.5	0	2	[mol/mol]
	$x_{i,n}$	0.5	0	2	[mol/mol]
	$T_{n,eq}$	370	-200	500	[K]
Condenser	$x_{(i-water),C\_out}$	0.98	0	1	[mol/mol]
	y_{i,C_out}	0.5	0	1	[mol/mol]
	$\dot{L}_{C\_out}$	100	-20000	50000	[mol/s]
Splitter	$\dot{d}$	0.1	-1000	1000	[kg/s]
Reboiler	T_{eq}	370	200	500	[K]

5.4 Adjustments for the application of ethanol/water

The application of the ethanol/water mixture appeared to be numerically stable. The available initial state equations in the column stages are directly set equal the ones from the final state. Therefore an if-case is implemented within the column model equations. By defining the components, the system automatically activates the right initialisation procedure.

The initialisation procedure is foresighted designed as system-independent (in regard to flow, composition and component system) as possible. Considering new component systems this could unfortunately not be implemented completely. The enthalpy initialisation for the column stages contains fixed starting and end values that might cause problems by the application of other binary systems. Additionally, the equilibrium initialisation procedure for a defined recovery of the light key component in the reboiler should be adapted in regard to other component mixtures. This means, that new component systems have to be tested and might require adjustments of the initialisation procedures.

6 Model validations

6.1 Comparison and discussion of results

To validate the models, simulation results of two different binary systems are cross-referenced with Aspen Plus.

Aspen Plus is a process simulation tool developed by Aspen Technology (Advanced System for Process Engineering). Aspen simulation packages include large experimental databanks for thermodynamic and physical parameters. With the model library provided by Aspen Plus it is possible to develop process flow diagrams for complete plants by interconnecting several units. Aspen Plus is mainly used for steady state simulations, design and optimisation planning. “The sophisticated Aspen software tool can simulate large processes with a high degree of accuracy” (Jana, 2009). This statement from Jana shows that Aspen Plus is an adequate reference for the comparison of the simulation results [Aspen Technology, 2014] and [Jana, 2009].

For the following simulations, NRTL is chosen as activity coefficient model.

6.1.1 Physical properties

To find out, if the physical properties provided by Multiflash are in accordance with literature data, different quantities such as the enthalpy, dew and bubble temperatures are compared. The following graphs show the dew and bubble temperatures of the water/methanol mixture at a fixed pressure. In addition to the experimental data from literature (data from [Perry, 1997], see also 2.2.4), the values provided by Aspen Plus are added as a second reference.

The data determined by Multiflash show very good accordance with both, the data from literature and Aspen Plus.

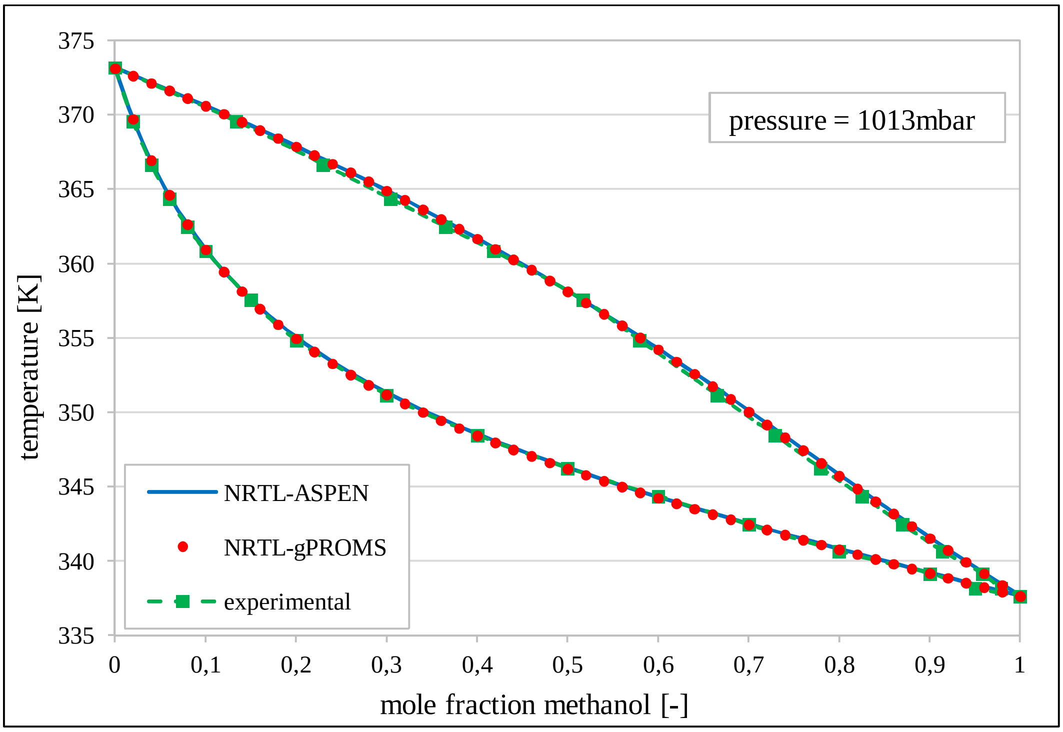


Figure 31: Bubble and dew temperature plot water/methanol mixture, experimental data from [Perry, 1997]

6.1.2 Short-cut method

Table 7 summarises the process parameters used in chapter 2.3.1 extended by the ones needed for the application of the short-cut method and the methanol/water separation problem.

Table 7: Process parameters of the water/methanol separation

Parameter	Unit	Value
Concentration methanol in feed	[wt%]	5
Concentration methanol in distillate	[wt%]	95
Recovery methanol	[%]	90
Feedflow	[kg/h]	10000
Temperature feed	[K]	293
Temperature feed after preheating	[K]	313
Pressure	[mbar]	1013
Pressure drop	[mbar]	0

The results of the short-cut method WUG are compared with the DSTWU model (Aspen Plus). DSTWU is a shortcut distillation design model in Aspen Plus that is used to specify column operating conditions. It is based on the Winn-Underwood-Gilliland correlations.

Within DSTWU und WUG the theoretical number of stages has to be defined. Therefore the value obtained by the McCabe Thiele method (see chapter 2.3.1) is used.

Table 8: Results of the short-cut methods

Short-cut method	$n_{th.min}$	n_{th}	feed stage	v_{min} (mole based)	v (mole based)
DSTWU	5.36	12	10.51	5.24	5.88
WUG	5.71	12	9.47	5.24	6.55

The comparison of the results in table 8 shows that the two models WUG and DSTWU provide almost the same values, which serve as good initial parameters for the column models.

In a second comparison the number of stages is varied within the WUG und DSTWU models. The values of the minimum reflux ratio and minimum number of stages remain the same. The graphs of the actual reflux ratio and the feed stage location are illustrated in the following figure.

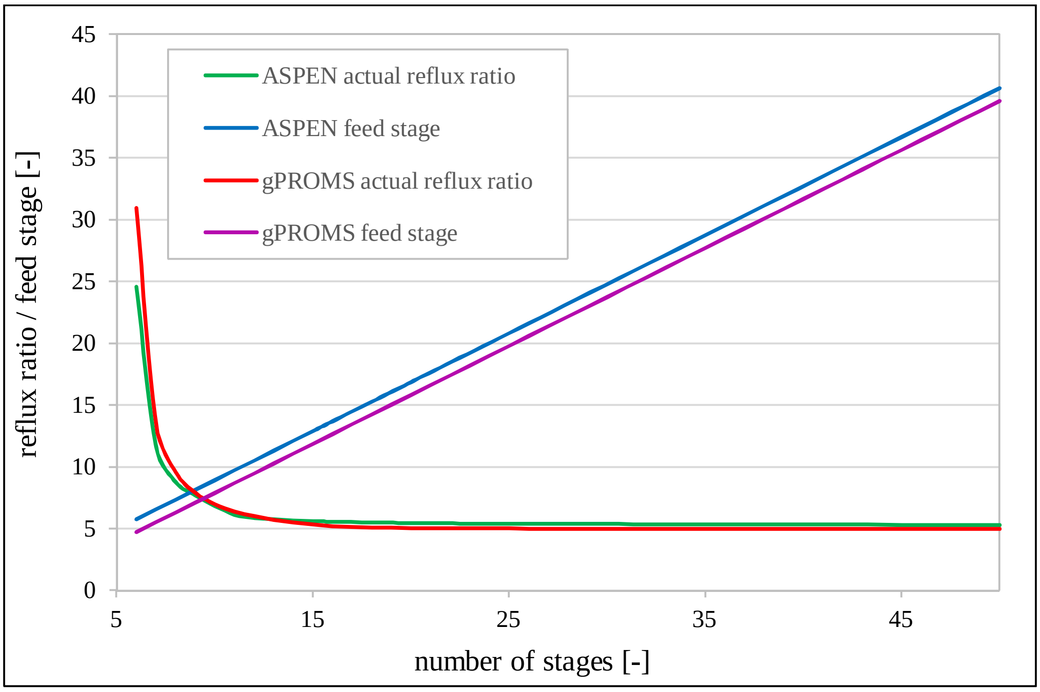


Figure 32: Results of the short-cut methods by variation the number of stages

Both short-cut methods, WUG and DSTWU show the same curve progressions for the actual reflux ratio and feed stage location. The gPROMS feed stage location is constantly one stage higher than the ones determined by Aspen Plus. The actual reflux ratio calculated with WUG is for lower number of stages higher than in Aspen Plus. Despite this fact the curves overlap by increasing number of stages. Since the value of the reflux ratio should not be too high in regard to economic efficiency, this deviation at low number of stages is not of big importance due to high reflux ratio.

6.1.3 CH_3OH/ H_2O mixture

The results obtained with the short-cut methods need to be further adapted to meet the desired specifications (see table 7) and if possible to minimize the required heat supply. Therefore the number of stages, the reflux ratio and the feed stage location are varied within the gPROMS column model and the model in Aspen Plus. Radfrac is chosen as a column in Aspen Plus with a Kettle reboiler and the condenser is set as total condenser. In both models the number of stages is counted from top to bottom. It has to be considered, that the reboiler and condenser are counted as stages within the Radfrac model but not in the gPROMS model. For the comparison of the results, the reboiler and condenser data of the gPROMS simulations are added as stages in the graphs.

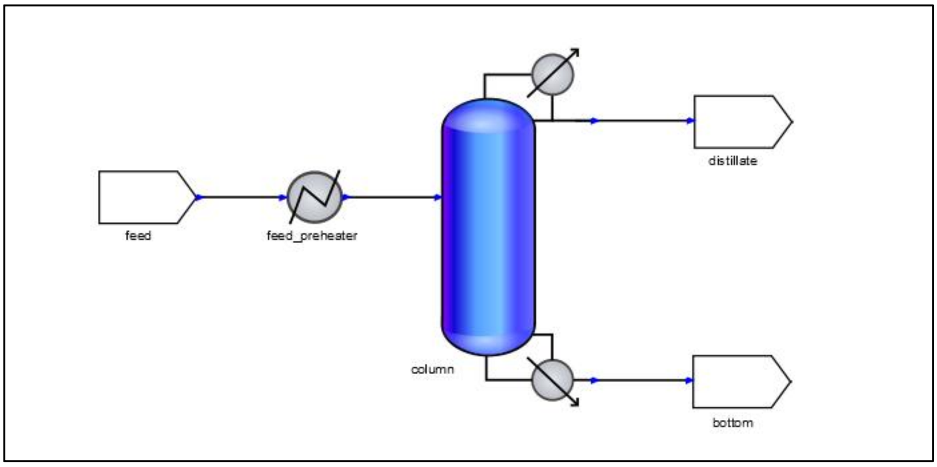


Figure 33: gPROMS flowsheet of the methanol/water separation problem

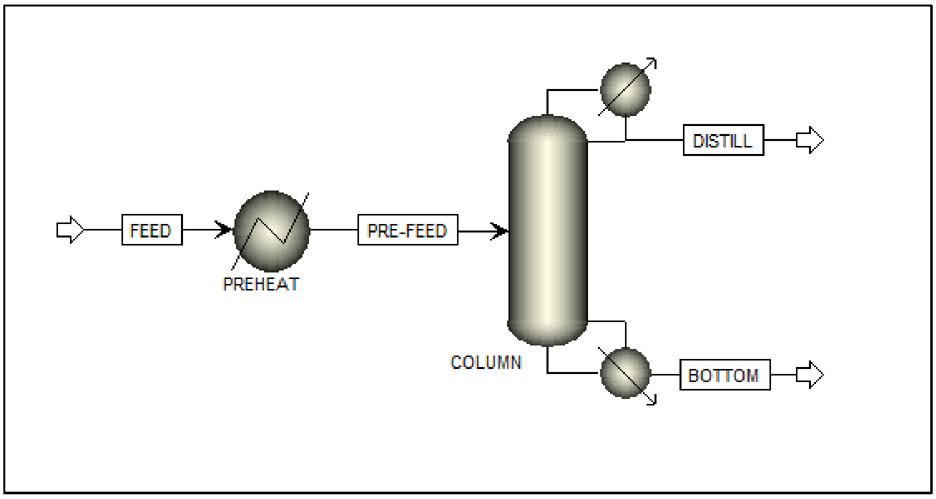


Figure 34: Aspen Plus flowsheet of the methanol/water separation problem

Figure 33 and 34 illustrate the gPROMS and Aspen Plus flowsheets for the separation of methanol and water. The feed is preheated with a heater, before it is fed to the column. The model Heater is used as a feed preheater in Aspen Plus.

The optimisation of the input parameters led to the following specifications for the flowsheet calculation:

Table 9: Column input parameters for the water/methanol separation

Column	Number of stages [-]	Feed stage location [-]	Reflux ratio (mole based) [-]	Distillate rate [kg/h]
Radfrac (Aspen Plus)	12	7	4.36	474
Column Model (gPROMS)	10	6	4.36	474

Table 10 summarises the simulation results of the top and bottom performances of both models and the necessary heat duty for preheating. It is visible, that the results are in good accordance. Only the applied heat duties required in the heater and reboiler are higher in the gPROMS model. In the condenser the obtained heat duty is higher in Aspen Plus. Therefore the gPROMS flowsheet needs more total heat supply. The causes for those deviations are small differences between the model’s enthalpies.

Table 10: Simulation results of the water/methanol separation

Simulation results	Aspen Plus		gPROMS	
	Bottom	**Top**	**Bottom**	**Top**
Temperature [K]	373	339	373	339
Heat duty [kW]	1503	-819	1522	-810
Methanol massflow [kg/h]	49	451	50	450
Water massflow [kg/h]	9477	23	9476	24
Methanol massfraction [-]	0.01	0.95	0.01	0.95
Water massfraction [-]	0.99	0.05	0.99	0.05
Heater heat duty [kW]	213		235	

Figure 35 shows the temperature and methanol fraction column profile. There are no deviations visible between the models.

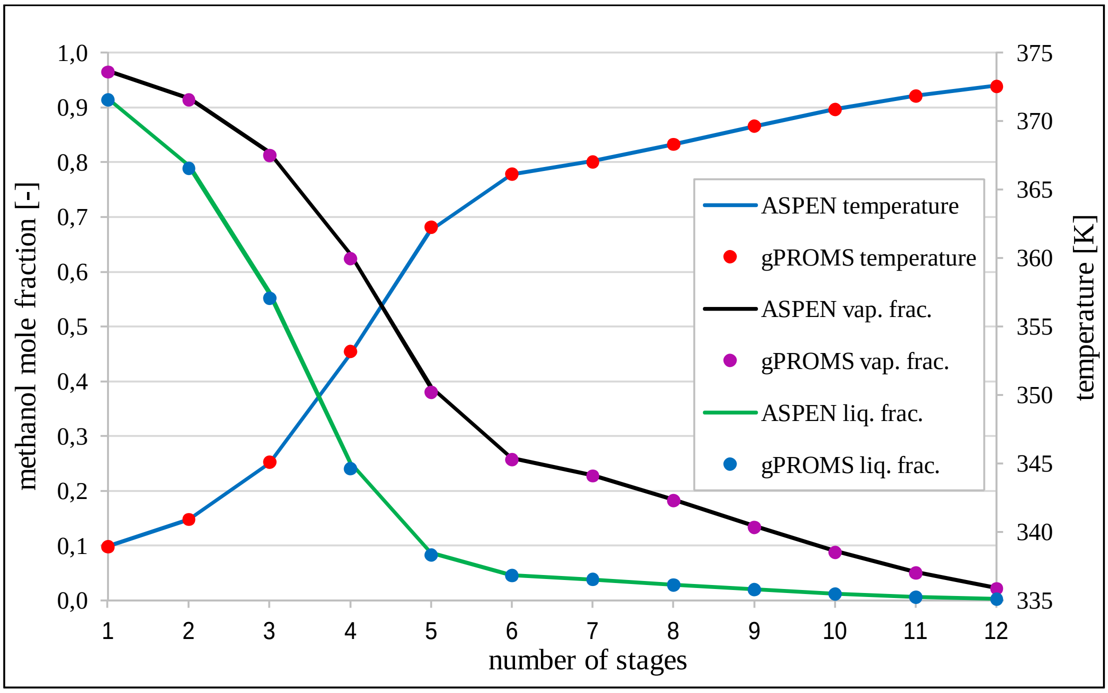


Figure 35: Temperature and methanol fraction column profile

Additional investigations are made with a sensitivity analysis. Therefore the distillate rate is kept at constant level and the reflux ratio is changed. The influence on product purity and necessary heat duty are illustrated in figure 36.

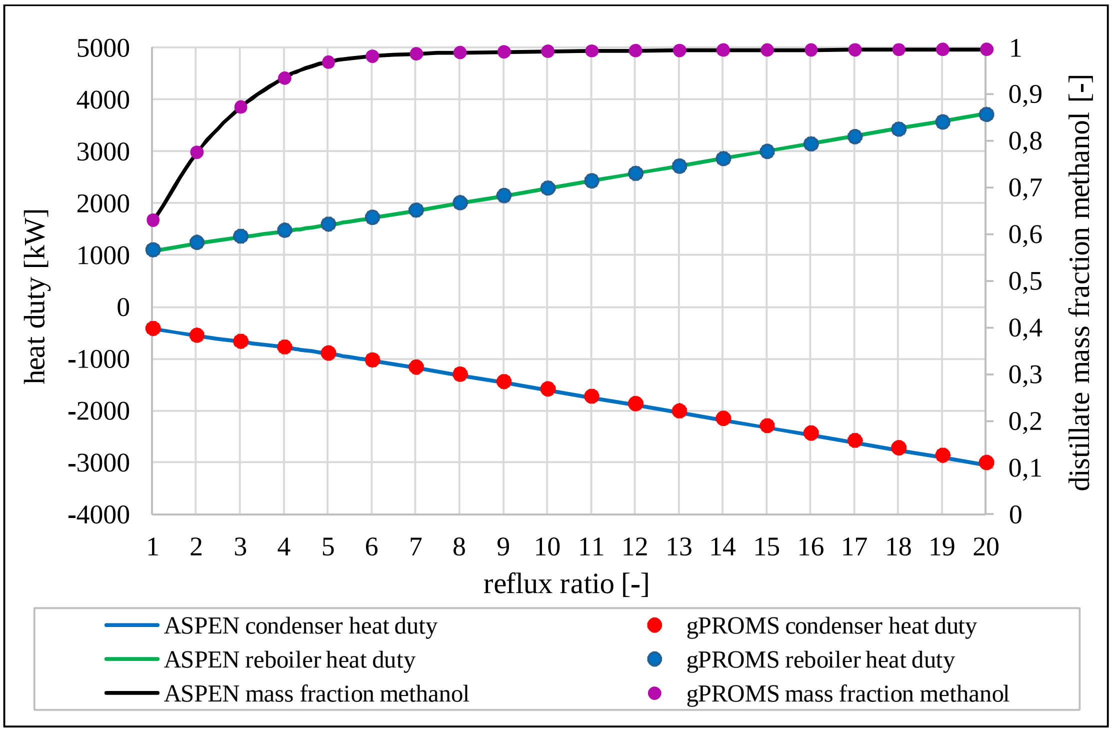


Figure 36: Results of sensitivity analysis: influence of the reflux ratio on product purity and necessary heat duty

It can be seen that the results and curve progressions of gPROMS and Aspen Plus match well. The sensitivity analysis shows that the increase of the reflux ratio leads to higher product purity and induces a higher required heat supply. For reflux ratio higher than 9, the product purity doesn't change anymore. Due to this fact and by considering the increasing heat supply it is, under these conditions, not necessary to use reflux ratios that are higher than 9.

6.1.4 C_2H_5OH/ H_2O mixture

For further validations of the gPROMS column model an ethanol/water separation problem was investigated. This separation process is part of the bioethanol production. In recent years bioethanol got more and more important as an alternative renewable energy source. Through the addition of enzymes the agricultural feedstocks are transformed to glucose. In a fermentation stage the glucose is further converted into alcohol by yeast. Afterwards the alcoholic mash is fed to distillative separation which is, in general, built up by 2 or 3 column systems. Due to energy savings the columns operate at different pressure levels. Within the columns the crude alcohol is purified to reach desired specifications, whereas a complete separation of the ethanol/water mixture is not possible due to the azeotropic point. The first column (=low pressure column) works at atmospheric pressure in order to achieve a maximum of water removal with as little energy input as possible. At the top of the second column (=high pressure column) the ethanol/water mixture can be removed near the azeotropic point. To get a completely anhydrous ethanol, the mixture has to be further dehydrated [Huber, 2005].

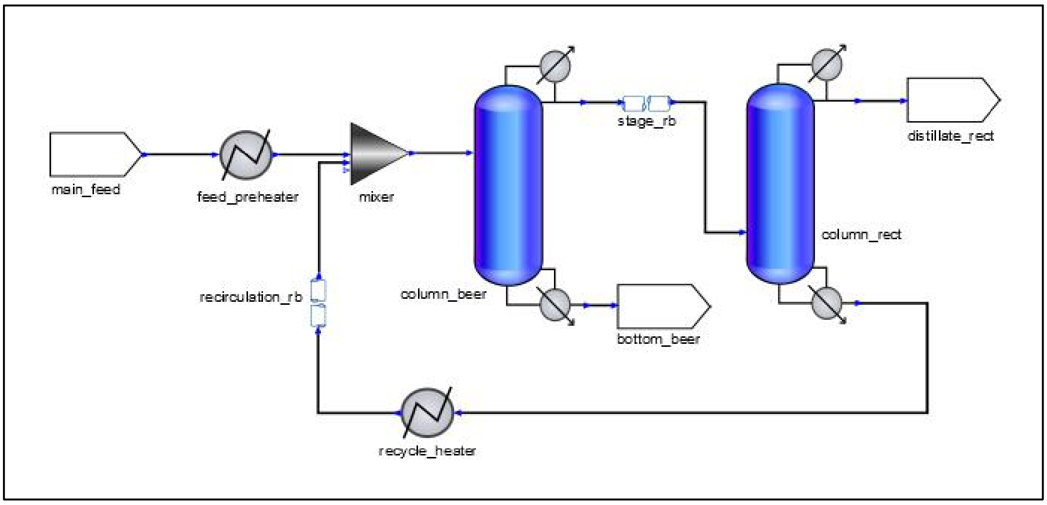


Figure 37: gPROMS flowsheet of the ethanol/ water separation problem

In the gPROMS and Aspen Plus flowsheets in figure 37 and 38 it is visible, that two columns, the beer and rectification column are arranged in series. The bottom's product stream of the rectification column is mixed with the preheated main feed. The mixed stream is then fed into the beer column. The top stream of the beer column serves as input for the following rectification column.

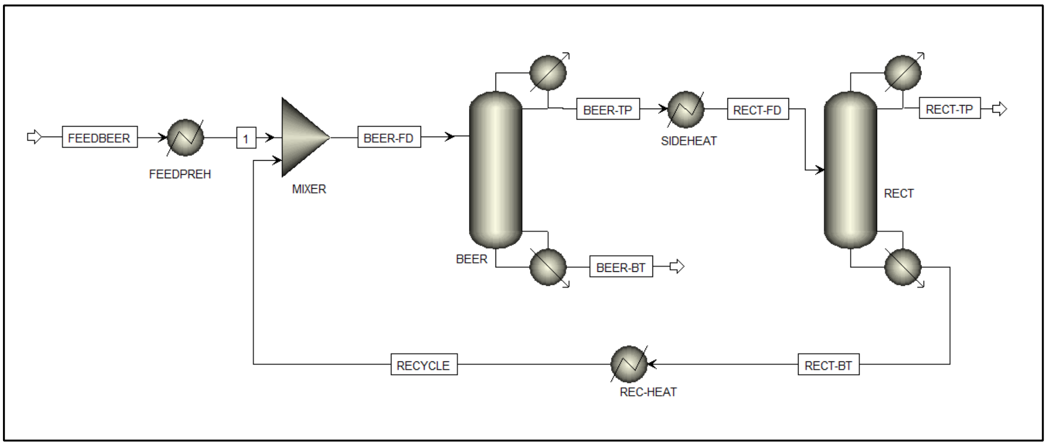


Figure 38: Aspen Plus flowsheet of the ethanol/ water separation problem

Table 11 shows the input specifications of the ethanol/ water separation problem. Due to the fact, that the number of stages in Aspen Plus includes the reboiler and condenser, the gPROMS input must be two stages less. Additionally the feed stage location is adapted (one lower in gPROMS).

Table 11: Input parameters for the water/ethanol separation

Specifications					
Feed	Feedflow [kg/h]	89000		**Beer column**	**Rectification column**
	Concentration ethanol in feed [wt %]	0.08	Number of stages [-]	15/13	12/10
	Temperature feed [K]	308	Feed stage location [-]	4/3	8/7
	Pressure feed [mbar]	1000	Reflux ratio (mole based) [-]	1	9
Feed preheater	Temperature [K]	368	Condenser pressure [mbar]	1000	3500
	Pressure [mbar]	2000	Column pressure drop [mbar]	200	500
Mixer	Pressure [mbar]	2000			
Recycle heater	Temperature [K]	308	Distillate rate [kg/h]	15822	9360
	Pressure [mbar]	1000	Murphree efficiency [-]	0.43	0.60

Table 12: Simulation results of the water/ethanol separation- beer column

Beer column	**Bottom**		**Top**	
	Aspen Plus	**gPROMS**	**Aspen Plus**	**gPROMS**
Temperature [K]	378	378	355	355
Heat duty [kW]	15839	15730	-14685	-14524
Ethanol massflow [kg/h]	3	8	7117	7112
Water massflow [kg/h]	79637	79632	8705	8710
Ethanol massfraction [-]	37 ppm	96 ppm	0.45	0.45
Water massfraction [-]	1.00	1.00	0.55	0.55
Pressure [mbar]	1200	1200	1000	1000

Table 12 summarises the top and bottom performances of the beer column. The results of this separation column show as well good accordance. Only the heat duties and mass flows differ in small ranges. In gPROMS the amount of the ethanol flow is smaller at the top and higher at the bottom. The total heat demand in the gPROMS beer column is in this case smaller than in Aspen Plus. In proportion to the total feed input, those deviations can be neglected. The following figure shows the good agreement of the temperatures and ethanol fractions throughout the whole beer column.

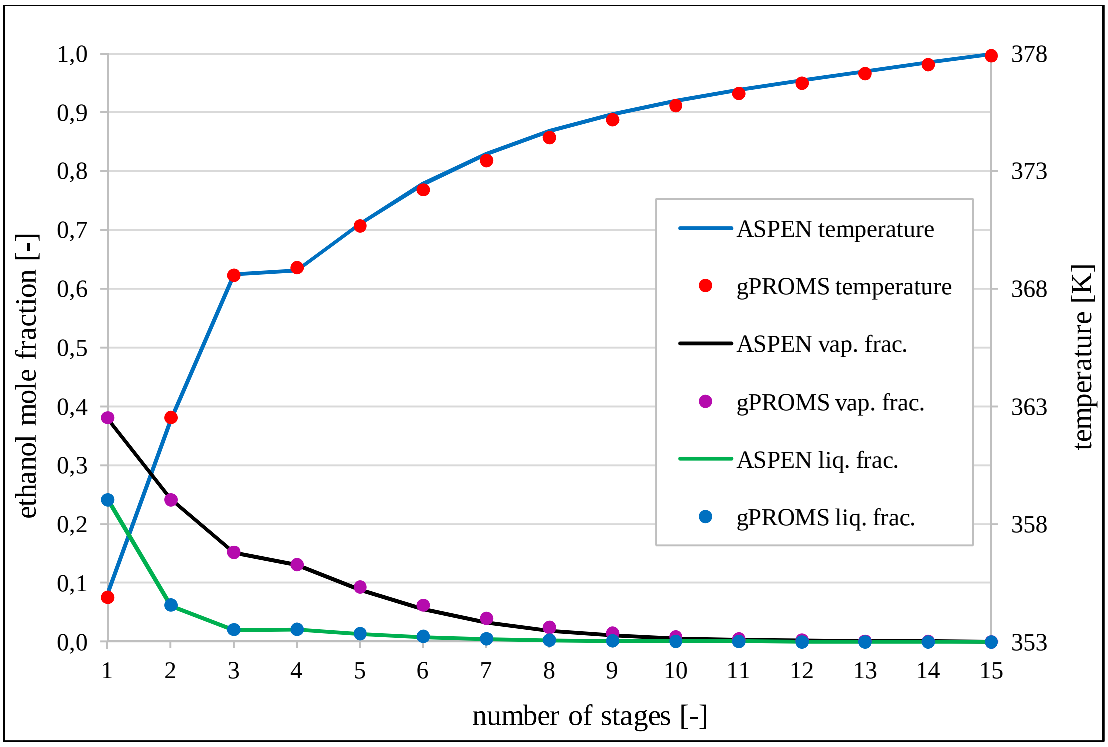


Figure 39: Temperature and ethanol fraction profile- beer column

Table 13: Simulation results of the water/ethanol separation- rectification column

Rectification column	Bottom		Top	
	Aspen Plus	gPROMS	Aspen Plus	gPROMS
Temperature [K]	417	417	388	387
Heat duty [kW]	30221	29685	-29352	-28823
Ethanol massflow [kg/h]	0.03	0.09	7117	7112
Water massflow [kg/h]	6462	6462	2243	2248
Ethanol massfraction [-]	5 ppm	14 ppm	0.76	0.76
Water massfraction [-]	1.00	1.00	0.24	0.24
Pressure [mbar]	4000	4000	3500	3500

Table 13 summarises the rectification column performance. The top and bottom values of the rectification column show exactly the same deviations than the beer column. In figure 40 it can be seen, that the column profiles obtained in gPROMS and Aspen Plus match well to each other.

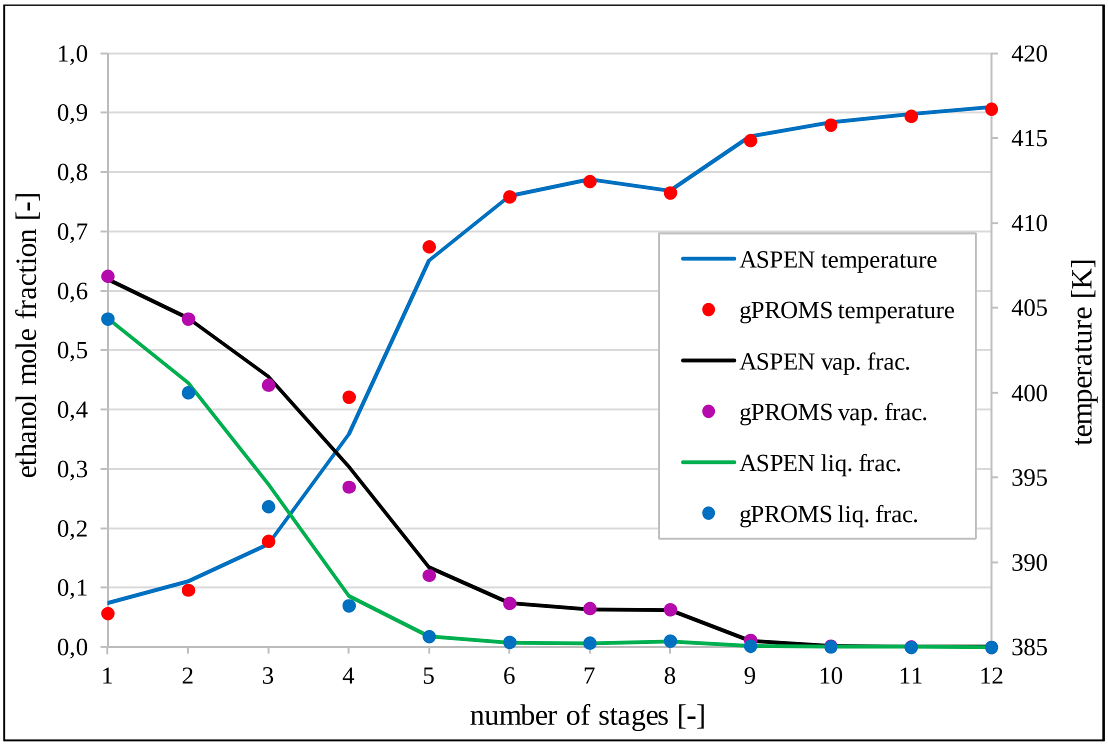


Figure 40: Temperature and ethanol fraction profile- rectification column

The following table shows the simulation results for the feed preheater, recycle heater and the mixer. The outlet temperature is the same in both simulations. The heat supply for feed preheating and the provided heat of the recycle heater are smaller in the Aspen Plus model.

Table 14: Simulation results of the water/ethanol separation- heater and mixer performance

	Feed preheater	**Recycle heater**	**Mixer**
	Heat duty [kW]	Heat duty [kW]	Outlet temperature [K]
Aspen Plus	6201	-860	364
gPROMS	6403	-889	364

The simulation results of the flowsheet total heat supply show that the gPROMS energy input is 218 kW higher than in Aspen Plus. Nevertheless, in proportion to the total heat amounts and the high material flows those deviations are as well negligible.

7 Summary and Outlook

The aim of this thesis is to develop a distillation column model for binary mixtures within the environment of gPROMS.

For the estimation of the initial operating conditions the short-cut model WUG was developed. The precise steady state column is designed as an n-equilibrium stage model. Therefore the following assumptions are made:

- no accumulation of material or energy on the stages (steady-state)
- no reaction terms
- consideration of equilibrium stages:
    - perfect mixing of the vapour and liquid phase
    - thermal equilibrium for every component on each stage
- no heat loss

The mass and energy balances, equilibrium and summation conditions completely describe the stages. Additionally the Murphree efficiency and pressure drop specifications enable the user to consider deviations from the equilibrium state. For handling the complex system of non-linear coupled equations an initialisation procedure was developed. The physical properties for both models are retrieved from Multiflash, which is a standalone software to generate files for the application in gPROMS.

In order to achieve good results with the models the physical properties need to be cross referenced with literature data first. The bubble and dew temperature plot in section 6.1.1 makes visible that the data provided by Multiflash are broadly the same as the experimental ones and the ones determined with Aspen Plus. The enthalpies calculated with Multiflash show small differences in comparison to Aspen Plus, which are visible in the deviations of the heat duties within the two flowsheet calculations. Those negligible differences are caused due to differences in the property databases.

The simulation results of the WUG-model are compared with the DSTWU model (Aspen Plus) in section 6.1.2. It can be seen that the results are in good accordance. In regard to the WUG-model, it has to be considered, that short-cut methods are usually not valid for azeotropic mixtures. Above the azeotropic point the number of stages will take on a negative value. However, cautiously it might be applied for compositions below the azeotropic point.

To validate the steady state column model, it was used in flowsheets. The simulation results of two different binary systems are cross referenced with Aspen Plus. Radfrac was chosen as column model in Aspen Plus. The comparisons of the water/methanol separation in section 6.1.3 and the water/ethanol separation in section 6.1.4 indicate that the column profiles, bottom and top temperatures, material flows and concentrations determined with the gPROMS column model show hardly any differences to the Radrac column. The use of the equilibrium temperature instead of the dew temperature in the condenser does not affect the simulation results in any negative way.

The designed initialisation procedure is intended for a column model up to 50 stages. The validation with the ethanol/water mixture underlines the flexibility and robustness for different user defined specifications of the model. The implementation of an effective initialisation procedure emerged to be a difficult part in developing the column model. The reason for that is the requirement of the availability of multiple changeable input specifications. The initialisation procedure was foresighted designed as system-independent as possible. In regard to new component systems this could unfortunately not be implemented completely. The enthalpy initialisation for the column stages and the equilibrium initialisation procedure for a defined recovery of the light key component in the reboiler could cause problems by the application of other component systems. Further tasks could therefore include the modification of the available initialisation procedure in order to design a completely component-independent and universally valid initialisation procedure.

During programming, different input variations were simulated. However, it has to be considered, that only a small number of combinations could be tested and validated. Errors could be handled by changing the input specifications and additionally applying a process schedule to meet the desired specifications.

Furthermore the column model was never tested in regard to what happens by reaching the azeotropic point. Since there is no restriction implemented at the moment, the model might not recognize the azeotropic point and the simulation will continue.

Up to now another fact is that the feed was always added as subcooled liquid. To use a vapour input stream, the model must probably be adapted.

Summing up it can be stated, that the developed column model provides good results and serves as a robust and flexible base model for further adaptions.

As future outlook investigation and validation of different binary systems should be considered. In regard to azeotropic mixtures, it would be interesting to evaluate the behaviour by reaching the azeotropic point. Additionally, examinations with other activity coefficient models could be made. Besides that, the model could be adapted in different ways, e.g. extended with a second feed stream and convert the condenser to a partial condenser. Another point, which should be applied with priority, is to find an algorithm which automatically identifies the heavy and light key component. It might as well be considered to extend the system to a multicomponent system. What needs to be considered, however, is that multicomponent systems will be very challenging regarding the design of an effective initialisation procedure.

Regarding the tutorial, creating a Multiflash file, the implementation of foreign objects and the retrieve of physical properties could be taken into account as extension topics. Further additions could be the definition and application of selectors and the implementation of an initialisation procedure.

8 Nomenclature

Variables

α	Relative volatility	[-]
$\bar{\alpha}$	Geometric average value relative volatilities	[-]
γ	Liquid activity coefficient	[-]
ϑ	Underwood-factor	[-]
$\theta_{LK}, \beta_{\frac{LK}{HK}}$	Winn-equation constants at a fixed pressure	[-]
φ_i^V	Vapour fugacity coefficient of component i	[-]
φ_i^L	Liquid fugacity coefficient of component i	[-]
ψ	Variable in Molokanov equation	[-]
a	Abscissa value in Newton-method	[-]
$\dot{B}$	Bottom molar flow	[mol/s]
$\dot{b}$	Bottom mass flow	[kg/h]
c_P	Specific molar heat capacity	[J/mol K]
$\dot{D}$	Distillate molar flow	[mol/s]
$\dot{d}$	Distillate mass flow	[kg/h]
E	Murphree efficiency	[-]
f	Fugacity	[Pa]
$\dot{F}$	Feed molar flow	[mol/s]
$\dot{f}$	Feed mass flow	[kg/h]
f(a)	Function value for value a (Newton-method)	[-]
f´(a)	First derivative of function f at point a (Newton-method)	[-]
$\dot{G}$	Vapour molar flow	[mol/s]
$\dot{G}^+$	Vapour molar flow stripping section	[mol/s]
h^v	Molar vapour enthalpy	[J/mol]
h^l	Molar liquid enthalpy	[J/mol]
h_b	Molar liquid enthalpy at boiling temperature	[J/mol]
h_{check}	Enthalpy check variable (initialisation variable)	[J/s]
Δh_V	Difference of molar enthalpies of evaporation	[J/mol]
h_V	Molar enthalpies of evaporation	[J/mol]

k	Feed stage location	[-]
K	Distribution coefficient	[-]
$\dot{L}$	Liquid molar flow	[mol/s]
$\dot{L}^{+}$	Liquid molar flow stripping section	[mol/s]
M	Molecular weight	[g/mol]
n	Number of stages	[-]
nc	Number of components	[-]
n_r	Number of stages above the feed stage (rectification part)	[-]
n_s	Number of stages below the feed stage (stripping part)	[-]
ns	Number of streams	[-]
$n_{th,min}$	Minimum number of (theoretical) stages	[-]
n_{th}	Number of (theoretical) stages	[-]
NT	Number of stages	[-]
p	Pressure	[Pa]
p_0	Saturation vapour pressure	[Pa]
p_i	Partial pressure of component i	[Pa]
$\varDelta p$	Pressure drop	[Pa]
$\dot{Q}$	Heat flow	[J/s]
q	Caloric factor	[-]
r	Recovery	[-]
$\dot{R}$	Reflux molar flow	[mol/s]
R	Specific gas constant	[kJ / kgK]
$\dot{S}$	Boilup molar flow	[mol/s]
T	Temperature	[K]
ΔT	Temperature difference	[K]
T_C	Critical temperature	[K]
v	Acutal reflux ratio	[-]
v^{+}	Acutal reflux ratio stripping section	[-]
v_{min}	Mimimum reflux ratio	[-]
w	Mass fraction	[-]
x	Liquid molar fraction	[-]
x^{*}	Theoretical equilibrium liquid molar fraction	[-]
y	Vapour molar fraction	[-]

y^*	Theoretical equilibrium vapour molar fraction	[-]

Indices

1,2, ..., n	Numbering sequence (number of stages,...)
B	Bottom
C	Condenser
D	Distillate
eq	Equilibrium
F	Feed
HK	Heavy key component
i	Component i
in	Incoming stream variables
j	Component j
L	Liquid
LK	Light key component
M	Methanol
out	Outgoiung stream variables
R	Reboiler
V	Vapour

Abbreviations

BDNLSOL	Block Decomposition NonLinear SOLver
CA	Case Indicator
gPROMS	general PROcess Modelling System
GUI	Graphical User Interface
IPPFO	Ideal Physical Properties Foreign Object
LLE	Liquid liquid equilibrium
MFL	Multiflash-file
NRTL	Non-Random-Two-Liquid
PSE	Process System Enterprise
UNIQUAC	Universal Quasi-Chemical Theory
UNIFAC	Universal Functional Group Activity Coefficient
VLE	Vapour liquid equilibrium
VLL	Vapour liquid liquid
WUG	Winn-Underwood-Gilliland

9 Bibliography

[1] Aspen Technology, Inc. AspenTech. (2012). Process modeling environment for conceptual design, optimization, and performance monitoring of chemical processes. United States.

[2] Aspen Technology, Inc. AspenTech. (2014). Aspen Plus Help, Aspen Plus V7.3 - aspenONE: Winn Method in DSTWU.

[3] Barton, P. I. (1992). The modelling and simulation of combined discrete/continuous processes, Doctoral thesis. University of London.

[4] Behr, A. A. (2010). *Einführung in die Technische Chemie.* Heidelberg: Spektrum Akademischer Verlag.

[5] Christa, S., Almpanis-Lekkas, O., & Wukovits, W. (2014). *Poster: Modelling and Initialisation of a Distillation Column for the Separation of Binary Systems*. Vienna.

[6] Coker, A. K. (2010). *Ludwig's Applied Process Design for Chemical and Petrochemical Plants: Volume 2: Distillation, packed towers, petroleum fractionation, gas processing and dehydration.* Oxford: Gulf Professional Publishing.

[7] Forbes, R. J. (1970). *A Short History of the Art of Distillation: From the Beginning Up to the Death of Cellier Blumenthal.* White Mule Press.

[8] Friedl, A. (2000). *Lecture notes for "Thermische Verfahrenstechnik 1" course 159.731, version 3.0.* Vienna University of Technology.

[9] Friedl, A. (2010). *Lecture notes for "Thermische Verfahrenstechnik 2": course 159.995.* Vienna University of Technology.

[10] Friedl, A., Harasek, M., & Wukovits, W. (2010). *Lecture notes for "Prozesssimulation": course 166.038.* Vienna University of Technology.

[11] Huber, O. (2005). Simulation der destillativen Abtrennung von Bioethanol, Master thesis. Vienna University of Technology.

[12] Jana, A. K. (2009). *Process Simulation and Control using AspenTM.* New Delhi: PHI Learning Private Limited.

[13] KBC Advanced Technologies plc. (2014). Retrieved may 2014, from http://www.kbcat.com/technology/infochem-software

[14] Kiss, A. A. (2013). *Advanced distillation technologies: Design, Control and Applications.* Chichester: Wiley.

[15] Lüdecke, C., & Lüdecke, D. (2000). *Thermodynamik: physikalisch-chemische Grundlagen der thermischen Verfahrenstechnik.* Berlin, Heidelberg: Springer.

[16] Perry, R. H. (1997). *Perry's chemical engineers' handbook - 7th ed.* New York: McGraw-Hill.

[17] Process Systems Enterprise Limited. (1997-2012). Model Developer Guide, Release v3.6. London.

[18] Process Systems Enterprise Limited. (2014a). Retrieved june 2014, from http://www.psenterprise.com/modelbuilder.html

[19] Process Systems Enterprise Limited. (2014b). Retrieved may 2014, from http://www.psenterprise.com/gproms/physprops/index.html

[20] Ramirez, W. (1997). *Computational Methods for Process Simulation, Second Edition.* Oxford: Butterworth-Heinemann.

[21] Sattler, K. (1988). *Thermische Trennverfahren: Grundlagen, Auslegung, Apparate.* Weinheim, Basel, Cambridge: VCH Verlagsgesellschaft mbH.

[22] Schönbucher, A. (2002). *Thermische Verfahrenstechnik: Grundlagen und Berechnungsmethoden für Ausrüstungen und Prozesse.* Berlin, Heidelberg, New York: Springer.

[23] Seider, W. D. (2010). *Product and Process Design Principles: Synthesis, Analysis, and Evaluation.* Asia: John Wiley & Sons (Asia).

[24] Stephan, P. S. (2010). *Thermodynamik: Grundlagen und technische Anwendungen-Band 2: Mehrstoffsysteme und chemische Reaktionen.* Berlin, Heidelberg: Springer.

[25] VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC). (2013). part D3: Stoffwerte von sonstigen reinen Fluiden. In *VDI-Wärmeatlas, 11th edition.* Berlin: Springer.

List of figures

List of tables

10 Appendix

10.1 McCabe Thiele calculations

Given parameters of chapter 2.3.1:

Table 2: Process parameters of the water/methanol separation

Parameter	**Unit**	**Value**
Concentration methanol in feed	[wt%]	5
Concentration methanol in distillate	[wt%]	95
Recovery methanol	[%]	90
Feedflow	[kg/h]	10000
Temperature feed after preheating	[K]	313

Molecular weights [VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC), 2013]:

$M_M = 32.04 \frac{g}{mol}$ $\quad\quad$ $M_W = 18.02 \frac{g}{mol}$

Feed:

$\dot{f} = 10000 \frac{kg}{h}$ $\quad\quad$ $w_{M,F} = 0.05$ $\quad\quad$ $w_{W,F} = 0.95$

Overall balance and recovery r:

$$r = \frac{\dot{d} \cdot w_{M,D}}{\dot{f} \cdot w_{M,F}} => \dot{d} = \frac{0.9 \cdot 10000 \cdot 0.05}{0.95} = 473.6842 \frac{kg}{h} \tag{10.1}$$

$$\dot{f} = \dot{d} + \dot{b} => \dot{b} = 9526.32 \frac{kg}{h} \tag{10.2}$$

$$\dot{f} \cdot w_{M,F} = \dot{d} \cdot w_{M,D} + \dot{b} \cdot w_{M,B} => w_{M,B} = 0.00524862 \tag{10.3}$$

Molar fractions:

$$x_{M,F} = \frac{\frac{0.05}{32.04}}{\frac{0.05}{32.04}+\frac{0.95}{18.02}} = 0.02874 \quad\quad x_{M,D} = \frac{\frac{0.95}{32.04}}{\frac{0.95}{32.04}+\frac{0.05}{18.02}} = 0.9144 \quad\quad 10.4, 10.5$$

$$x_{M,B} = \frac{\frac{0.0052}{32.04}}{\frac{0.0052}{32.04}+\frac{1-0.0052}{18.02}} = 0.00296 \qquad 10.6$$

Intersection line:

$$y = \frac{q}{q-1} \cdot x - \frac{x_F}{q-1} \qquad 2.20$$

The caloric factor q can be calculated as follows (for liquid mixtures):

$$q = 1 + \frac{c_{P,F} \cdot \Delta T}{\Delta h_{V,F}} \qquad 10.7$$

For the calculations of the enthalpy of evaporation h_V and the specific heat capacities c_P at feed temperature the following approaches are used [VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC), 2013]:

$$h_v = R \cdot T_C \cdot \left[A \cdot \left(1 - \frac{T}{T_C}\right)^{\frac{1}{3}} + B \cdot \left(1 - \frac{T}{T_C}\right)^{\frac{2}{3}} + C \cdot \left(1 - \frac{T}{T_C}\right) + D \cdot \left(1 - \frac{T}{T_C}\right)^2 + E \cdot \left(1 - \frac{T}{T_C}\right)^6 \right] \qquad 10.8$$

$$c_P = R \cdot \left(\frac{A}{1 - \frac{T}{T_C}} + B + C \cdot \left(1 - \frac{T}{T_C}\right) + D \cdot \left(1 - \frac{T}{T_C}\right)^2 + E \cdot \left(1 - \frac{T}{T_C}\right)^3 + F \left(1 - \frac{T}{T_C}\right)^4 \right) \qquad 10.9$$

Data used for the calculations:

Table 15: Physical properties for the calculations, data from [VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC), 2013]

Physical properties	**water**		**methanol**	
T_C [K]	647.1		519.9	
R [kJ / kgK]	0.461526		0.259487	
	equ. 10.8	**equ. 10.9**	**equ. 10.8**	**equ. 10.9**
A	6.85307	0.2399	5.87513	0.5687
B	7.43804	12.8647	13.91543	14.11
C	-2.937595	-33.6392	-5.817880	-11.9505
D	-3.282093	104.7686	-5.692542	-23.4782
E	8.397378	-155.4709	6.867206	57.5551
F	-	92.3726	-	-25.6027

The obtained data are summarised in table 16.

Table 16: Calculated physical properties of water and methanol

Physical properties	$h_{V,F}$ (313.15 K) [J/mol]	$c_{P,F}$ (313.15 K) [J/mol K]
water	43307.7	75.29
methanol	36751.5	83.59

$$q = 1 + \frac{\left[c_{P,F,water} \cdot (1 - x_{F,methanol}) + c_{P,F,methanol} \cdot x_{F,methanol}\right] \cdot \left(T_{F,boiling} - T_F\right)}{h_{V,F,water} \cdot (1 - x_{F,methanol}) + h_{V,F,methanol} \cdot x_{F,methanol}} =$$

$$q = 1 + \frac{75.529 \frac{J}{mol\,K} \cdot 54.77\,K}{43119.28 \frac{J}{mol}} = 1.09594$$

$$\underline{\boldsymbol{y = 11.42 \cdot x - 0.29957}}$$

With the graphically determined reflux ration *v*, the enriching line gets:

$$y = \frac{x_D}{v + 1} + \frac{v}{v + 1} \cdot x \qquad 2.12$$

$$\underline{\boldsymbol{y = 0.79\,x + 0.1921}}$$

gPROMS

Getting started
building and running processes
within the gPROMS ModelBuilder®

INTRODUCTION

This tutorial should help beginners to get started and to familiarise themselves with gPROMS. A detailed guide and simple examples have the intention to provide an overview of the capabilities and functionalities of the gPROMS environment. The user should gain a basic knowledge about programming simple unit operations, creating a flowsheet in gPROMS, executing a simulation and viewing results without any previous knowledge or additional support.

What is gPROMS

gPROMS (**g**eneral **PRO**cess **M**odelling **S**ystem) is an equation-oriented modelling and simulation software developed by Process System Enterprise (PSE). The gPROMS ModelBuilder® combines modelling capabilities with a process flowsheeting environment and is applicable for both, steady-state and dynamic simulation. Other capabilities are advanced parameter estimation and steady-state, respectively dynamic optimisation [http://www.psenterprise.com/].

SESSIONS IN THIS TUTORIAL:

BASICS

- Start gPROMS
- The gPROMS main window
- Help/ Documentation
- The gPROMS Project

SIMPLE MIXING MODEL

- The Project tree
- Variable Types
- Models
- The gPROMS language
- Processes

SUBOMDELS

- Advantages of submodels
- Hierarchical structure
- Create a submodel for the "simple mixing" model

FLOWSHEET MIXER

- Connection Types
- Units for flowsheeting (Source, Mixer, Sink)
- Flowsheeting
- Schedules
- Annotations

PML (Process Model Library)

- Introduction
- PML library
- Example Flowsheets

FOR PRACTICE

BASICS

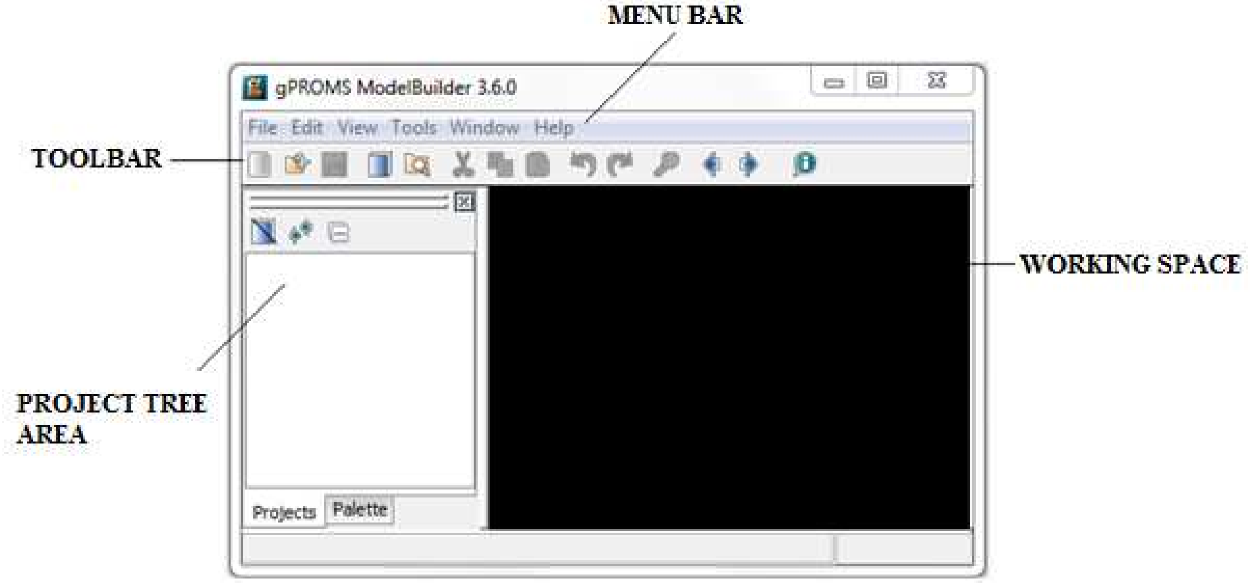
How to start gPROMS

To open gPROMS you can either

- use the the **ModelBuilder 3.n.x** icon on your desktop ModelBuilder 3.6.0
- or click on **Start (Windows)** , select **all programs**, open the **Process Systems Enterprise** and the **ModelBuilder 3.n.xx** folder. Then select **ModelBuilder 3.n.xx**.

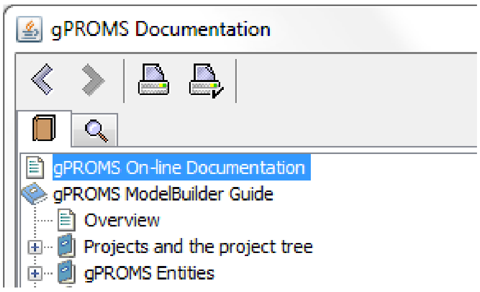
The gPROMS main window

After starting gPROMS the main window appears:



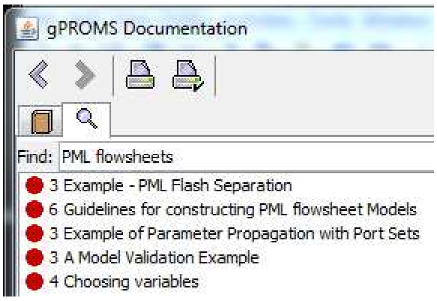
Help/ Documentation

If descriptions and help are needed, click on **Help** in the **menu Bar** and select **Documentation**. The gPROMS documentation allows to search for keywords in the Table of Contents (left figure) or directly by Search (right figure).



gPROMS Documentation

gPROMS On-line Documentation
gPROMS ModelBuilder Guide
Overview
Projects and the project tree
gPROMS Entities



gPROMS Documentation

Find: PML flowsheets

- 3 Example - PML Flash Separation
- 6 Guidelines for constructing PML flowsheet Models
- 3 Example of Parameter Propagation with Port Sets
- 3 A Model Validation Example
- 4 Choosing variables

The gPROMS Project

Create a new project / open an existing project

To create a **new project** either

- select **File** on the **menu bar** and then **New** or
- use the -icon in the toolbar.

To open an existing file:

- select the **Open** command in the **File menu** or
- click on the -icon in the **toolbar** and choose the project you want to open. Or
- double click on the file in Windows.

The Project appears in the project tree area (shown in the graphic below).

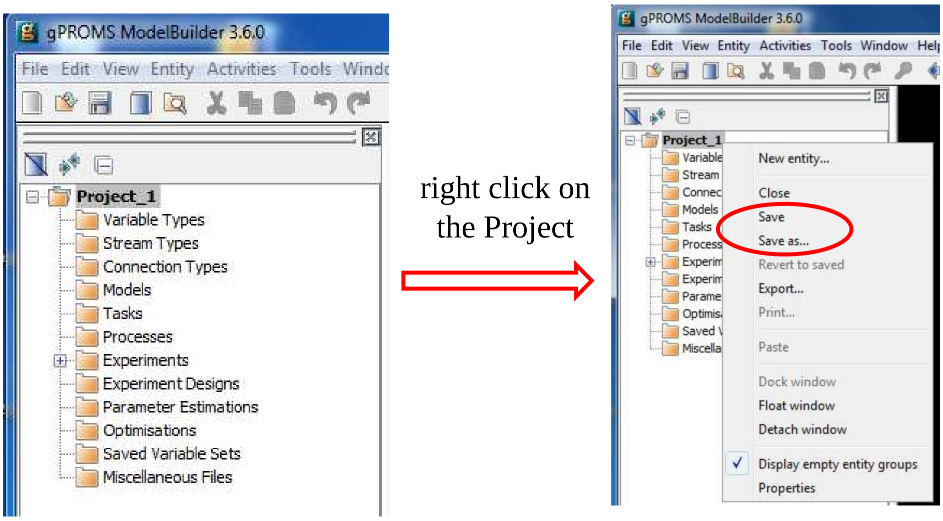
Rename and Save a project

Rename the Project to "tutorial":

1. choose **Save (Ctrl+S)** or **Save as…** in the **File Menu** or in the **Project Menu** after right click on the **Project**.
2. or just use the **Save symbol** on the **toolbar**.

Choose your **working folder** and type "tutorial" as name. Click **Save**.

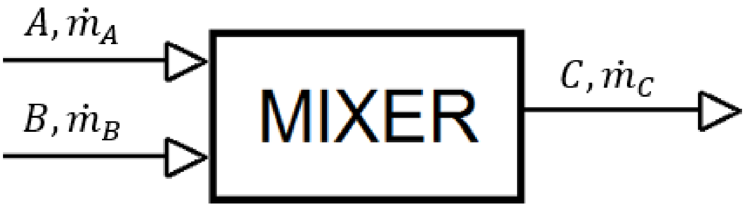
After defining / redefining the name in the Project tree will change automatically.



For a better understanding of the project tree contents, they will be explained step by step with the "simple mixing" and the "flowsheet mixer" exercises.

SIMPLE MIXING

A defined amount of components (methanol and water) in the streams A and B are mixed.



Known variables:

$$\dot{m}_A = 35\ \frac{kg}{h}; \qquad w_{A,meth.} = 0.12; \qquad \dot{m}_B = 27\ \frac{kg}{h}; \qquad w_{B,water} = 0.37;$$

With the knowledge of the mass fractions it's easy to find the complete composition of the input streams.

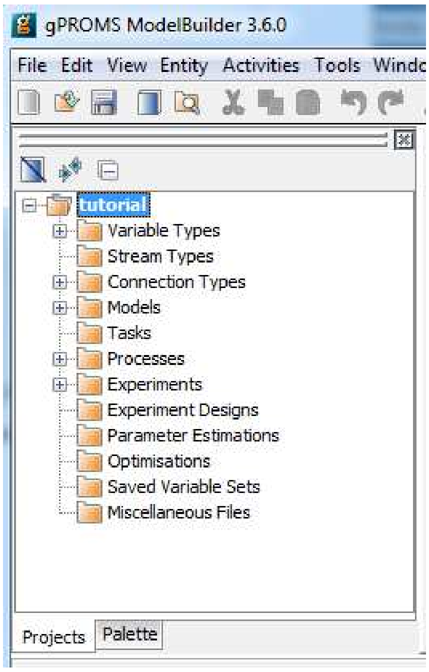
Therefore three unknown variables are left:

$\dot{m}_C$, $w_{C,meth.}$, $w_{C,water}$

To solve this model the following steps have to be done:

- Define variable types
- Create a model that includes a list of all used variables, all mathematical correlations written in equations and the variable assignment.

The Project Tree



The most important sections within the project tree are:

- Variable Types
- Connection Types
- Models
- Processes.

To breakdown the Project Tree right click on the **Project** and uncheck the **Display empty entity groups** check box.

Variable Types:

Variables are time variant, adaptable factors, which can be calculated with equations. All variable Types (e.g mass, fraction, length…) used in the models have to be specified:

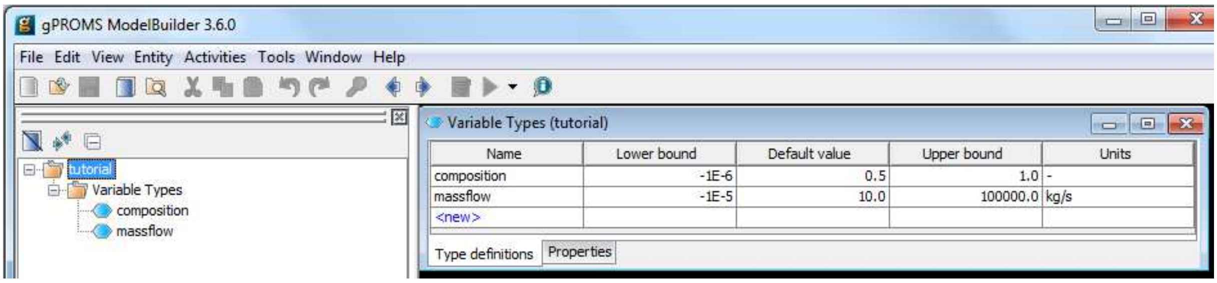
- Definition of the variable units (e.g. m or kg)
- Set realistic numerical limits for the Lower/Upper bounds. To avoid errors chose small negative numbers (e.g. -1E-7) as low boundaries instead of zero (e.g. for the mass- or mole fraction).
- Declare default values as initialisation.

Watch out that you use the “point” as “comma”.

Create a new Variable Type

- Right-click on the **Project** folder and select **New Entity…** or use **Ctrl+E**
- choose an adequate name for the variable type (use the underline sign if the name consists of more than one words, it won’t work with “space” or “-“)
- select **Variable Type** in the **dropdown menu** and
- unselect the **Use template?** checkbox.
- Click **OK**
- Set the **Lower** and **Upper bounds**, the **Default value** and the **Units.**

To define more variable types, just list them in the lines below.


gPROMS ModelBuilder 3.6.0
File Edit View Entity Activities Tools Window Help
tutorial
Variable Types
composition
massflow
Variable Types (tutorial)
Type definitions
Properties

Name	Lower bound	Default value	Upper bound	Units
composition	-1E-6	0.5	1.0	-
massflow	-1E-5	10.0	100000.0	kg/s
<new>				

By closing the **Variable type** window the changes will be saved automatically.

The next step is to create a model.

Models

In gPROMS it is possible to create simple basic (sub -) models, which can be used as standalone units or as base elements for developing superior models (flowsheeting). The first step is to create a standalone unit. Afterwards we will build up an easy superior model with sub models and connection types.

Create a new Model:

- Right-click on the **Project** folder and select **New Entity…** or use **Ctrl+E**
- Use “mixing” as entity name
- select **Model** in the **dropdown menu** and
- unselect the **Use template?** checkbox
- click **OK**.

General Model user interface

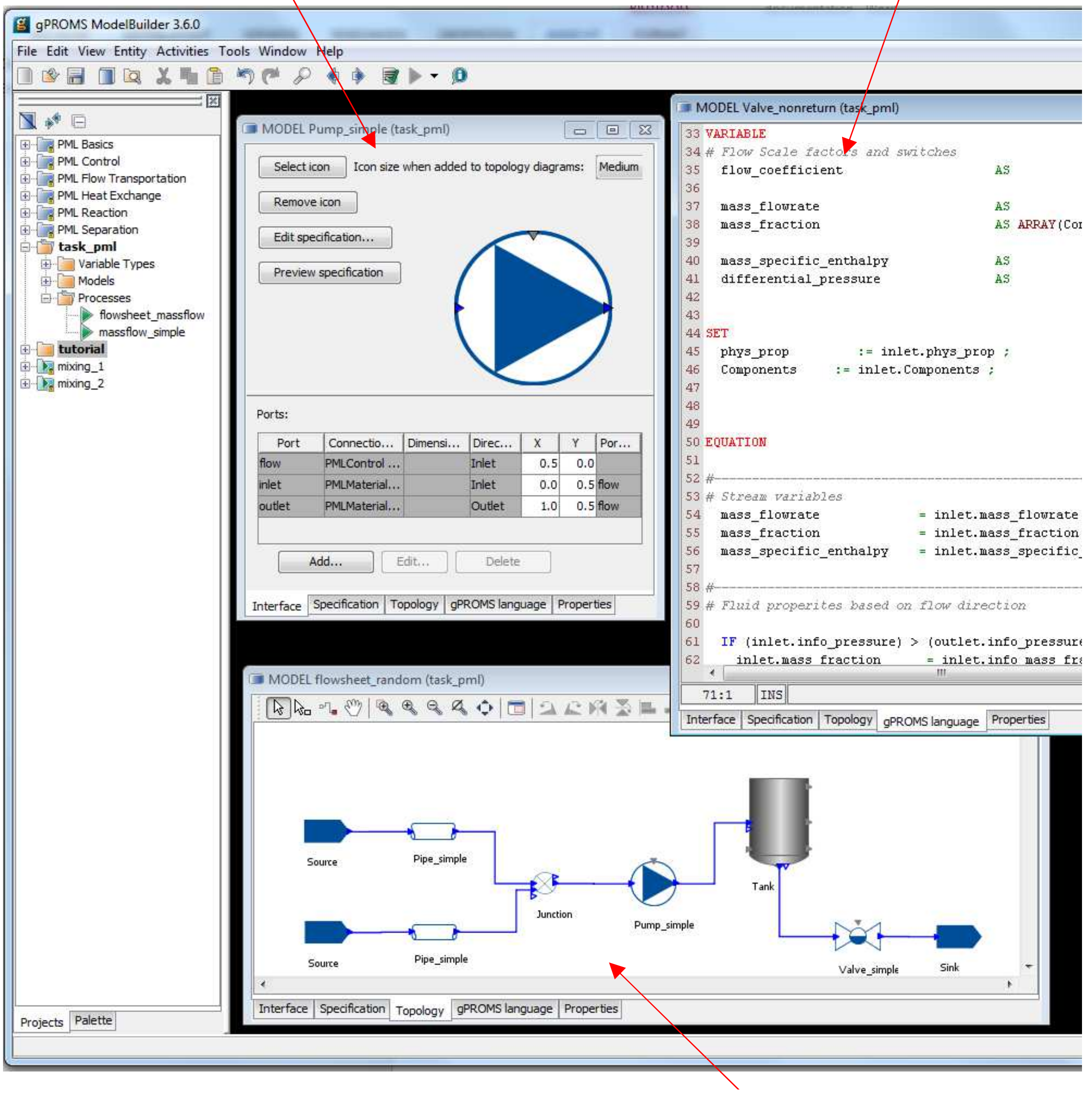
The general gPROMS model includes the following tabs:

- Interface: The model Interface can be used to adapt specifications (input and report), add connection ports and change the graphical appearance (icon) of the Model in the Topology
- Specification: The specification window shows the xml code of the specifications adapted in the Interface (not important for the tutorial)
- Topology: The Topology is the graphical interface used for flowsheeting
- gProms language: The gProms language is the worksheet, which contains the model description. Therefore it needs to include all variables, parameters and equations that are needed to exactly determine a system.
- Properties: with no importance for that tutorial

As an illustration three of them are shown in the screenshot below.

Interface- for connection- and specification adaptation

gPROMS language- editing



Icon size when added to topology diagrams: Medium

Ports:

Port	Connectio...	Dimensi...	Direc...	X	Y	Por...
flow	PMLControl ...		Inlet	0.5	0.0	
inlet	PMLMaterial...		Inlet	0.0	0.5	flow
outlet	PMLMaterial...		Outlet	1.0	0.5	flow

```
VARIABLE
# Flow Scale factors and switches
  flow_coefficient                        AS

  mass_flowrate                           AS
  mass_fraction                           AS ARRAY(Cor

  mass_specific_enthalpy                  AS
  differential_pressure                   AS


SET
  phys_prop         := inlet.phys_prop ;
  Components     := inlet.Components ;



EQUATION

#---------------------------------------------------------------
# Stream variables
  mass_flowrate               = inlet.mass_flowrate
  mass_fraction               = inlet.mass_fraction
  mass_specific_enthalpy      = inlet.mass_specific_

#---------------------------------------------------------------
# Fluid properites based on flow direction

  IF (inlet.info_pressure) > (outlet.info_pressure
    inlet.mass_fraction       = inlet.info_mass_fra
```

71:1 INS

Topology- surface for flowsheeting

The gPROMS language

For the exact determination of the “simple mixing” system and for calculating the unknown variables, the following equations are required:

- mass balance
- composition balances of water and methanol

The basic model is intended to contain the following issues:

- List and declaration of used Variables and Parameters - Variables must be allocated to Variable Types
- Parameter set
- Model equations
- Assignment of known variables

```
VARIABLE
massflow_A, massflow_B, massflow_C                          AS massflow
composition_A_water, composition_A_methanol,
composition_B_water, composition_B_methanol,
composition_C_water, composition_C_methanol                 AS composition

EQUATION
#mass balance
massflow_C = massflow_A + massflow_B;

#composition balance methanol

composition_C_methanol = (composition_A_methanol * massflow_A +
composition_B_methanol * massflow_B)/massflow_C;

#composition balance water

composition_C_water = (composition_A_water * massflow_A + composition_B_water *
massflow_B)/massflow_C;
```

```
ASSIGN

#composition and massflow A

massflow_A := 35;
composition_A_methanol := 0.12;
composition_A_water := 0.88;

#composition and massflow B

massflow_B := 27;
composition_B_methanol := 0.63;
composition_B_water := 0.37;
```

Use:

“ = “ in EQUATION(S)

“ := “ in SET and ASSIGN (fixed values)

Always set “ ; “ after the expressions of Equations, Assignments and Sets!

There is as well the possibility to use two more equations (sum of compositions (water and methanol) = 1), but it’s not necessary for the understanding of the gPROMS syntax.

TIP:

Don’t forget to **save** your work in between

- Use the **save symbol** 💾 or
- **Ctrl+S**.

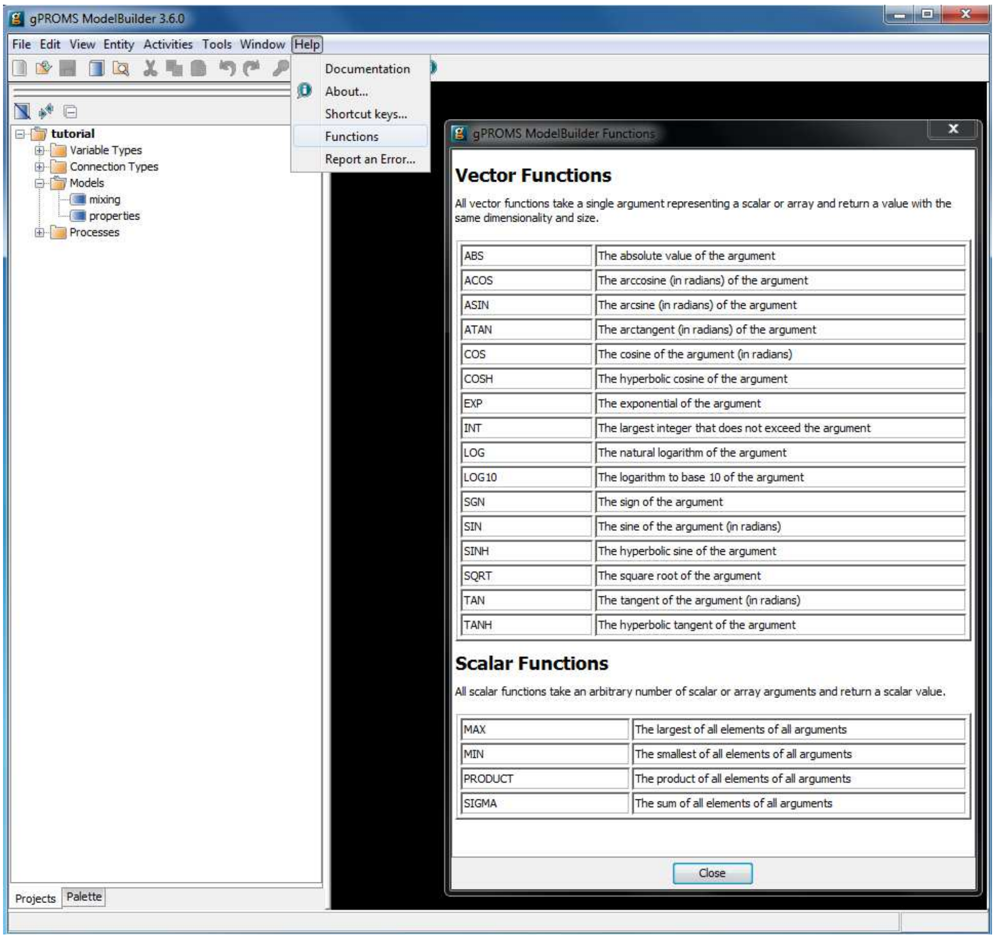
To write a comment in the gPROMS language use the “Hash”-key (#).

gPROMS has a special Syntax-order. To avoid mistakes in the gPROMS language, watch out that you follow that order. It is added at the end of the tutorial.

Functions in the gPROMS language

To display a brief description of the built-in functions in the gPROMS model builder:

- Open the **Help** command window in the **menu bar**
- Click on **Functions**



Vector Functions

All vector functions take a single argument representing a scalar or array and return a value with the same dimensionality and size.

ABS	The absolute value of the argument
ACOS	The arccosine (in radians) of the argument
ASIN	The arcsine (in radians) of the argument
ATAN	The arctangent (in radians) of the argument
COS	The cosine of the argument (in radians)
COSH	The hyperbolic cosine of the argument
EXP	The exponential of the argument
INT	The largest integer that does not exceed the argument
LOG	The natural logarithm of the argument
LOG10	The logarithm to base 10 of the argument
SGN	The sign of the argument
SIN	The sine of the argument (in radians)
SINH	The hyperbolic sine of the argument
SQRT	The square root of the argument
TAN	The tangent of the argument (in radians)
TANH	The hyperbolic tangent of the argument

Scalar Functions

All scalar functions take an arbitrary number of scalar or array arguments and return a scalar value.

MAX	The largest of all elements of all arguments
MIN	The smallest of all elements of all arguments
PRODUCT	The product of all elements of all arguments
SIGMA	The sum of all elements of all arguments

Processes

Convert a model into a process

To execute a calculation the model has to be converted into a **Process**:

- Right-click on the **model** you want to convert
- Choose **Edit PROCESS**

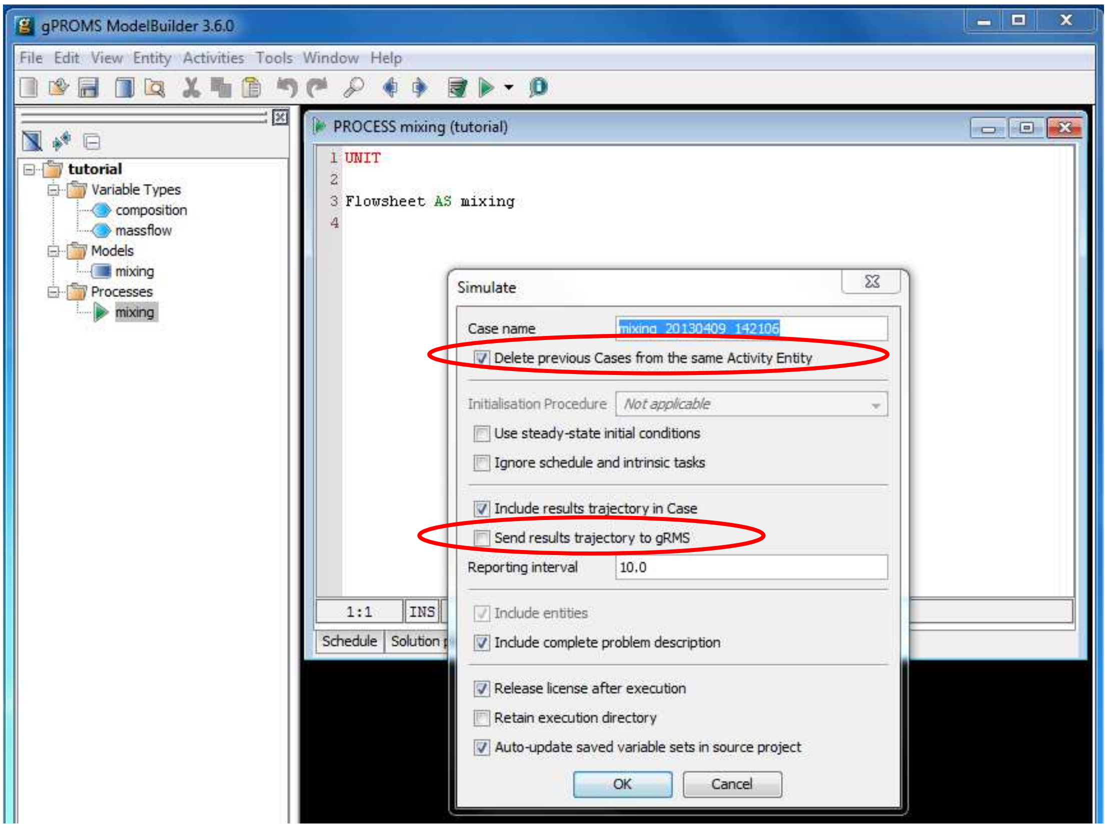
The “mixing-PROCESS” appears as separate **window** and as well in the **Project tree (folder Processes)**.

Run the Simulation

To run the Simulation click on the **play-icon** ▶ in the **Toolbar** or right-click on the **Process (mixing)** and choose **Simulate...**

The **Simulate** window appears:

- select **Delete previous Cases from the same Activity Entity** and
- uncheck the box **Send results trajectory to gRMS**
- click OK to start the simulation


gPROMS ModelBuilder 3.6.0
File Edit View Entity Activities Tools Window Help
tutorial
Variable Types
composition
massflow
Models
mixing
Processes
mixing
PROCESS mixing (tutorial)
1:1
INS
Schedule
Solution p
OK
Cancel

```
UNIT

Flowsheet AS mixing
```

Simulate

Case name: mixing_20130409_142106

☑ Delete previous Cases from the same Activity Entity

Initialisation Procedure: *Not applicable*

□ Use steady-state initial conditions

□ Ignore schedule and intrinsic tasks

☑ Include results trajectory in Case

□ Send results trajectory to gRMS

Reporting interval: 10.0

☑ Include entities

☑ Include complete problem description

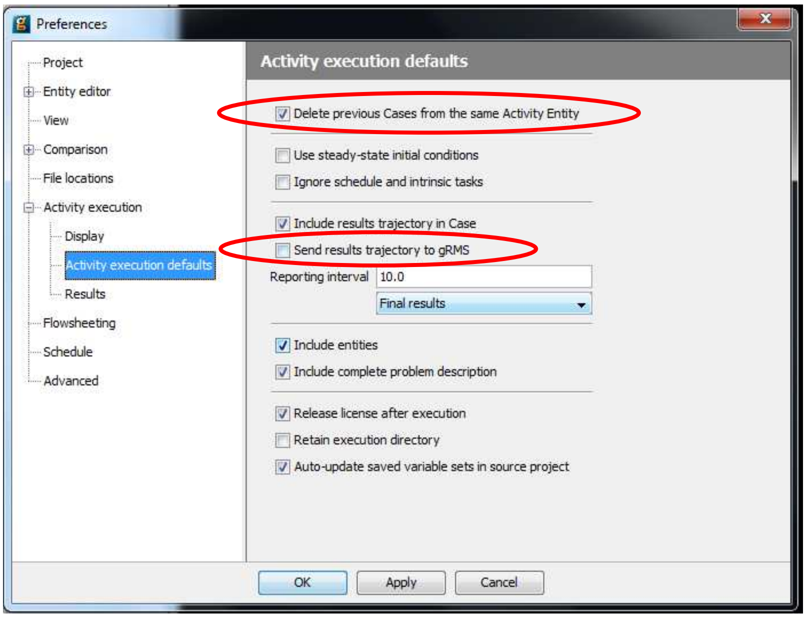
☑ Release license after execution

□ Retain execution directory

☑ Auto-update saved variable sets in source project

To change those settings for all the Simulations:

- Open the **Edit** Menu on the **Menu bar**
- Select **User Preferences…**
- Then **Activity execution** and **Activity execution defaults**
- select **Delete previous Cases from the same Activity Entity** and
- uncheck the box **Send results trajectory to gRMS**
- click **OK**



Preferences

- Project
- Entity editor
- View
- Comparison
- File locations
- Activity execution
  - Display
  - Activity execution defaults
  - Results
- Flowsheeting
- Schedule
- Advanced

Activity execution defaults

☑ Delete previous Cases from the same Activity Entity

□ Use steady-state initial conditions
□ Ignore schedule and intrinsic tasks

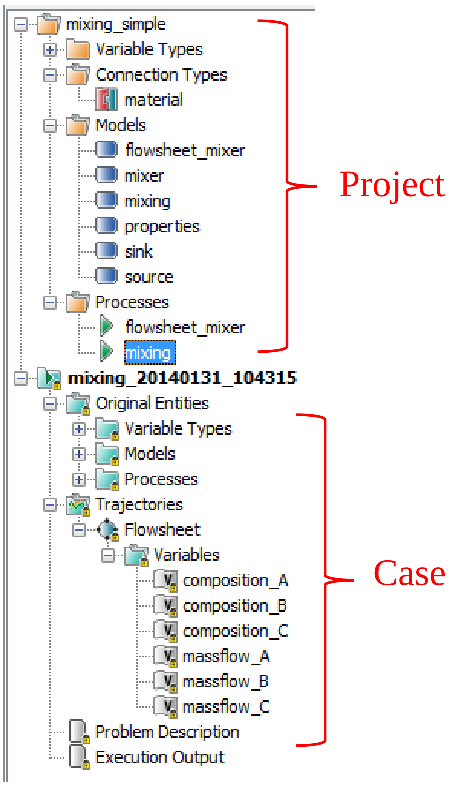
☑ Include results trajectory in Case
□ Send results trajectory to gRMS
Reporting interval: 10.0
Final results

☑ Include entities
☑ Include complete problem description

☑ Release license after execution
□ Retain execution directory
☑ Auto-update saved variable sets in source project

Simulation Results

The results are now available in the **Case** file that appears in the **Project Tree** area.

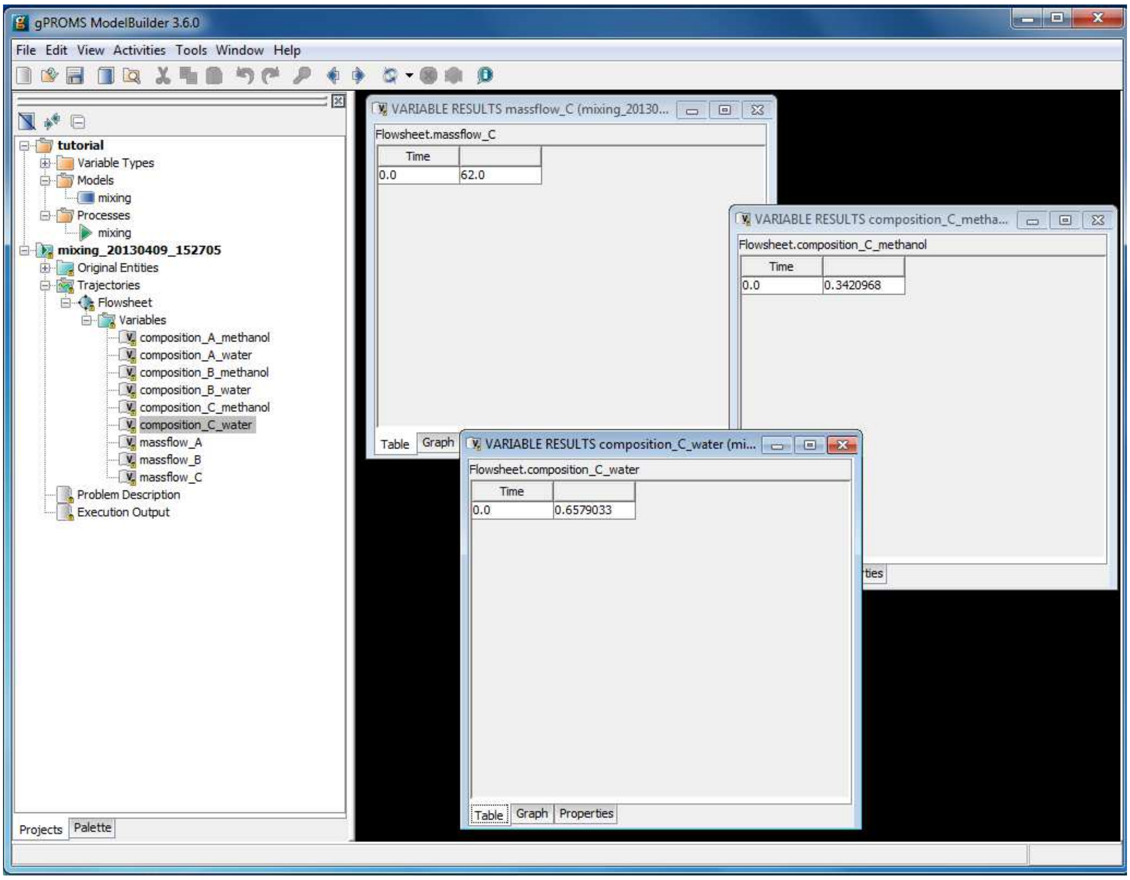


- mixing_simple
  - Variable Types
  - Connection Types
    - material
  - Models
    - flowsheet_mixer
    - mixer
    - mixing
    - properties
    - sink
    - source
  - Processes
    - flowsheet_mixer
    - mixing
- mixing_20140131_104315
  - Original Entities
    - Variable Types
    - Models
    - Processes
  - Trajectories
    - Flowsheet
      - Variables
        - composition_A
        - composition_B
        - composition_C
        - massflow_A
        - massflow_B
        - massflow_C
  - Problem Description
  - Execution Output

It contains the following informations:

- the **Original Entities** folder shows all elements declared in the project
- the **Trajectories** folder contains the declared and calculated Variables (double click on the one that should be shown)
- the Problem Description sheet shows the gPROMS input file plus the variable declarations and connection types
- the Execution Output inludes:
  - Output: solver and structural analysis report
  - Properties: document properties

For this tutorial the **Trajectories** folder is the only important one as it implies all calculation results!



Flowsheet.massflow_C

Time	
0.0	62.0

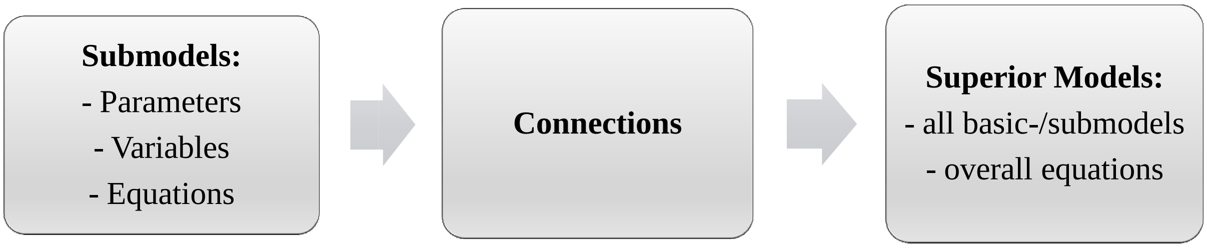
Flowsheet.composition_C_methanol

Time	
0.0	0.3420968

Flowsheet.composition_C_water

Time	
0.0	0.6579033

SUBMODELS AS BASE FOR SUPERIOR MODELS



Submodels:
- Parameters
- Variables
- Equations

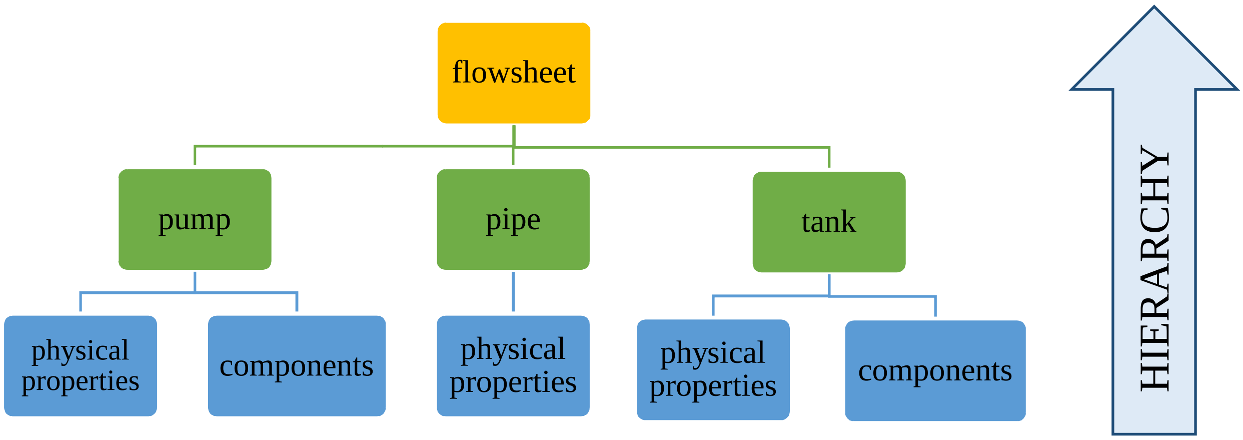
Connections

Superior Models:
- all basic-/submodels
- overall equations

What’s the advantage of using submodels?

- Complex models can be build up with several small and less extensive models → easier to write, read/analyse and especially for debugging
- Possibility to use a model several times: If the model is changed, all of them will adopt the modifications automatically (especially for flowsheeting)
- Endless usage of submodels in the hierarchical (de)composition.

Hierarchical structure (illustration)

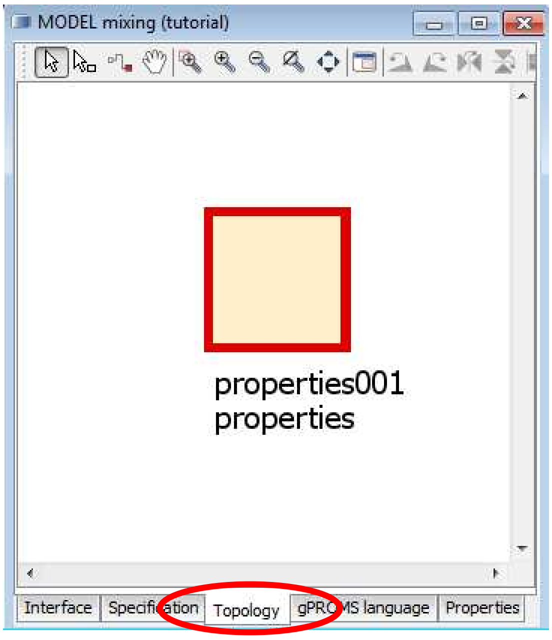


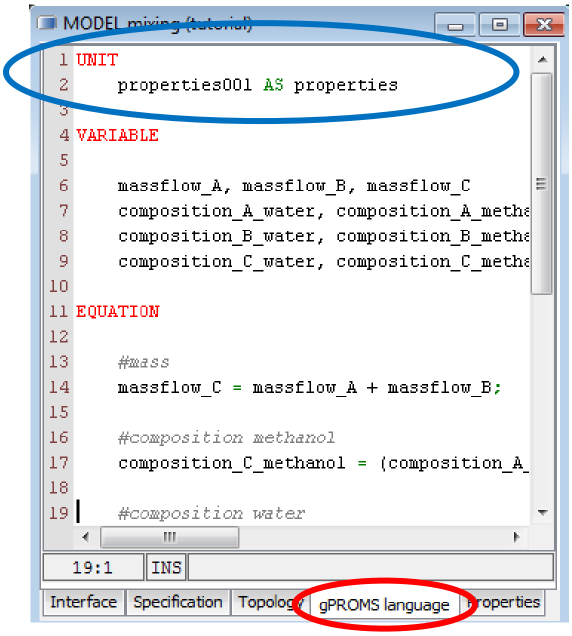
Create a submodel in gPROMS?

To simplify the “mixing” model, a submodel called “properties” will be implemented:

- Create a **new model** “properties”
- Activate the **Topology** tab in the “mixing” model
- Drag and drop the properties model into the mixing **Topology**

Now the “properties” unit appears in the “mixing” **Topology** and as well in the **gPROMS language**.





```
UNIT
    properties001 AS properties

VARIABLE

    massflow_A, massflow_B, massflow_C
    composition_A_water, composition_A_metha
    composition_B_water, composition_B_metha
    composition_C_water, composition_C_metha

EQUATION

    #mass
    massflow_C = massflow_A + massflow_B;

    #composition methanol
    composition_C_methanol = (composition_A_

    #composition water
```

In the **Topology** two names are shown:

- The upper one is the unit-name
    - Change it with double click to “properties” (it also changes in the **gPROMS language**)
    - It can be hidden by unselecting the N symbol in the toolbar
- The second one is the model’s name that was defined
    - It can be hidden by unselecting the M symbol in the toolbar

To reduce the number of intermediate variables in the “mixing” model the species will be declared as parameter in the submodel “properties”.

Parameter

As distinguished from variables parameters are time-invariant quantities and can’t be determined from the results of a calculation. Parameters have the advantage of reducing the number of variables in the system, but the disadvantage that they may never be treated as an unknown to be calculated. Parameters might be: species, their molecular weights, number of elements in an array dimension (expression involving other parameters), dimensions of a vessel…

The following Parameter types are possible:

- real, integer or logical values or
- real, integer or logical expressions
- foreign objects

Every declared parameter needs to be assigned in the SET section, either with values or appropriate expressions involving other parameters.

Syntax of the submodel

As a reminder what a model should contain:

The basic model is intended to contain the following issues:

- List and declaration of used Variables and Parameters - Variables must be allocated to Variable Types
- Parameter set
- Model equations
- Assignment of known variables

In the “properties” model only one parameter will be generated and set:

```
PARAMETER
species                          AS ORDERED_SET
SET
species := [ “water”, “methanol” ];
```

The ORDERED_SET is an array of indices!

Implementation in the superior model

To implement the submodel “properties” completely, the “mixing” model has to be adapted:

- Define the parameter as ORDERED_SET
- Change the variables, watch out that the compositions are changed to “AS ARRAY(species) OF composition”

- Set the parameter: In the SET-section, the assignment firstly contains the name of the submodel, where the parameter should be queried from, and then the name of the parameter:
  parameter = unitname(submodel).parametername
- Change the _methanol/_water expressions to (“water”)/ (“methanol”) in all equations as well as in the assignment.

```
PARAMETER
    species                                       AS ORDERED_SET

UNIT  # this term appears automatically with drag and drop the properties in the “mixing” topology
    properties AS properties

VARIABLE
massflow_A, massflow_B, massflow_C                AS massflow
composition_A, composition_B, composition_C  AS ARRAY(species) OF composition

SET
species := properties.species; # this term means that gPROMS should take the species from the
                                 unit properties- unitname.parametername
```

```
EQUATIONS
…
composition_C(“methanol”) = …..
…..

ASSIGN
…
composition_A(“methanol”) := 0.12;
…
```

By using the submodel it was possible to reduce the number of variables. Another advantage is that the composition results are now shown as array of both species.

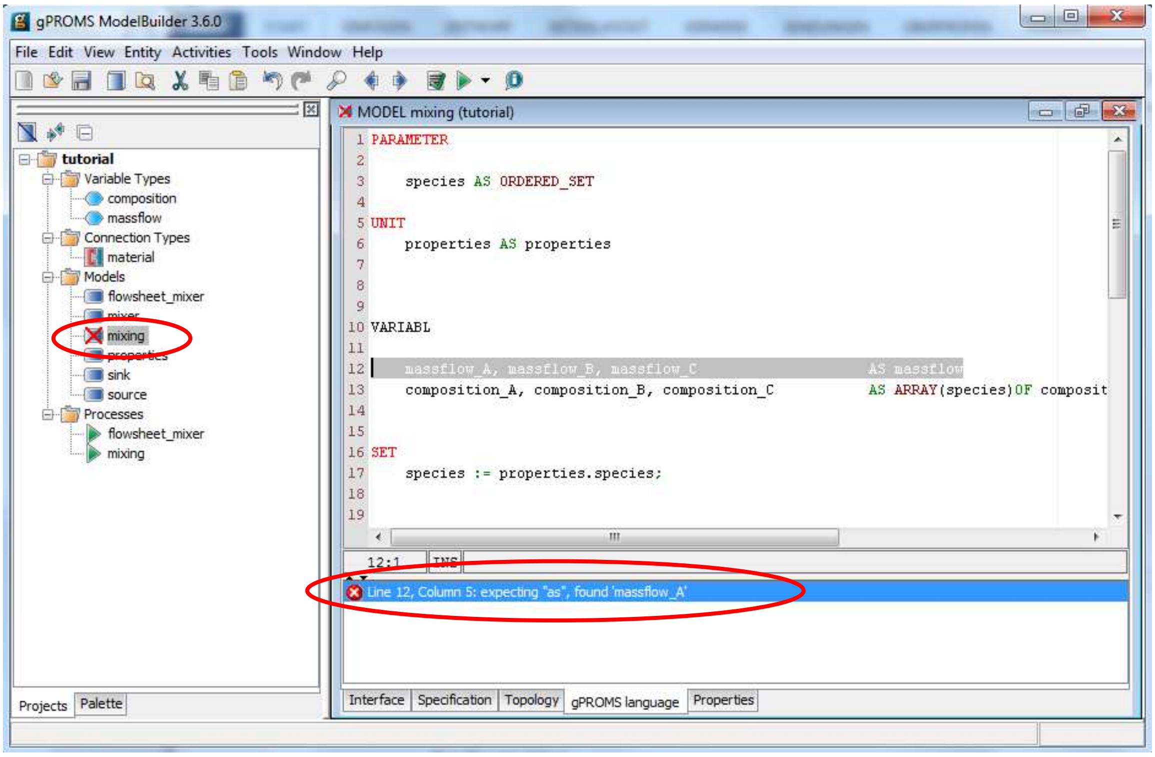
There is as well a possibility to reduce the number of equations (this procedures are very helpful for multicomponent calculations):

The two balances can be written as follows:

```
EQUATIONS
…
FOR i IN species DO
    composition_A(i) * massflow_A + composition_B(i) * massflow_B =
    massflow_C * composition_C(i);
END
```

If there is any error in the model it's displayed at the bottom of the worksheet and a red **cross** appears on the model symbol in the **project tree**.

To check if the syntax is right, click on the -icon in the **toolbar**.



```
PARAMETER

    species AS ORDERED_SET

UNIT
    properties AS properties



VARIABL

    massflow_A, massflow_B, massflow_C                 AS massflow
    composition_A, composition_B, composition_C        AS ARRAY(species)OF composit


SET
    species := properties.species;
```

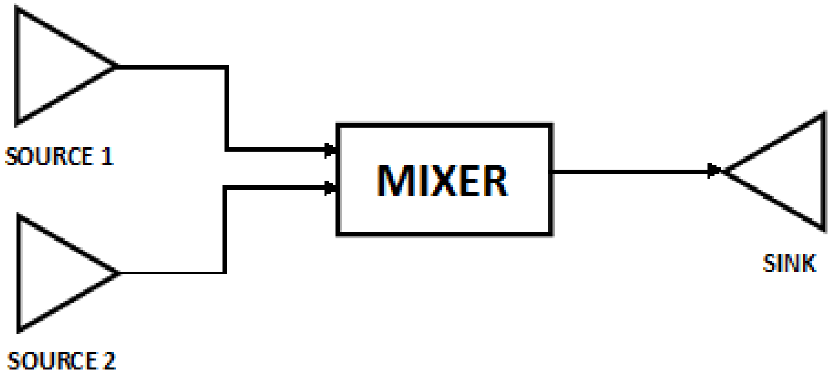
Line 12, Column 5: expecting "as", found 'massflow_A'

If there is no error indicated start the Simulation with the Run button . The Simulation Results can again be displayed in the result folder.

FLOWSHEET MIXER

In this section we will build up a simple flowsheet-model including three submodels:

- Source
- Mixer
- Sink



Connection Types:

For the construction of the mixer flowsheet connection types need to be declared. Connection types define the material / information that is transferred between the units.

Create a new Connection Type

- Right click on the **model** or select **Entity** on the **menu bar**
- Choose **New Entity… (Ctrl+E)**
- Select a name (for instance “material”)
- Pick **Connection Type** in the **dropdown menu** (Entity type)
- Click **OK**.

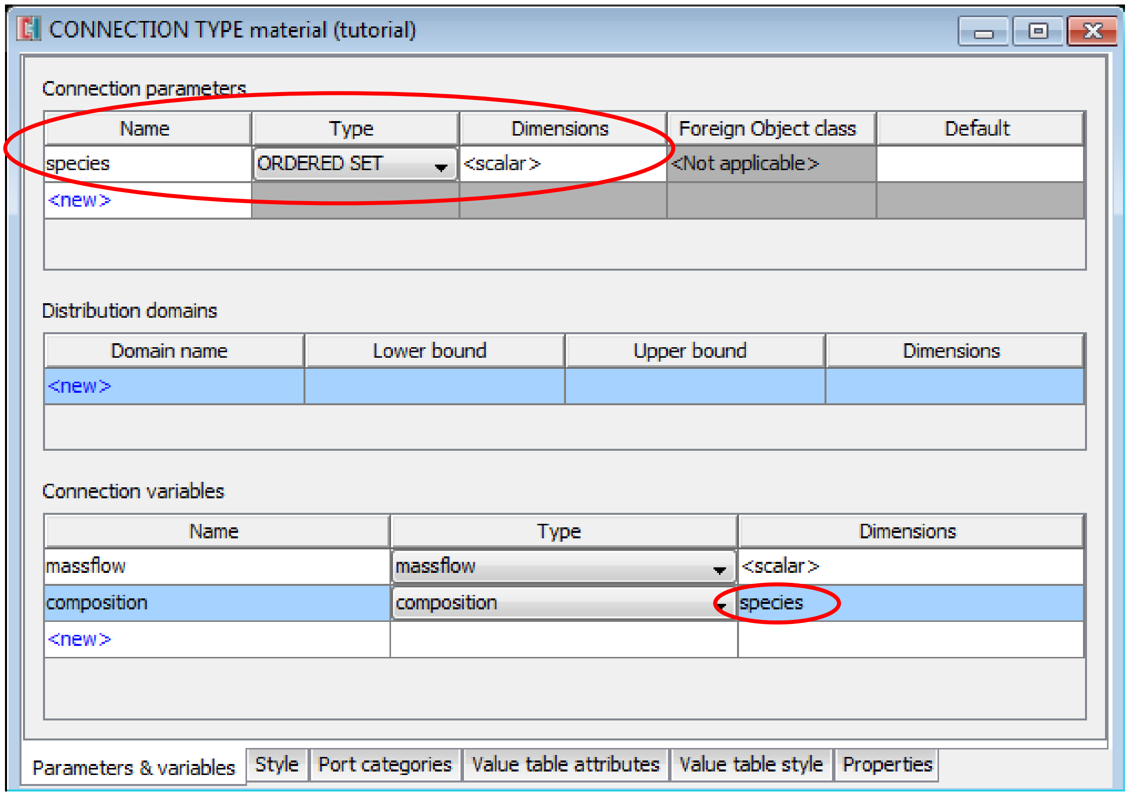
The **Connection Type** window includes the following tabs:

- Parameters & Variables: Tab to define parameters and variables (shown in the graphic) that are to be transported
- Style: gives the possibility to change the port and connection colours as well as the connection thickness

- Port Categories: In addition to transfer material it’s possible to transport signals or values (e.g. electrical signals) which can be defined here
- Value table attributes: implies those variables that are shown in the stream table report (has to be established separately)
- Value table style: for changing the graphics of the report
- Properties: with no importance for that tutorial

Define connection parameters and variables for the model:

- In the section **Connection parameters** type the name of the used parameter(s).
- The parameter’s **Type** of species is “ORDERED_SET” (the submodel properties will be implemented as well in that model)
- Leave the **Dimensions** as <scalar>



CONNECTION TYPE material (tutorial)

Connection parameters

Name	Type	Dimensions	Foreign Object class	Default
species	ORDERED SET	<scalar>	<Not applicable>	
<new>				

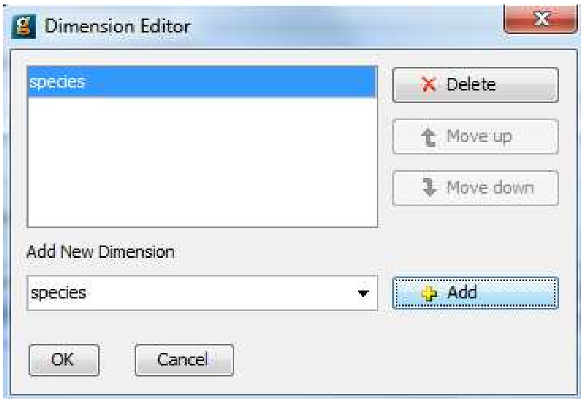
Distribution domains

Domain name	Lower bound	Upper bound	Dimensions
<new>			

Connection variables

Name	Type	Dimensions
massflow	massflow	<scalar>
composition	composition	species
<new>		

- Define the Variables and allocate them to their type.
- The mass flow **Dimension** is <scalar>, the one from the composition is an array of species.
- To set “species” in the **Dimensions box**, click on the **dimensions field**.
- Click **Add** in the **Dimensions Editor** and **OK**. “Species” will appear in the box.



- Close the ConnectionType window with the close icon ☒. The changes will be saved automatically.

Units for “flowsheet mixer”

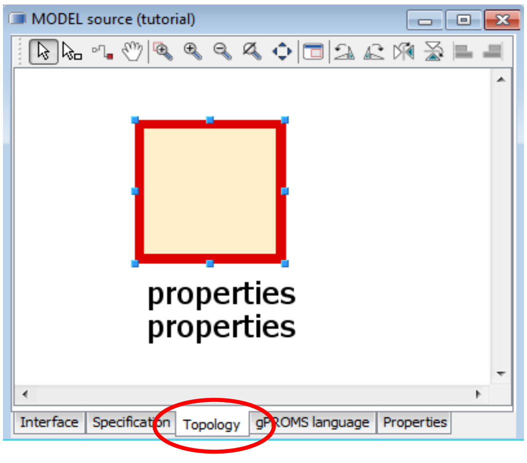
After defining the Connection Types the single units needed for the flowsheet can be generated.

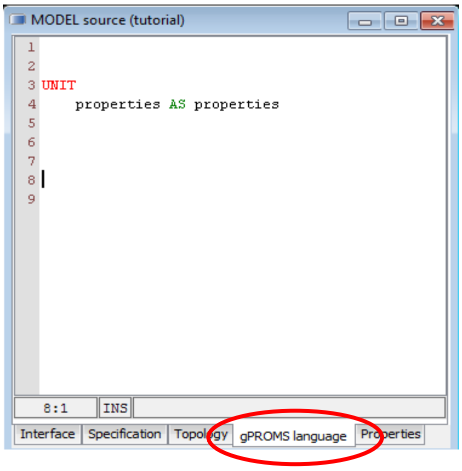
Source

The source is a unit that defines an input-stream to the flowsheet. The output of the source will be the input of the next unit. Sources can only have one port, therefore every input stream requires a separate source unit.

Create the source model

- Create a new model “source”
- Drag and drop the “properties” model in the “source” **Topology**
- Rename “properties001” to “properties”
- The submodel is now visible in the graphical surface of the **Topology** and as well as syntax in the **gPROMS language** (UNIT)

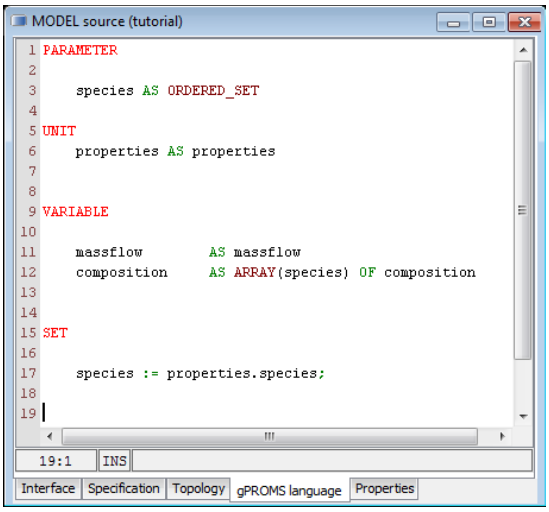




```
UNIT
    properties AS properties
```

Variable and parameter declaration

- Declare and set the parameter(s)
- List the variables (allocated to the right variable type)



```
PARAMETER

    species AS ORDERED_SET

UNIT
    properties AS properties


VARIABLE

    massflow        AS massflow
    composition     AS ARRAY(species) OF composition


SET

    species := properties.species;

```

The model Interface

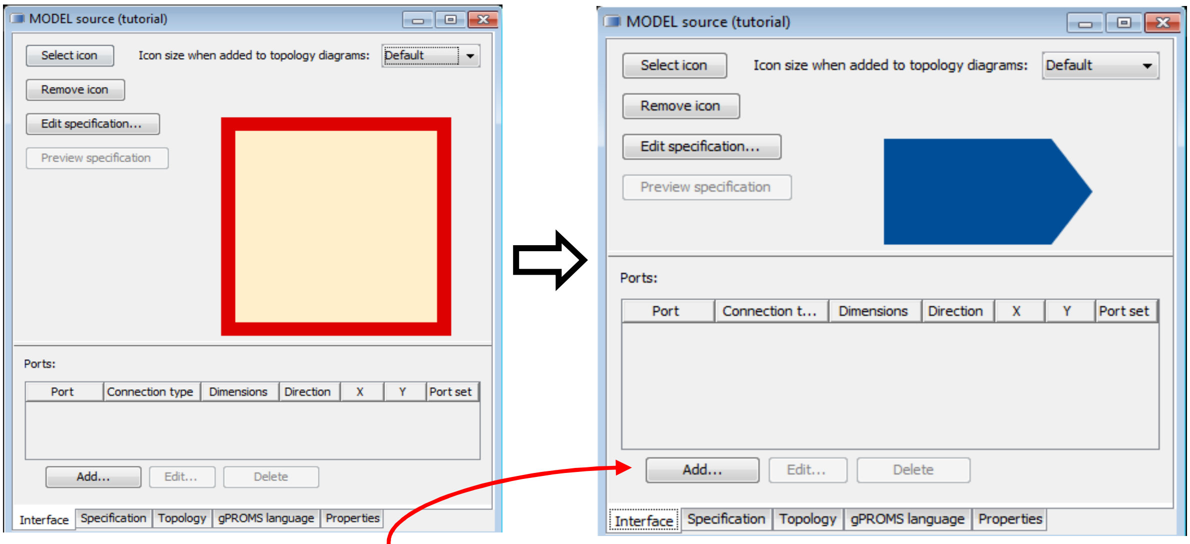
The model Interface can be used to adapt specifications, add connection ports and change the graphical appearance (icon) of the Model in the **Topology**.

Select icon:

- Click on the **Select icon** button
- Select an image in .png format:
    - Can be created on your own (e.g. in paint)
    - Or choose one of the **PML library**:

      C:\Program Files (x86)\PSE\ModelBuilder_3.n.x.x\libraries\PML\icons

If you click on the **Remove icon** button, the old picture appears again.

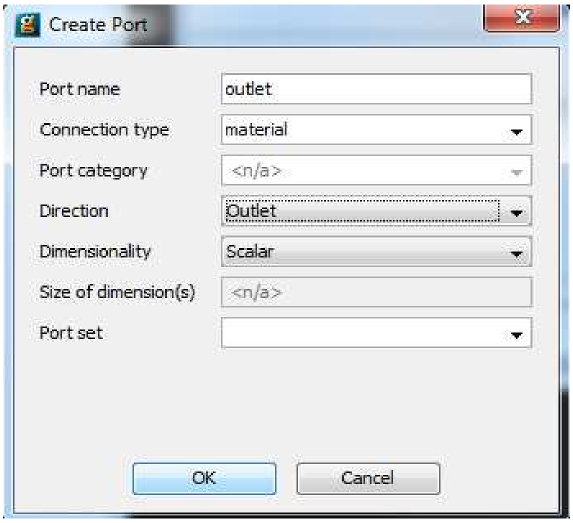


Port	Connection type	Dimensions	Direction	X	Y	Port set

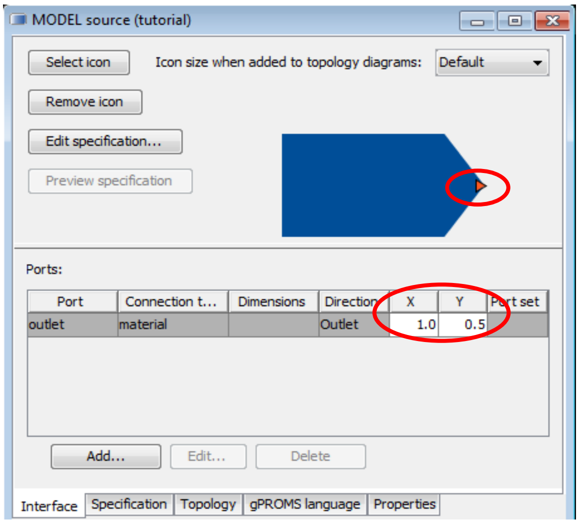
Add a port

To add a port it’s important to always define the Connection Types first!

- Click on the **Add…** button in the model **Interface**- the **Create Port** window appears
- Choose a **Port name** (e.g. outlet)
- Select “material” as **Connection type**
- Leave the **Port category** field blank
- As it is a source, choose **Outlet** in the **Direction dropdown menu**
- Take **Scalar** in **Dimensionality**
- Click **OK**



Port name	outlet
Connection type	material
Port category	<n/a>
Direction	Outlet
Dimensionality	Scalar
Size of dimension(s)	<n/a>
Port set	

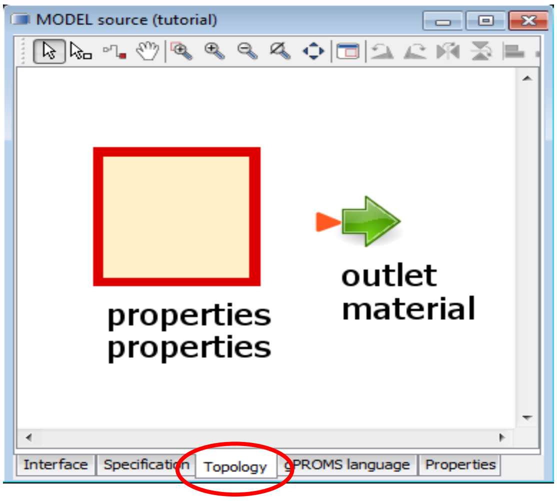


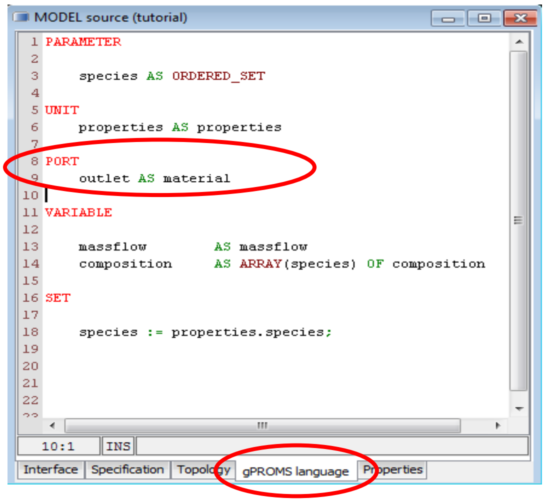
Port	Connection t...	Dimensions	Direction	X	Y	Port set
outlet	material		Outlet	1.0	0.5	

To position the port:

- Type the coordinates in the **x/y field**
- Or drag the port with the mouse to the prefered position

After adding a port, it appears in both the T**opology** and the **gPROMS language**:




MODEL source (tutorial)

```
PARAMETER

    species AS ORDERED_SET

UNIT
    properties AS properties

PORT
    outlet AS material

VARIABLE

    massflow        AS massflow
    composition     AS ARRAY(species) OF composition

SET

    species := properties.species;
```

10:1 INS

To delete a port, select the one that should be removed in the **Interface** and click on the **Delete button**.

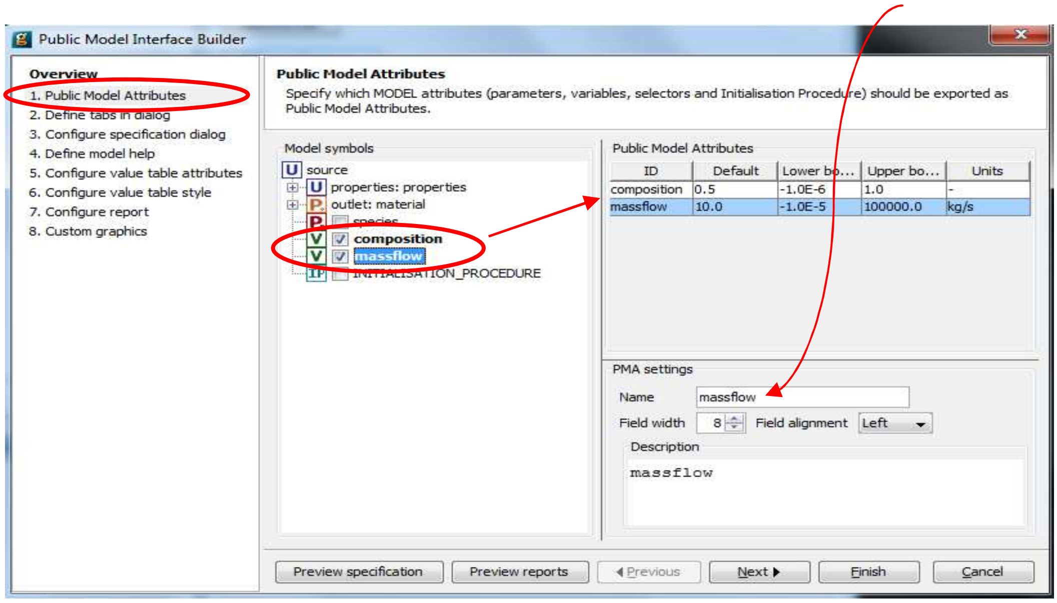
Edit Specification

To define known variables there are two possibilities:

- the assignment in the **gPROMS language** or
- within the **Specification** window.

The assignment in the **gPROMS language** was already carried out in the “simple mixing” model. In this section we choose the alternative way:

- click on the **Edit Specification…** button in the model **Interface**
- select composition (V=Variable) and massflow in the **Public Model Attributes**
- the two variables are shown in the **Public Model Attributes area**
- to display an easier/another name in the specification edit it in the **PMA settings.**


Public Model Interface Builder

Overview
1. Public Model Attributes
2. Define tabs in dialog
3. Configure specification dialog
4. Define model help
5. Configure value table attributes
6. Configure value table style
7. Configure report
8. Custom graphics

Public Model Attributes

Specify which MODEL attributes (parameters, variables, selectors and Initialisation Procedure) should be exported as Public Model Attributes.

Model symbols
- U source
  - U properties: properties
  - P outlet: material
  - P □ species
  - V ☑ composition
  - V ☑ massflow
  - IP □ INITIALISATION_PROCEDURE

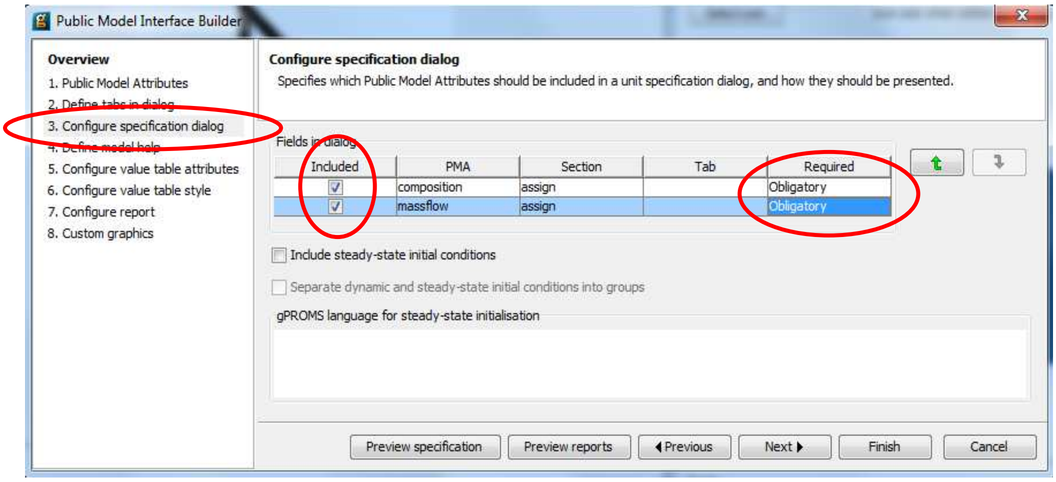
Public Model Attributes

ID	Default	Lower bo…	Upper bo…	Units
composition	0.5	-1.0E-6	1.0	-
massflow	10.0	-1.0E-5	100000.0	kg/s

PMA settings

Name: massflow
Field width: 8
Field alignment: Left
Description: massflow

- Go to point **3. Configure specification dialog** (with the **Next** button or just click on it in the **Overview**)
- Tick **Included** for both variables
- And set them as “Obligatory” in the **Required** field (means that they must to be declared afterwards)
- Click **Finish**



Public Model Interface Builder

Overview
1. Public Model Attributes
2. Define tabs in dialog
3. Configure specification dialog
4. Define model help
5. Configure value table attributes
6. Configure value table style
7. Configure report
8. Custom graphics

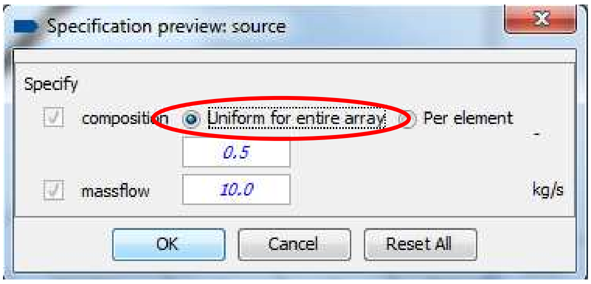
Configure specification dialog
Specifies which Public Model Attributes should be included in a unit specification dialog, and how they should be presented.

Fields in dialog

Included	PMA	Section	Tab	Required
☑	composition	assign		Obligatory
☑	massflow	assign		Obligatory

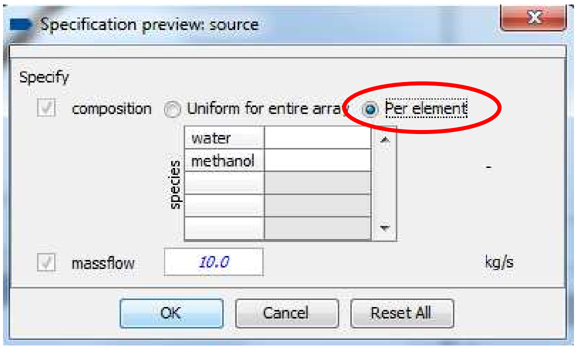
□ Include steady-state initial conditions
□ Separate dynamic and steady-state initial conditions into groups
gPROMS language for steady-state initialisation

Click on the **Preview Specification** button to see the **specification window**:



Specification preview: source

Specify
☑ composition ● Uniform for entire array ○ Per element 0.5 -
☑ massflow 10.0 kg/s



Specification preview: source

Specify
☑ composition ○ Uniform for entire array ● Per element -

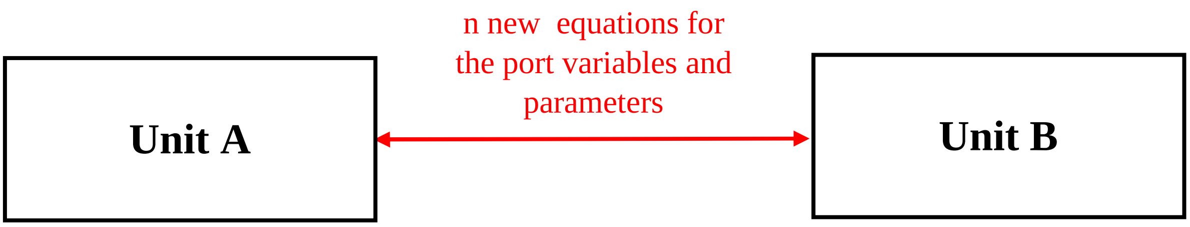
species	
water	
methanol	

☑ massflow 10.0 kg/s

EQUATIONS

The defined ports are the starting point for the transport of variables and parameters within the connections. As an attached unit doesn’t know the values of the previous unit, those variables and parameters need to be declared in either the starting unit (as outlet) or the final unit (as inlet). This means n equations for all exchanged values:

- Parameters are defined within SET
- Variables are declared within EQUATIONS



Therefore it is required to define the outlet in the “source” model. Complete syntax in the **gPROMS language**:

```
SET
…
outlet.species := species;      #or outlet.species := properties.species


EQUATION
outlet.massflow = massflow;
outlet.composition = composition;
```

Mixer

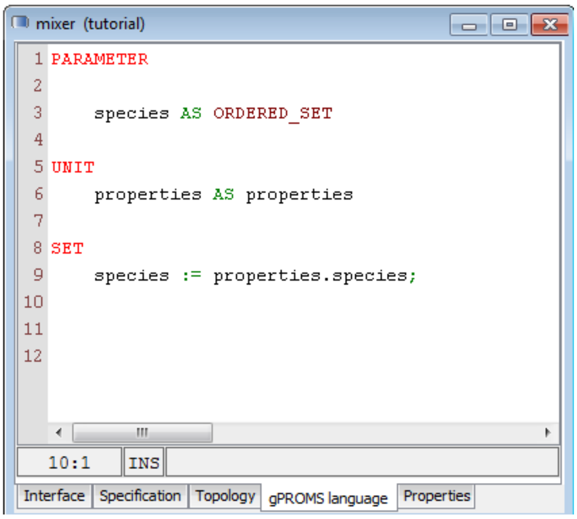
Create the mixer model

- Create a new model “mixer”
- Drag and drop the “properties” model in the “mixer” **Topology**
- Rename “properties001” to “properties”
- The submodel is now visible in the graphical surface **Topology** and as well as a syntax in the **gPROMS language** (UNIT)

Variable and Parameter declaration

The variables are already defined in the source model, which means that they don’t need to be declared again. But the submodel needs to be integrated again:

- Declare and set the parameter



```
PARAMETER

    species AS ORDERED_SET

UNIT
    properties AS properties

SET
    species := properties.species;
```

The model Interface

Select icon the same way as in the “source” model

Add ports

The mixer needs at least two input and one output port. Either we define two input ports, or we keep the input port variable by selecting a dynamic port (gives the opportunity to connect as many input-streams as wanted/required).

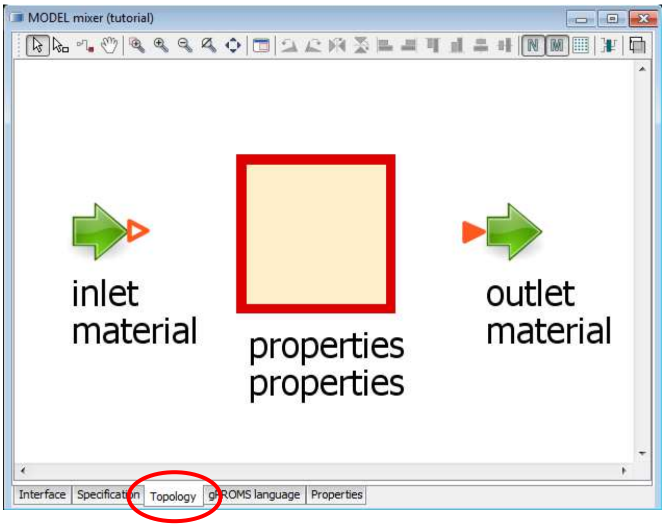
Inlet Ports (dynamic):

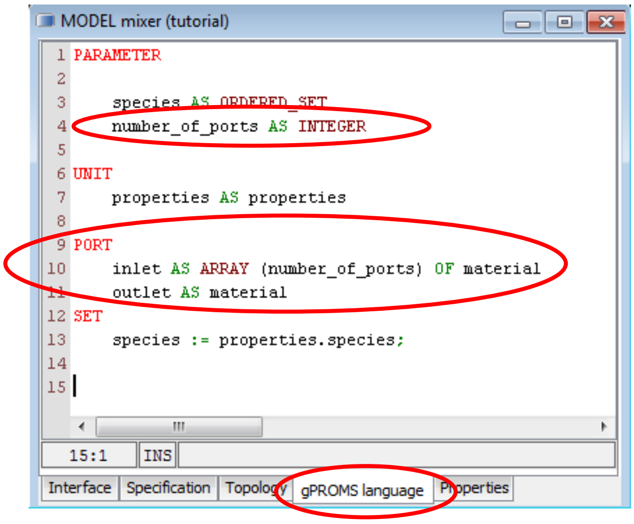
- Click on the **Add…** button in the model **interface-** the **Create Port** window appears
- Choose a **Port name** (e.g. inlet)
- Select “material” as **Connection type**
- Leave the **Port category** field blank
- Choose **Inlet** in the **Direction dropdown menu**
- Take **Dynamic size** in **Dimensionality**
- Type “number_of_ports” in the **Bound size parameter** field
- Click **OK**

Outlet Port (Scalar):

- Create one outlet port as Scalar (not dynamic).

As an illustration again the change in the **Topology** and the **gPROMS language**:





```
PARAMETER

    species AS ORDERED_SET
    number_of_ports AS INTEGER

UNIT
    properties AS properties

PORT
    inlet AS ARRAY (number_of_ports) OF material
    outlet AS material
SET
    species := properties.species;

```

EQUATIONS

The last thing required in the model “mixer” is to define the equations (component balance) as well as the output stream.

Complete the syntax in the **gPROMS language**:

```
SET
…
outlet.species := species;


EQUATION
outlet.massflow = SIGMA(inlet().massflow);


outlet.composition("water") = SIGMA(inlet().massflow * inlet().composition("water")) /
SIGMA(inlet().massflow);


outlet.composition("methanol") = SIGMA(inlet().massflow *
inlet().composition("methanol")) / SIGMA(inlet().massflow);
```

Syntax Remarks:

SIGMA-function: The sum of all elements of all arguments inlet().massflow*inlet().composition(“methanol”):

- multiplication of the massflow of inlet 1 and the methanol composition of inlet 1
- multiplication of the massflow of inlet 2 and the methanol composition of inlet 2

➔ the result is an array; with the SIGMA function it gets a scalar

If the bracket term is empty “()”, all inlet mass flows are included in the SIGMA calculations.

Another example shows how the SIGMA function considers only a part of an array. Therefore the properties model is extended in the following way:

```
PARAMETER
    species         AS ORDERED_SET
    sub,sub1        AS ORDERED_SET
SET
    species :=      ["water","methanol"];
    sub :=          ["water"];
    sub1 :=         ["methanol"];
```

The new parameters have to be implemented in the mixer-model:

```
PARAMETER
    species                 AS ORDERED_SET
    number_of_ports         AS INTEGER
    sub, sub1               AS ORDERED_SET
.
.
SET
    species :=      properties.species;
    sub :=          properties.sub;
    sub1 :=         properties.sub1;
```

There is the possibility to use all species in the SIGMA function with:

```
EQUATION
FOR i IN species DO
outlet.composition(i) = SIGMA(inlet().massflow * inlet().composition(i)) /
SIGMA(inlet().massflow); END
```

To consider defined parts of the species-array:

```
#for the calculation of the water composition
FOR i IN sub DO
outlet.composition(i) = SIGMA(inlet().massflow * inlet().composition(i)) /
SIGMA(inlet().massflow); END

#for the calculation of the methanol composition
FOR i IN sub1 DO
outlet.composition(i) = SIGMA(inlet().massflow * inlet().composition(i)) /
SIGMA(inlet().massflow); END
```

In that case the SIGMA- function will only consider the defined subcomponents “water” or “methanol”. If the species contain three or more components, for instance [“CO”, “CO2”, “N2”] the SIGMA-function could be calculated for only two (or a few) of them by defining one sub-array and using that in the calculations. E.g.: sub := [“CO2”, “N2”].

Sink

The sink is a unit that defines an output stream of the flowsheet. Sinks can only have one input port. Therefore every output stream requires a separate sink unit.

Create a new model “sink”

- Create a new model “sink”
- Drag and drop the “properties” model in the “sink” **topology**
- Rename “properties001” to “properties”
- Declare and set the parameter the same way as in the models before

The model Interface

Select icon the same way as in the “source” and “mixer” model

Add a port

Inlet Port (scalar):

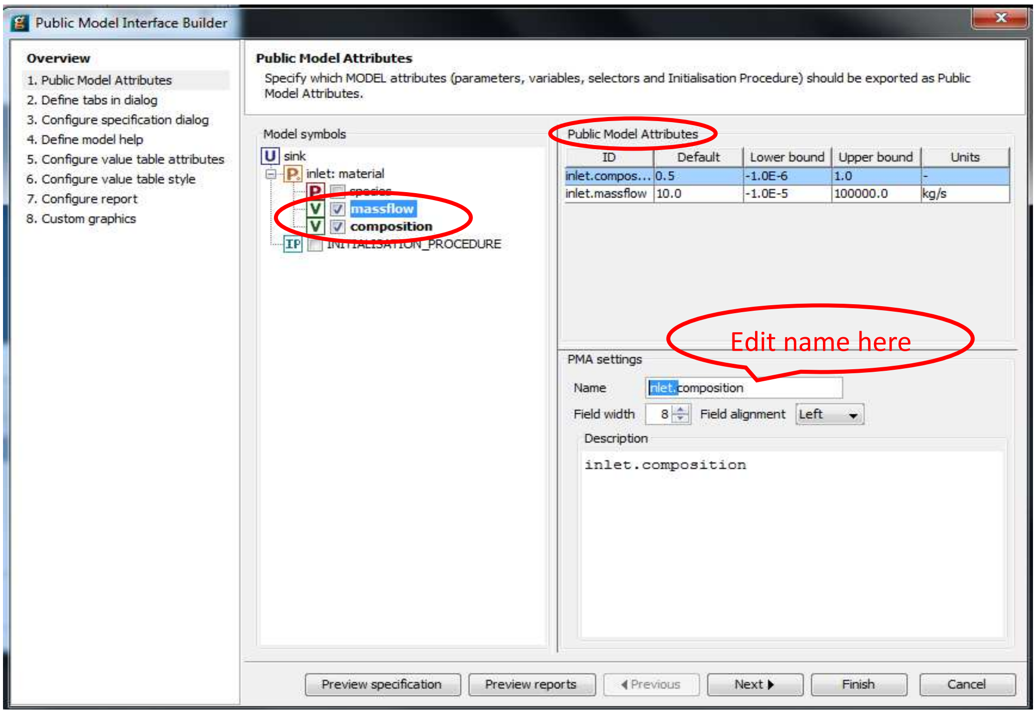
- Click on the **Add…** button in the model **interface-** the **Create Port** window appears
- Choose a **Port name** (e.g. inlet)
- Select “material” as **Connection type**
- Leave the **Port category** field blank
- As it is a sink, choose **Inlet** in the **Direction dropdown menu**
- Take **Scalar** in **Dimensionality**
- Click **OK**

Edit Report

The specification is not only a tool to assign variables. It’s as well possible to create reports. As this unit is a sink, we don’t need to declare variables, but the goal is that the simulation results are shown directly in the **process flowsheet** after double-click on the “sink” model.

To create **Report tables**:

- click on the **Edit Specification…** button in the model **Interface**
- select composition (V=Variable) and massflow in the **Public Model Attributes (inlet: material)**
- the two variables are shown in the **Public Model Attributes area**
- If you want to display an easier/another name in the specification edit it in the **PMA settings.**



Public Model Interface Builder

Overview

1. Public Model Attributes
2. Define tabs in dialog
3. Configure specification dialog
4. Define model help
5. Configure value table attributes
6. Configure value table style
7. Configure report
8. Custom graphics

Public Model Attributes

Specify which MODEL attributes (parameters, variables, selectors and Initialisation Procedure) should be exported as Public Model Attributes.

Model symbols

- U sink
  - P inlet: material
    - P □ species
    - V ☑ massflow
    - V ☑ composition
  - IP □ INITIALISATION_PROCEDURE

Public Model Attributes

ID	Default	Lower bound	Upper bound	Units
inlet.compos…	0.5	-1.0E-6	1.0	-
inlet.massflow	10.0	-1.0E-5	100000.0	kg/s

Edit name here

PMA settings

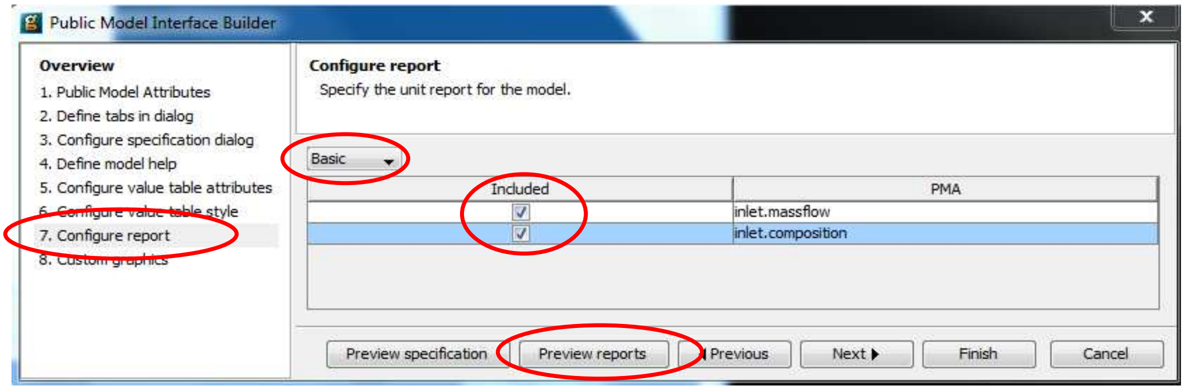
Name: inlet.composition

Field width: 8 Field alignment: Left

Description

inlet.composition

- Go to point **7.Configure report** (with the **Next** button or just click on it in the **Overview**)
- Select **Basic** in the **dropdown menu**
- Tick **Included** for both variables
- To see the preview click on the **Preview reports** button
- Click **Finish**


Public Model Interface Builder
Overview
1. Public Model Attributes
2. Define tabs in dialog
3. Configure specification dialog
4. Define model help
5. Configure value table attributes
6. Configure value table style
7. Configure report
8. Custom graphics
Preview specification
Preview reports
Previous
Next
Finish
Cancel

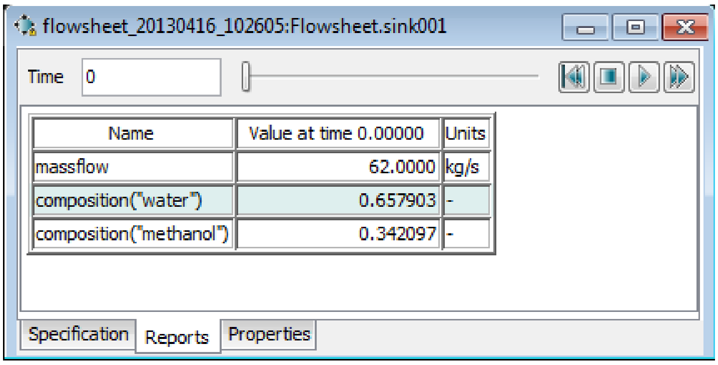
Configure report

Specify the unit report for the model.

Basic

Included	PMA
☑	inlet.massflow
☑	inlet.composition

With double-click on the “sink” model in the **process flowsheet** the report window appears:



Time 0

Name	Value at time 0.00000	Units
massflow	62.0000	kg/s
composition("water")	0.657903	-
composition("methanol")	0.342097	-

Flowsheeting

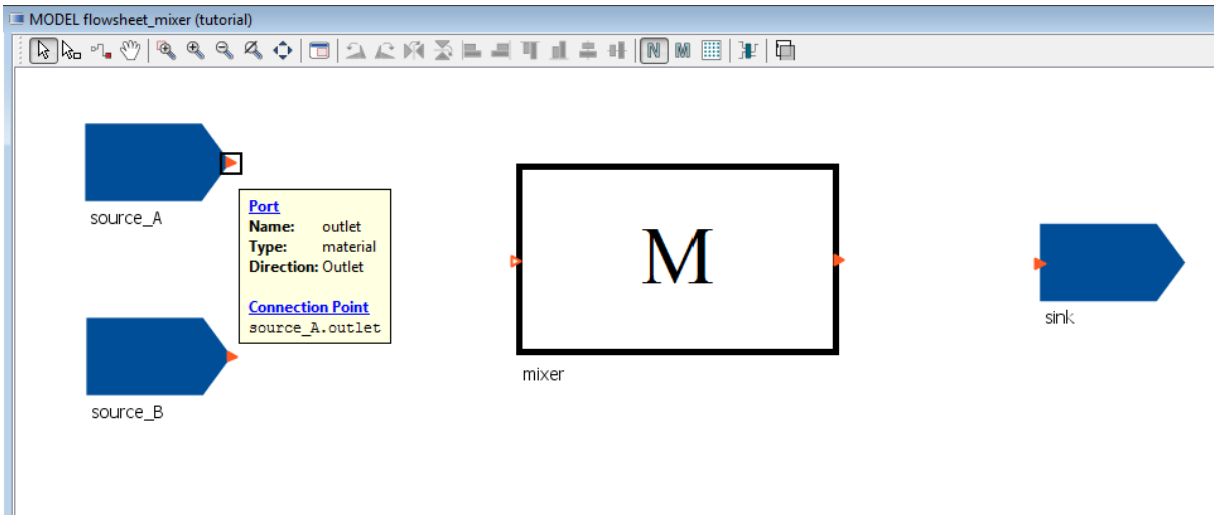
As all basic units are prepared the next step is to connect them in a superior “flowsheet_mixer” model.

- Create a new model “flowsheet_mixer”
- Drag and drop the “source”, “mixer” and “sink” models into the **Topology**
- Click on the “source” and copy it once **(Ctrl+c -> Ctrl+v)**
- If wanted, rename the units with double click on the name

Connect the units:

To connect the single units:

- Click on a port (the connection line appears)
- Connect the outlet with the inlet port



MODEL flowsheet_mixer (tutorial)

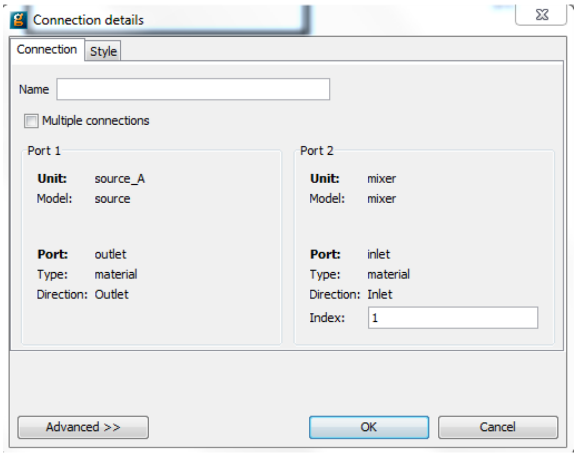
Port
Name: outlet
Type: material
Direction: Outlet

Connection Point
source_A.outlet

The Connection Details window:

After connecting two basic units, the **Connection Details** window appears:

- **Connection** tab: The **Connection** tab shows which units are connected. It's the section to name the connection and to choose the inlet index.
- **Style** tab: In the **Style** tab the visual representation of the connections in the **Topology** can be modified. Use this tab for changing the thickness and colour of your connections.



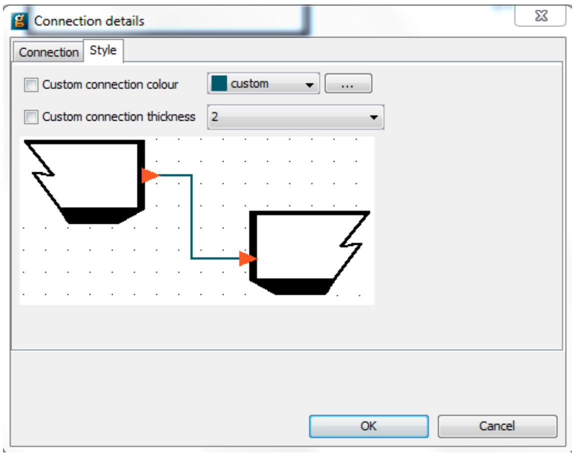
Connection details

Name:

□ Multiple connections

Port 1
Unit: source_A
Model: source
Port: outlet
Type: material
Direction: Outlet

Port 2
Unit: mixer
Model: mixer
Port: inlet
Type: material
Direction: Inlet
Index: 1



Connection details

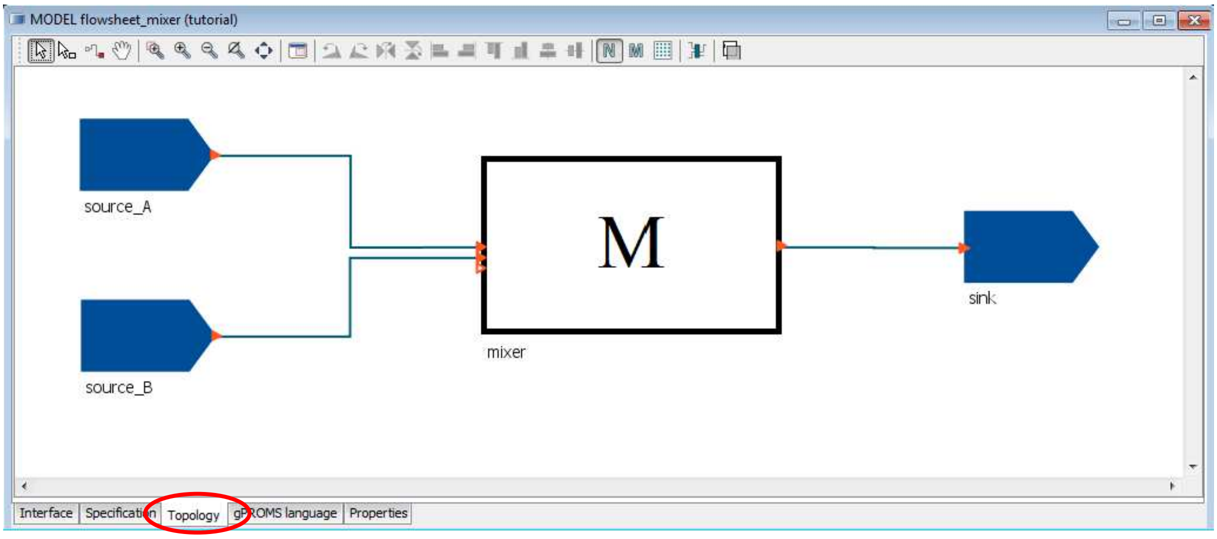
□ Custom connection colour: custom

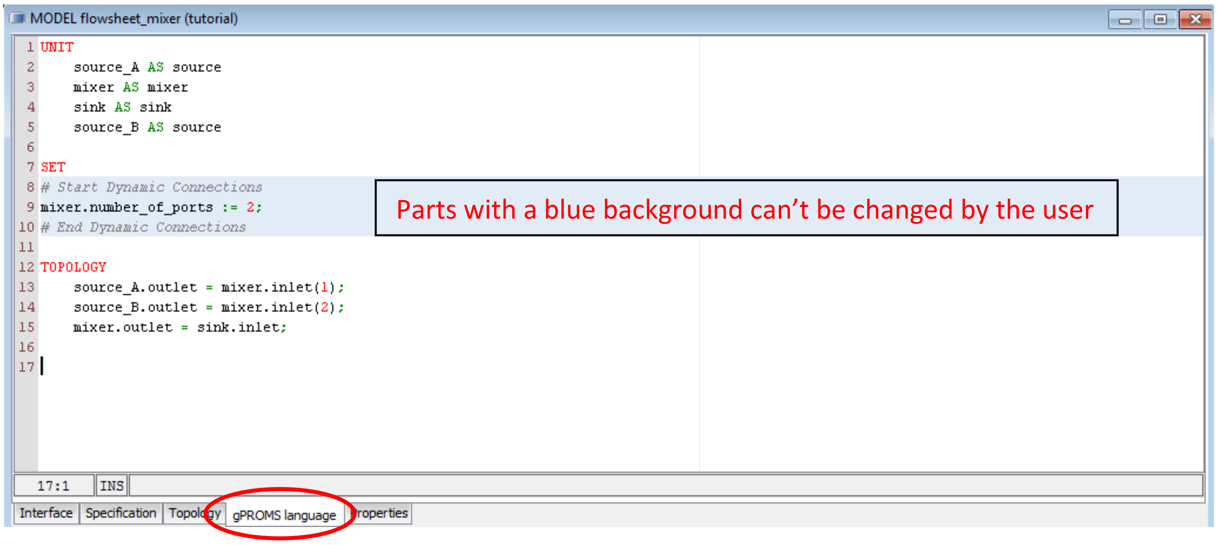
□ Custom connection thickness: 2

If desired the name and index can be changed, but it's not necessary. Just click **OK** and connect the other units as well.

Topology and gPROMS language:

The complete “flowsheet_mixer” looks as follows:





```
UNIT
    source_A AS source
    mixer AS mixer
    sink AS sink
    source_B AS source

SET
# Start Dynamic Connections
mixer.number_of_ports := 2;
# End Dynamic Connections

TOPOLOGY
    source_A.outlet = mixer.inlet(1);
    source_B.outlet = mixer.inlet(2);
    mixer.outlet = sink.inlet;

```

Parts with a blue background can’t be changed by the user

In the **gPROMS language** description:

- UNIT: List of all units used in the flowsheet
- SET: Shows the number of ports currently connected at the dynamic port
- TOPOLOGY: Describes the connections between the units

Everything that appears in the **gPROMS language** is carried out automatically, while adding or changing something in the **Topology**.

Specify the Input:

Double click on the “source” models in the **Topology** and define the input values (use either the ones from the “simple_mixing” or just random numbers).

Edit PROCESS

Convert the model into a process:

- Right-click on the **model** “flowsheet_mixer”
- Choose **Edit PROCESS**

The PROCESS appears as separate window and as well in the **Project tree**.

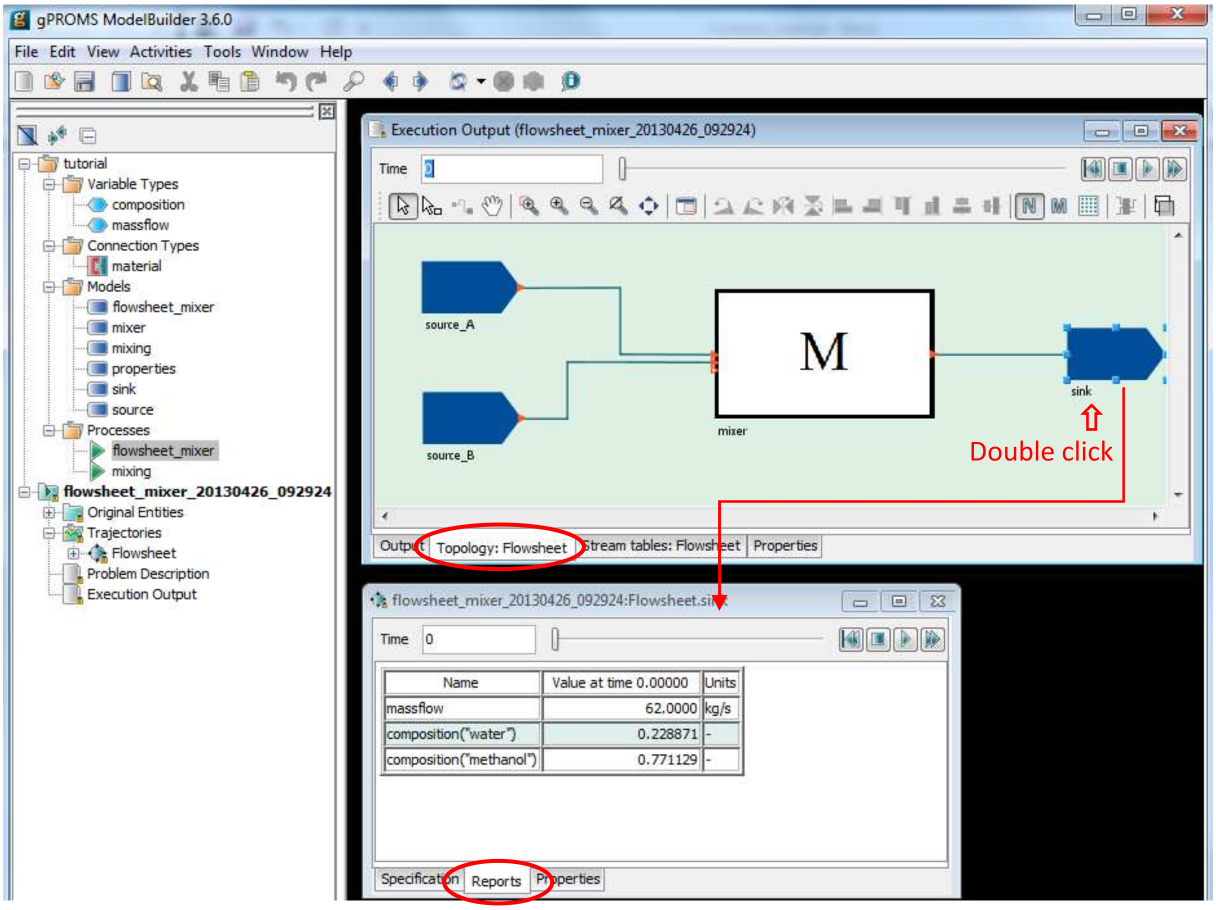
Run the Simulation

To run the simulation click on the **play-**icon ▶ in the **Toolbar** or right-click on the **Process** (“flowsheet_mixer”) and choose **Simulate...**

The **Simulate window** appears:

- click **OK** to start the simulation

Simulation Results



Double click

Name	Value at time 0.00000	Units
massflow	62.0000	kg/s
composition("water")	0.228871	-
composition("methanol")	0.771129	-

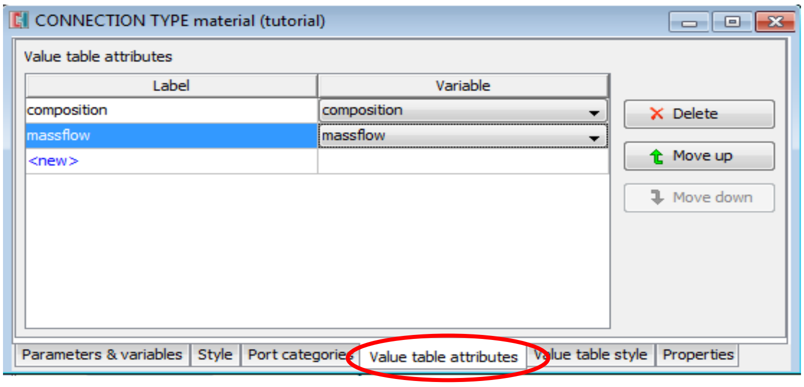
As the model is a flowsheet model now, the **Execution Output** window consists two more tabs (beside the **Output** and **Properties**):

- **Topology:** Shows our model
- **Stream tables:** Shows the values of the streams (empty in this case- how to create stream tables is described in the next step)

Create stream table

To create a stream table go to **Connection types** and double click on **material**.

- Choose the **Value table attributes** tab
- Type a name in **Label**: e.g. composition
- Select the right type in the **Variable** dropdown menu
- **Close** the window



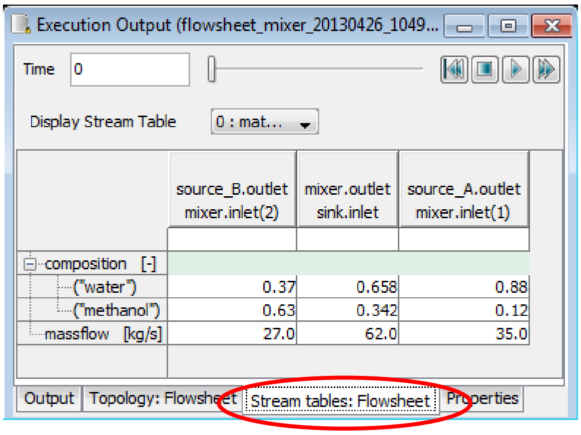
CONNECTION TYPE material (tutorial)

Value table attributes

Label	Variable
composition	composition
massflow	massflow
<new>	

The stream table is visible after running the simulation again.

- By selecting the **tab Stream tables: Flowsheet**



Execution Output (flowsheet_mixer_20130426_1049...

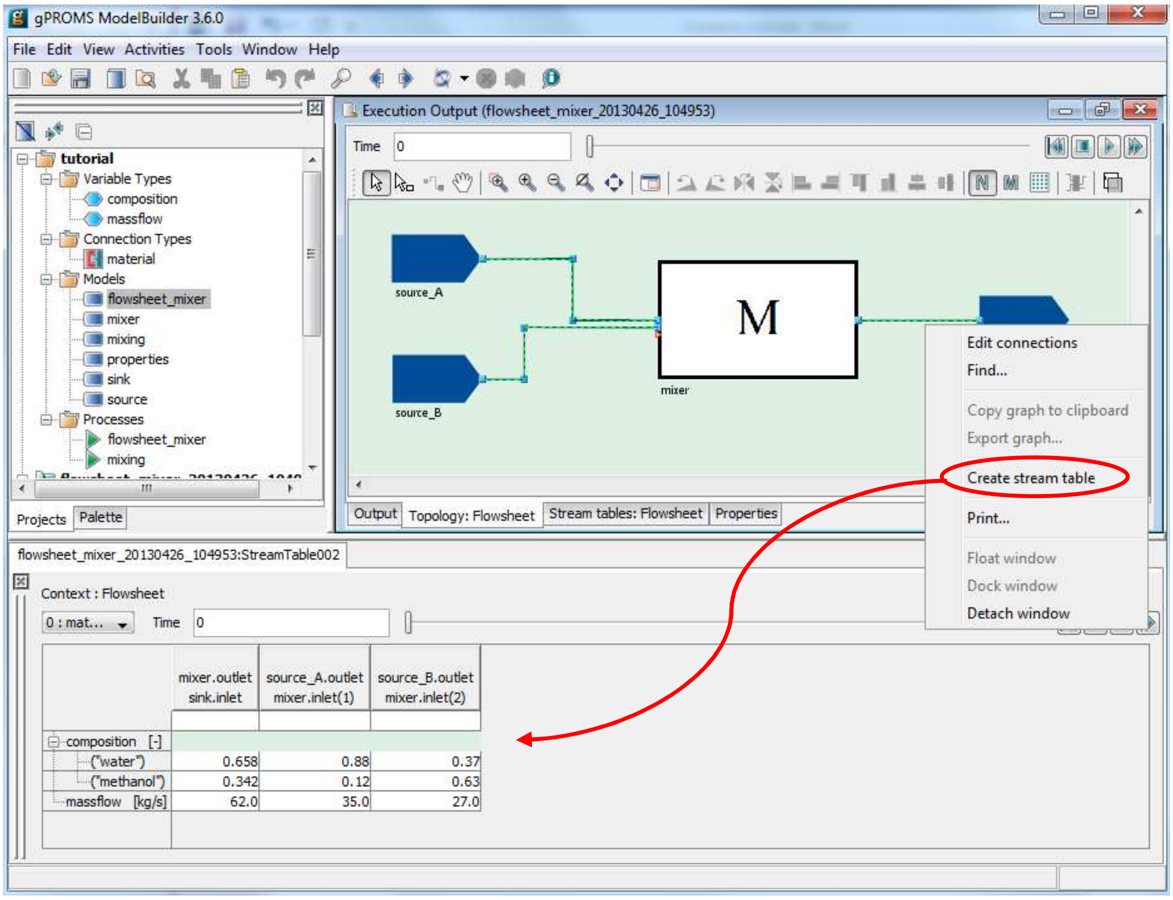
Time 0

Display Stream Table 0 : mat...

	source_B.outlet mixer.inlet(2)	mixer.outlet sink.inlet	source_A.outlet mixer.inlet(1)
composition [-]			
("water")	0.37	0.658	0.88
("methanol")	0.63	0.342	0.12
massflow [kg/s]	27.0	62.0	35.0

- Or keep **Ctrl** held and **left-click** one by one on the streams that are visible in the **Execution Output Topology: Flowsheet**
- Right click and select **Create Stream table**

The **Stream Table** will be displayed in the lower part of the **gRROMS** window.



	mixer.outlet sink.inlet	source_A.outlet mixer.inlet(1)	source_B.outlet mixer.inlet(2)
composition [-]			
("water")	0.658	0.88	0.37
("methanol")	0.342	0.12	0.63
massflow [kg/s]	62.0	35.0	27.0

Simplify the balance

As before in the “mixing” model it is possible to write the balances in only one equation. Try to find the solution with the “FOR i IN species“- expression on your own.

Schedules

“Schedules are used in gPROMS to define operating procedures. An operating procedure can be considered as a recipe that defines periods of undisturbed operation *along with* specified or conditional external disturbances to the system.” (gPROMS Documentation)

Schedules are part of the process. Schedules allow the user to program simple events such as changing one variable until a special event is reached.

gPROMS has a subdivision in Elementary- and Composite Tasks.

Elementary Tasks: can be used to specify external disturbances to the system (e.g. changing the values of the input variables, specify periods of changes…). In this tutorial only three of them will be described and applied:

- Reassign
- Replace
- Continue

Composite Tasks: those timing statements describe "when/how" do something.-How to implement elementary tasks. Two of them are:

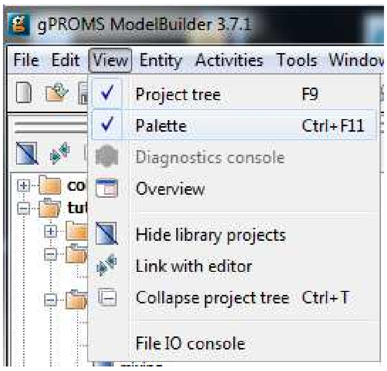
- Sequence
- Parallel

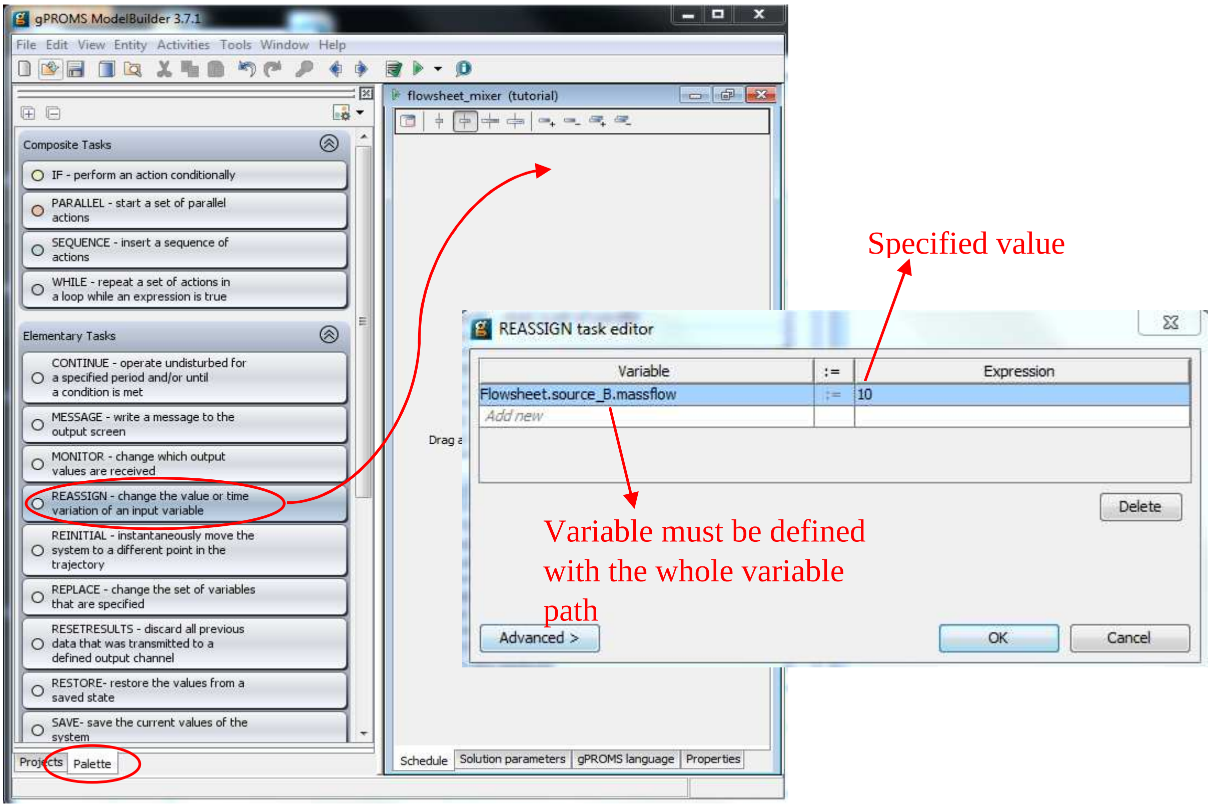
Schedules can be generated by using a graphical interface or entering the syntax in the gPROMS language. Both applications will be shown on the basis of the "flowsheet mixer"-process.

Elementary task- Reassign

The Reassign command enables you to change the values of specified variables. Let's see how it works in gPROMS.

- First the **Palette** tab needs to be activated by selecting **Palette** in the **View** menu in the **menu bar**.
- Open the "flowsheet_mixer" process and choose the **Schedule** tab
- Open the **Palette** tab to see the task possibilities.
- Drag the **REASSIGN** block into the **Schedule** tab.
- Define the Variable that should be changed and specify it with a value – click **OK**



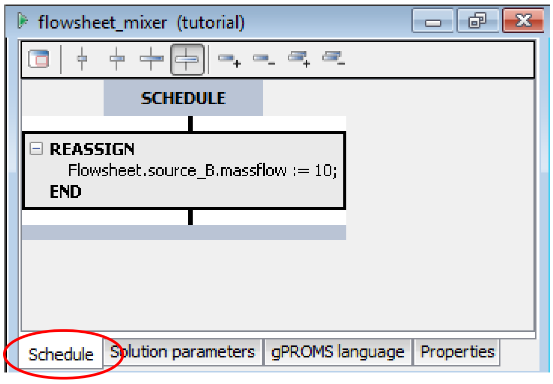

gPROMS ModelBuilder 3.7.1
File Edit View Entity Activities Tools Window Help
Composite Tasks
IF - perform an action conditionally
PARALLEL - start a set of parallel actions
SEQUENCE - insert a sequence of actions
WHILE - repeat a set of actions in a loop while an expression is true
Elementary Tasks
CONTINUE - operate undisturbed for a specified period and/or until a condition is met
MESSAGE - write a message to the output screen
MONITOR - change which output values are received
REASSIGN - change the value or time variation of an input variable
REINITIAL - instantaneously move the system to a different point in the trajectory
REPLACE - change the set of variables that are specified
RESETRESULTS - discard all previous data that was transmitted to a defined output channel
RESTORE- restore the values from a saved state
SAVE- save the current values of the system
Projects
Palette
flowsheet_mixer (tutorial)
Drag a
Schedule
Solution parameters
gPROMS language
Properties
REASSIGN task editor
Delete
Advanced >
OK
Cancel

Variable	:=	Expression
Flowsheet.source_B.massflow	:=	10
Add new		

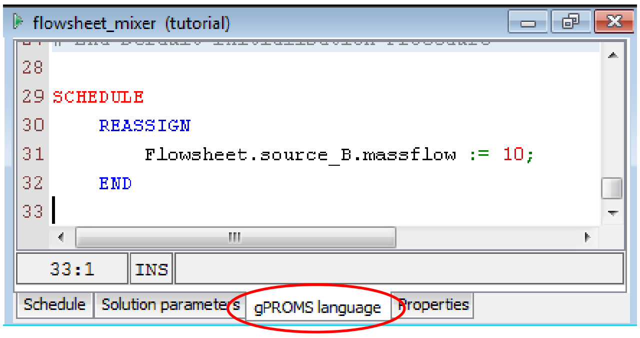
Specified value

Variable must be defined with the whole variable path

The schedule appears in the **Schedule** tab as well as in the **gPROMS language** tab. The other opportunity is to write the syntax directly into the **gPROMS language** tab.



```
REASSIGN
  Flowsheet.source_B.massflow := 10;
END
```



```
SCHEDULE
    REASSIGN
        Flowsheet.source_B.massflow := 10;
    END
```

Other applications of the Reassign task:

```
SCHEDULE
REASSIGN
    Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + value;
END
```

With that command, the new mass flow is higher than the starting value. Another option is to define a smaller one:

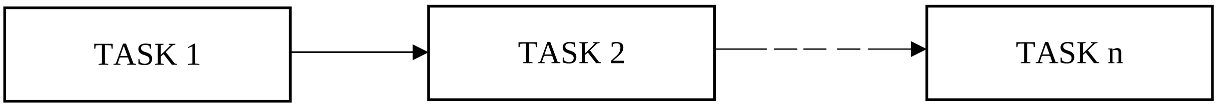
```
SCHEDULE
REASSIGN
    Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) - value;
END
```

Run the simulation after finishing the schedule. The results show two values- the first one is the calculation with the specified value for massflow_B and the second one shows the results after the modification. Be careful: with double click on the sink the report will only show the results of the last calculation (as they are both at the time 0).

If the reassignment should include more values in a row, it can be realized with a combination of composite and elementary tasks.

Composite Tasks- Sequence and Continue

The Sequence task executes a series of tasks within a sequential structure. It begins with the first task and only proceeds to the next when the execution has terminated.



The reassignment (value by value) can be done in a row. Use “+” for an increase of the starting value and “-“ to reduce it.

```
SCHEDULE
SEQUENCE
    REASSIGN
Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + / - value;
    END #reassign

    REASSIGN
Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + / - value;
     END #reassign
                                   .
                                   .
    END #sequence
```

This approach can be useful for just a few values.

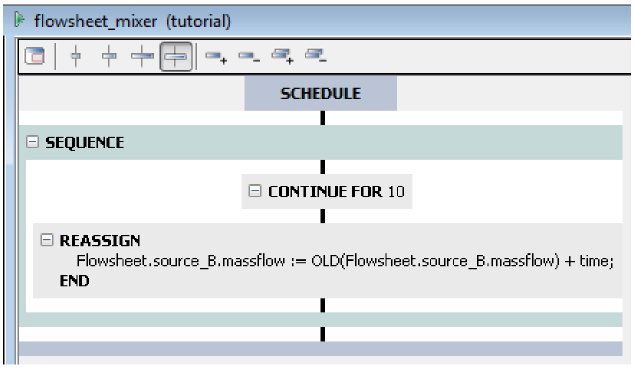
The Continue task executes any elementary task until it reaches a determined value. The end of the execution can either be the endpoint of a time period or a required condition that has been achieved. In our example those cases could look like the following:

Continue for a time period:

gPROMS language syntax:

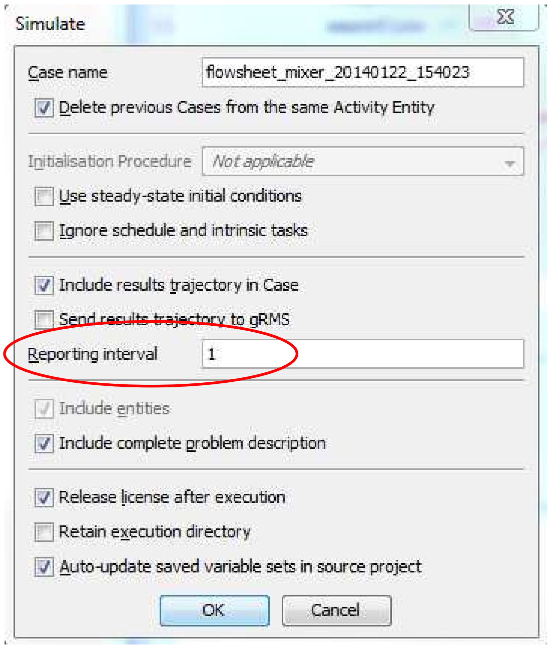
```
    SCHEDULE
        SEQUENCE
                REASSIGN
         Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + / -   time;
    END #reassign
         CONTINUE FOR 10;
        END #sequence
```

graphical method:



```
SCHEDULE
SEQUENCE
CONTINUE FOR 10
REASSIGN
  Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + time;
END
```

The massflow variable will now be reassigned ten times. To see all the steps in the results choose smaller values (1) as *Reporting interval* in the **Simulate window** (after starting the simulation with the **play-icon** ▶).



Simulate

Case name: flowsheet_mixer_20140122_154023

☑ Delete previous Cases from the same Activity Entity

Initialisation Procedure: *Not applicable*

□ Use steady-state initial conditions

□ Ignore schedule and intrinsic tasks

☑ Include results trajectory in Case

□ Send results trajectory to gRMS

Reporting interval: 1

☑ Include entities

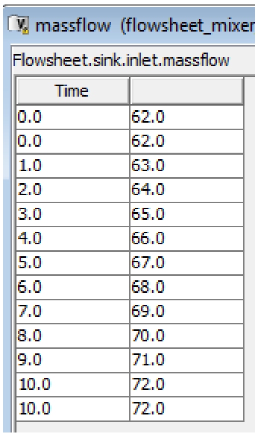
☑ Include complete problem description

☑ Release license after execution

□ Retain execution directory

☑ Auto-update saved variable sets in source project

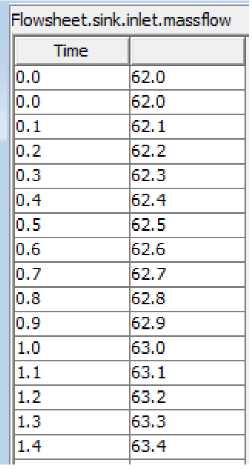
Results with reporting interval 1 on the left side and 0.1 on the right side.



massflow (flowsheet_mixer

Flowsheet.sink.inlet.massflow

Time	
0.0	62.0
0.0	62.0
1.0	63.0
2.0	64.0
3.0	65.0
4.0	66.0
5.0	67.0
6.0	68.0
7.0	69.0
8.0	70.0
9.0	71.0
10.0	72.0
10.0	72.0



Flowsheet.sink.inlet.massflow

Time	
0.0	62.0
0.0	62.0
0.1	62.1
0.2	62.2
0.3	62.3
0.4	62.4
0.5	62.5
0.6	62.6
0.7	62.7
0.8	62.8
0.9	62.9
1.0	63.0
1.1	63.1
1.2	63.2
1.3	63.3
1.4	63.4

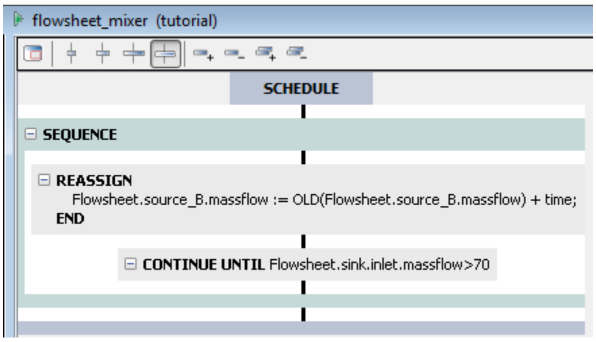
Continue until reaching a required condition

For this application it’s important that the use of “=” is not possible. The end of a sequence has to be formulated with ”<” , “>” , “<=” or “>=” expressions.

gPROMS language syntax:

```
SCHEDULE

SEQUENCE

    REASSIGN

    Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + / -   time; END
    #reassign

    CONTINUE UNTIL Flowsheet.sink.inlet.massflow > 70;

END #sequence
```

graphical method:



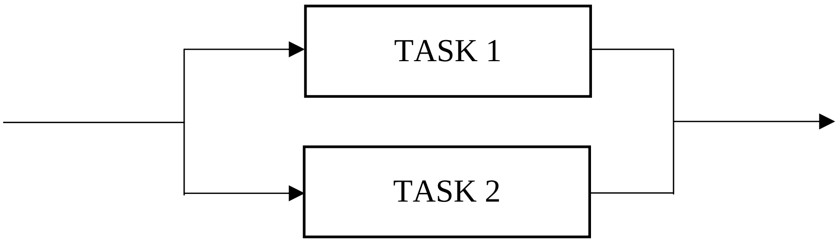
flowsheet_mixer (tutorial)

```
SCHEDULE
SEQUENCE
  REASSIGN
    Flowsheet.source_B.massflow := OLD(Flowsheet.source_B.massflow) + time;
  END
  CONTINUE UNTIL Flowsheet.sink.inlet.massflow>70
```

The simulation stops when the outlet massflow reaches the predefined value of 70. Within the Sequence it’s as well possible to reassign different variables succssively.

Composite Tasks- Parallel

The Parallel task executes two or a series of tasks simultaneously.



For instance:

```
SCHEDULE
PARALLEL
   REASSIGN
  Flowsheet.source_B.massflow := 10 ; END #reassign
  REASSIGN
  Flowsheet.source_A.massflow := 60 ; END #reassign
END #parallel
```

Have a look at the input massflows in the results to see what’s happening. Use the Schedule tab to see the graphical method.

Elementary task- Replace

The Replace task allows to replace a specified variable with an unspecified one. One advantage of is that an input variable can be substituted with one output. If a model has problems to do a calculation by defining an output variable, try to define an input variable and replace it within a schedule.

```
SCHEDULE
REPLACE
  Flowsheet.source_A.composition("methanol")
```

```
with
    Flowsheet.mixer.outlet.composition("methanol") := 0.8 ;
END #replace
```

Combine tasks

The Schedules are a special gPROMS tool which enables a lot of possible variations for sensitivity analysis, find design specifications by replacing the input with a desired output…

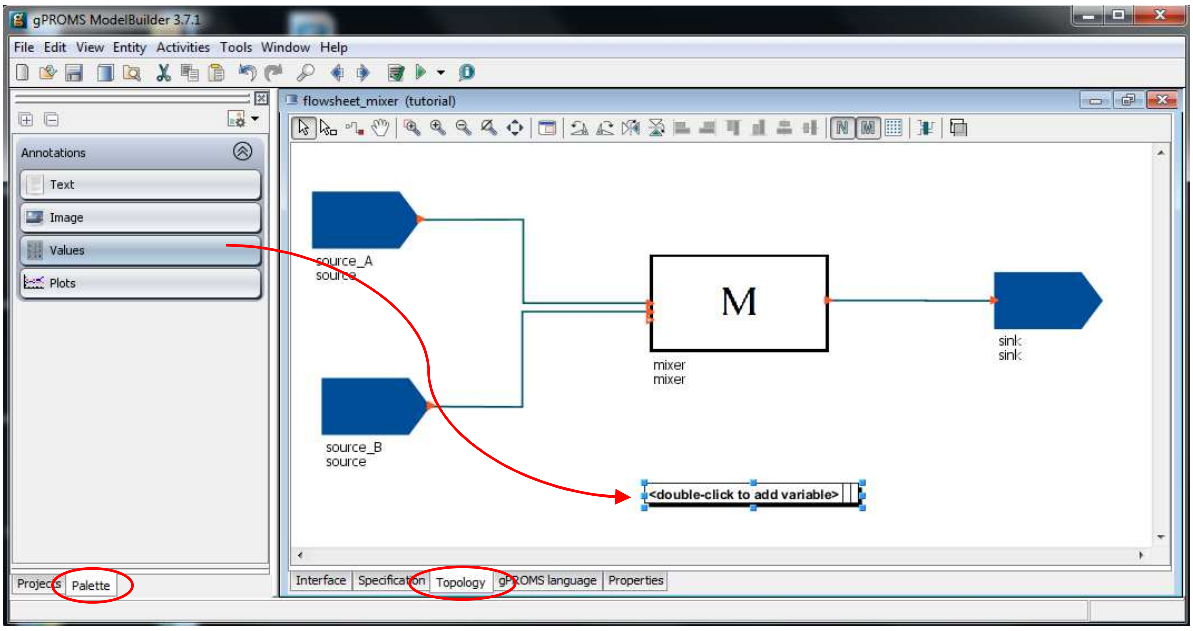
The tasks allow a lot of different combinations, one last example will be demonstrated:

```
SCHEDULE
PARALLEL
    REPLACE
    Flowsheet.source_A.composition("methanol")
    with
    Flowsheet.mixer.outlet.composition("methanol") := 0.8 ; END #replace
SEQUENCE
    REASSIGN
    Flowsheet.source_B.composition("water")  :=  OLD(Flowsheet.source_B.composition("water"))  -
    time/100;
END #reassign
    CONTINUE FOR 10
  END #sequence
END #parallel
```

Select 0.1 as *Reporting interval* in the **Simulate** window and carry out the simulation. For a better understanding what happens in this Schedule have a close look at the results (output as well as input values). It might happen that you reach limits with replacing or continuous reassigning. If that happens try to use other values or do the schedules with smaller steps, or even with value reassignment.

Adding graphs and other annotations to the flowsheet

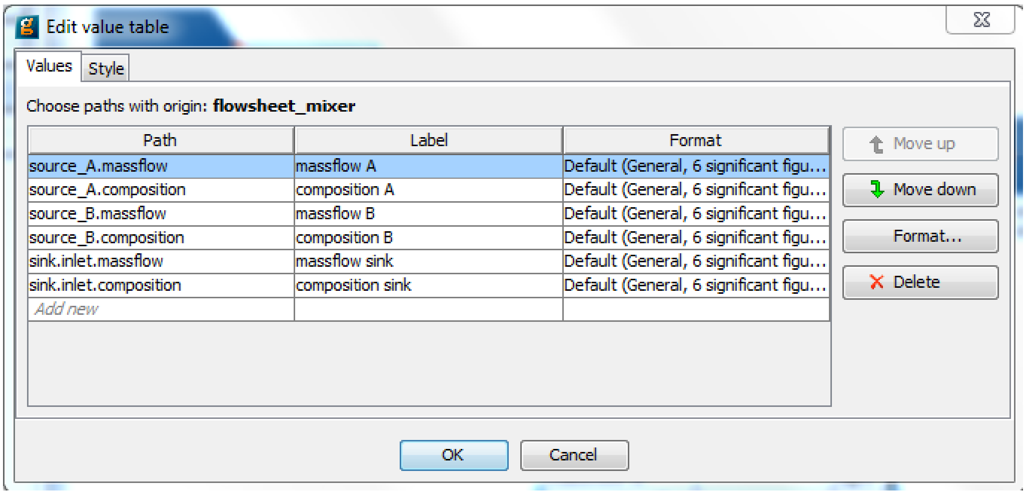
To insert annotations select the **Palette** tab on the bottom left. Choose a model and drag and drop it into the **Topology** tab of the flowsheet.



Values:

This annotation shows one ore a table of values of model variables.

- After placing it *double click* on the square that appears in the flowhseet.
- Type the whole variable paths that should be displayed (as you are in the flowsheet- you don't need to write that path here) in the colum **Path**
- Choose an appropriate name for the variable and type it into the **Label** column
- The **Format** column shows the number format of the values (can be changed with double click on the value and selecting the desired one or with the **Format…** button on the right side).

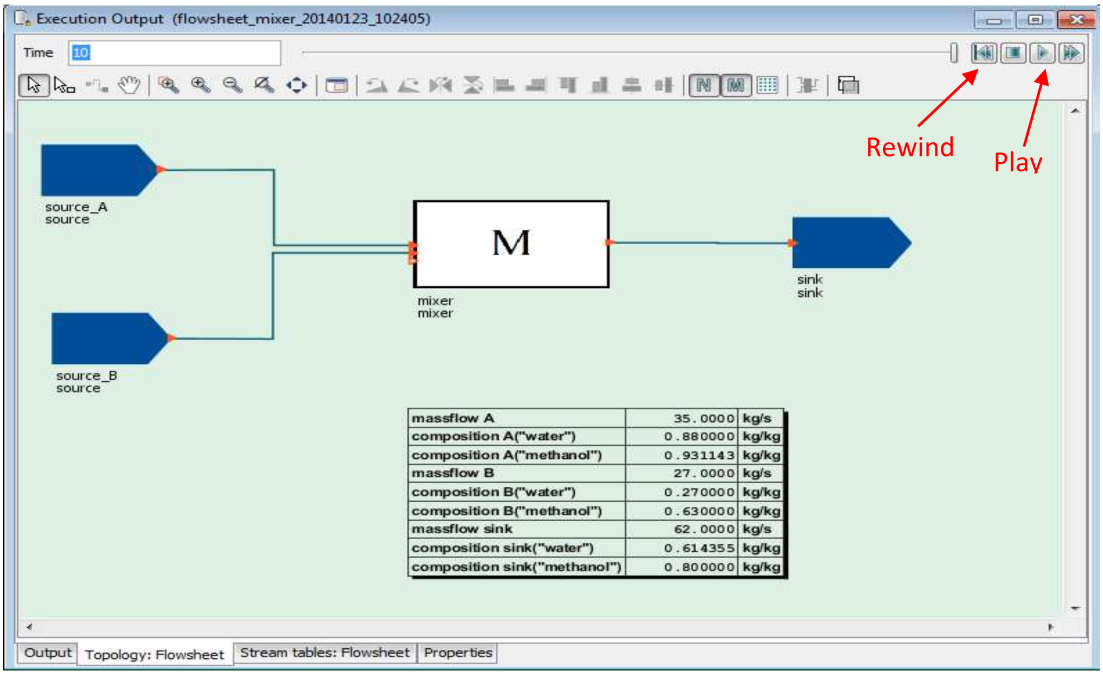


Choose paths with origin: **flowsheet_mixer**

Path	Label	Format
source_A.massflow	massflow A	Default (General, 6 significant figu…
source_A.composition	composition A	Default (General, 6 significant figu…
source_B.massflow	massflow B	Default (General, 6 significant figu…
source_B.composition	composition B	Default (General, 6 significant figu…
sink.inlet.massflow	massflow sink	Default (General, 6 significant figu…
sink.inlet.composition	composition sink	Default (General, 6 significant figu…
Add new		

- **Run** the Simulation **(***Reporting interval 1 or smaller)*

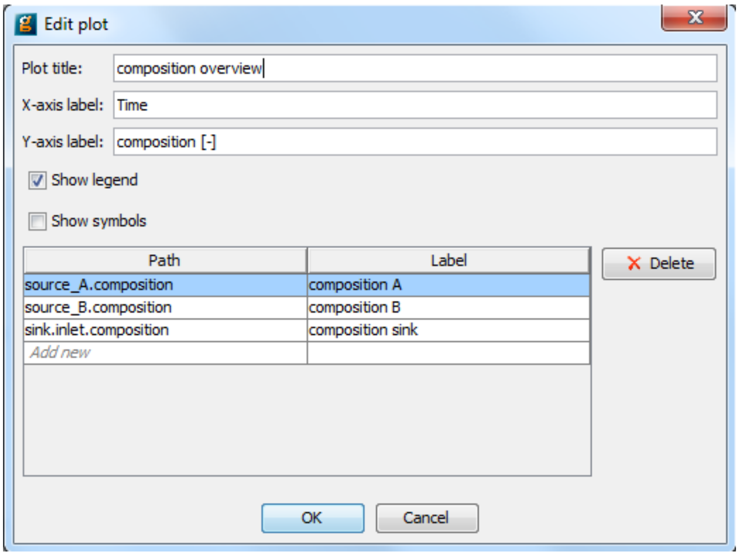
If there is a schedule in the process, the **Execution Output** shows the values from the latest time. To see the change of the values *click* on the **Rewind button** (reset displayed simulation time to zero) and then **Play**.



massflow A	35.0000	kg/s
composition A("water")	0.880000	kg/kg
composition A("methanol")	0.931143	kg/kg
massflow B	27.0000	kg/s
composition B("water")	0.270000	kg/kg
composition B("methanol")	0.630000	kg/kg
massflow sink	62.0000	kg/s
composition sink("water")	0.614355	kg/kg
composition sink("methanol")	0.800000	kg/kg

Plots:

This tool shows dynamic plots of chosen variables in order to easily follow the results of changes made with schedules. To create a plot *drag and drop* it onto the flowsheet and *edit* it in the **Edit Plot** window.



Plot title: composition overview

X-axis label: Time

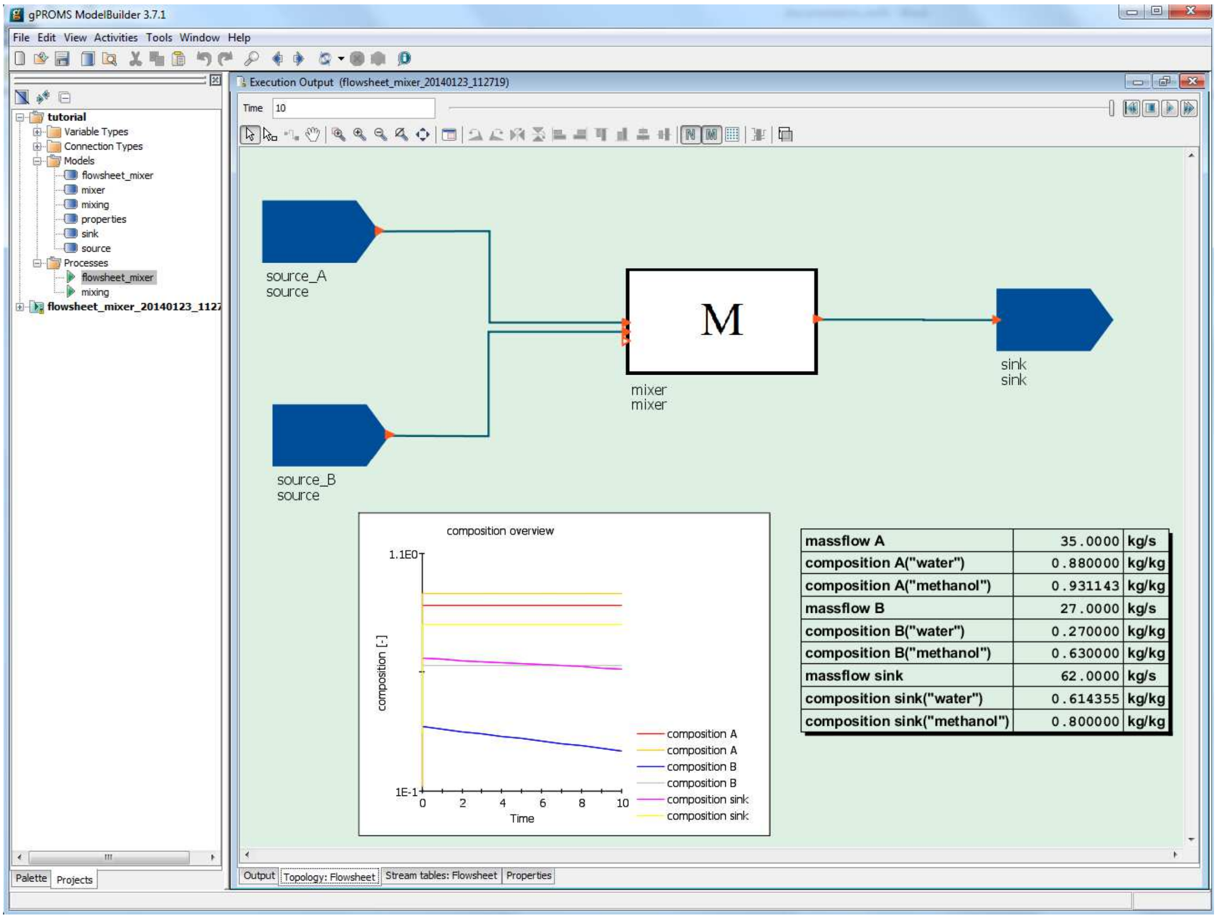
Y-axis label: composition [-]

☑ Show legend

□ Show symbols

Path	Label
source_A.composition	composition A
source_B.composition	composition B
sink.inlet.composition	composition sink
Add new	

After running the simulation the Output is as follows:



massflow A	35.0000	kg/s
composition A("water")	0.880000	kg/kg
composition A("methanol")	0.931143	kg/kg
massflow B	27.0000	kg/s
composition B("water")	0.270000	kg/kg
composition B("methanol")	0.630000	kg/kg
massflow sink	62.0000	kg/s
composition sink("water")	0.614355	kg/kg
composition sink("methanol")	0.800000	kg/kg

PML- Model Libraries

Introduction

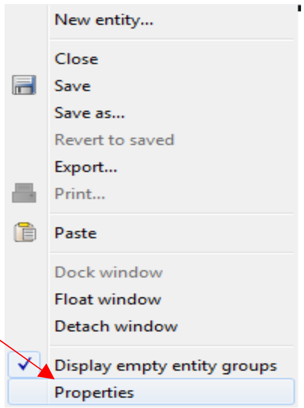
gPROMS provides a collection of basic units and flowsheets in the Process Model Library (PML). The PML models enable the gPROMS users to find solutions for their own model development, so it can be a very useful support.

Open the PML library:

- *Click* on the ▯ -icon in the toolbar.
- *Check* the **PML v3.n** box and *click* **OK**.
- The library with six complementary libraries is now visible in the **Project-tree** area
- Every library has its own documentation, they are saved under:

C:\Program Files (x86)\PSE\ModelBuilder_3.7.1.54883\doc\PML

Create a flowsheet



Create your own flowsheet using PML models:

- *Create* a new **Project**
- *Right click* on the **project** and select **Properties**
- Go to the tab **Cross-references**
- *Click* on the **Add-button** Add till it's not active anymore
- *Create* a new model
- Choose random models from the library
- If problems with connections arise- try to insert other units (pumps, valves, pipes…) in between
- If there are "physical properties" needed in any specification window, type: *"IPPFO::mass<methanol,water>"*
- Every input stream has to be created with a source and every output stream with a sink
- After creating the flowsheet specify the units, edit a Process and try to run it

Open a PML example flowsheet:

- *Click* on the -icon (Browse the example projects) in the toolbar
- *Open* the **PML flowsheets** folder and *select* one of the flowsheet examples
- Choose the Model and the Process (Hold *Ctrl* and *left click* on both)
- *Right click* and select *copy* (or *Ctrl+C*)
- Go to the Project *right click* and *Paste* (*Ctrl+V*)
- Now it can be adapted and used

Syntax-Order within the gPROMS language

gPROMS has a special **Syntax-order** (shown below- not complete in this illustration). To avoid mistakes in the **gPROMS language**, watch out that you follow that order:

PARAMETER

ParameterName AS INTEGER || REAL || LOGICAL < DEFAULT Value >

ParameterName AS ARRAY (Size < , ... >) OF INTEGER || REAL || LOGICAL < DEFAULT Value >

ParameterName AS FOREIGN_OBJECT < "ForeignObjectClass" > < DEFAULT "ForeignObjectValue" >

ParameterName AS ORDERED_SET < DEFAULT ["Name" < , ... >] >

UNIT

UnitName AS UnitModelName

UnitName AS ARRAY (Size || OrderedSet < , ... >) OF UnitModelName

PORT

PortName AS ConnectionType

PortName AS ARRAY (Size || OrderedSet < , ... >) OF ConnectionType

VARIABLE

VariableName AS VariableType

VariableName AS ARRAY (Size || OrderedSet < , ... >) OF VariableType

SELECTOR

SelectorName AS (Flag < , ... >) < DEFAULT Flag >

SelectorName AS ARRAY (Size || OrderedSet < , ... >) OF (Flag < , ... >) < DEFAULT Flag >

SelectorName AS DISTRIBUTION (DomainName < , ... >) OF (Flag < , ... >) < DEFAULT Flag >

SET

ParameterPath := Expression ;

ParameterPath := [Expression < , ... >];

ParameterPath := "<ForeignObjectClass::>ForeignObjectValue";

ParameterPath := ["Name" < , ... >];

TOPOLOGY

Unit connection equations

EQUATION

Model equations

ASSIGN

VariablePath := Expression ;

VariablePath := [Expression < , ... >];

INITIALSELECTOR

SelectorPath := FlagPath ;

SelectorPath := [FlagPath < , ... >];

INITIAL

Equations

The **syntax-order** can also be shown in the model:

Display the syntax order in the gPROMS language

- Create a new model
- Leave the **tick** at the **Use template?** box
- The syntax order is displayed in the **gPROMS language** tab.

FOR PRACTICE

A) Transform the mass flow into a molar flow (internally in the mixer model)- additionally try to display the value of the molar flow in the sink model

B) Implement the enthalpy balance (use enthalpy data from literature)