Modelica Chemical Library - User’s Guide

Marek Mateják1, Martin Tribula1, Filip Ježek2

1Department, University, Country, {name1,name3}@university.org

Company, Country, name2@company.com

Abstract

New library for chemical calculations is released. It is based on equilibrating the electro-chemical potentials of the substances as following the modern theory of physical chemistry. The library is very general. The chemical solution has full thermodynamic state with possibility to connect it with thermal, mechanical and electrical components of Modelica Standard Library 3.2.1. This paper show how to implement the lead-acid battery, the hydrogen burning engine and the chloride shift membrane channel in cellular membrane of human red cells. All these examples uses only very small amount of base components such as component for chemical solution, component for chemical substance or component for chemical reaction. Behind these components are only fundamental laws of thermodynamics and physical chemistry. The most of parameters are properties of the chemical substances, so the user does not set the dissociation coefficients of the reactions, because they can be calculated from Gibbs energies of the chemical reaction, which can be expressed from formation Gibbs energies of the substances using Hess’ law.

Keywords: keyword1, keyword2

# Introduction

The Chemical library becomes from very successive library for physiological calculations called Physiolibrary. We used the Physiolibrary to implement the biggest model of human physiology of the year 2010 called HumMod. Now we called the model Physiomodel and we are continuing the extended it in more detailed microscopic and chemical levels. The macroscopic processes and regulations of human physiology are already validated by experiments on animals and human. However the chemical processes in the models was until today a black-boxes with inputs and outputs defined more with empirical relationships than with strict physical chemistry theory. Using this empirical behavior it was well formed expectations for the elementary processes into the chemical library, which can be theoretically described in more details using last modern fundamental relations of thermodynamics and physical chemistry. This chemical library allows to move different substances in different direction through membrane at the same time, which was not possible using Modelica.Fluid package because stream constructs move the substances only in direction of the main stream.

This new chemical library solves the more detailed chemical environment of human cells and cellular chemical processes, where the Physiolibrary failed. For example the equilibration of osmolarities on capillary membrane works well, because of small amount of impermeable proteins. However, we found that in cellular membrane these relations was not in good agreement with measured data. These data shows Raftos et al. for human blood red cells, called erythrocytes, where total molarity of plasma is around 289 mmol/L and molarity of intracellular space is 207 mmol/L at osmotic equilibrium. The explanation of these disproportions was hidden in physical chemistry and when we equilibrate the chemical potential of water instead of osmolarity the capillary membrane and also this cellular membrane reach the expected values as measured in experiments. The other problem with old Physiolibrary approach was, that it does not automatically calculate the membrane equilibriums. The very specific blocks for calculating the Donnan’s equilibriums at glomerular membrane was created to reach expected concentrations of electrolytes at semipermeable membrane. However, there was not generated any membrane electric potential, which is the result of this electrolyte’s equilibrium. Even the neural information was spread by simplification describing of amplitudes of frequencies on specific neural drives. Now with the chemical library it can be automatically solved the Donnan’s equilibria on semipermeable membrane together with Nernst membrane potential as a simple consequence of the equilibrated electro-chemical potentials of the permeable substances.In the following section, short style guidelines are given.

During building we realized the new Modelica support of these chemical processes that they are general enough to describe much more than the equilibrium on the erythrocyte cellular membrane.

The result is a library, which allows to create any type of chemical reaction in any type of solution at any conditions. We made it in one hand with studding thermodynamics and physical chemistry relations behind. In Modelica it seems really that they can be only rewritten the selected base definitions from these theoretical approach. For example the definition of electro-chemical potential at chemical substance component, the thermodynamic relations at chemical solution component and the simple equilibration of electro-chemical potentials in all components of chemical processes. And all starts to work above our expectations.

Now we define only tree models of substances: ideal gas, incompressible substance and subunit substance. However the user can very easy redefine their own model of substance just by inserting the right expression for pure substance activity coefficient, molar volume, free molar formation entropy and free molar formation enthalpy based on substance data, temperature, pressure, electric potential and ionic strength. As object-oriented design allows also the definition of substance data is the part of the substance model. So the user can design the substance model based on totally different substance data, because this record is used only in the previously mentioned functions of the model, which should be also redefined in new substance model.

In our examples we will work only with ideal gas substances, which will be all gas substances, and with incompressible substances, which will be all liquid or solid substances. The definition data are molar mass of the substance, number of charge of the substance, molar heat capacity of the substance, free formation enthalpy (as known tabulated value for many chemical substances) and free formation Gibbs energy (as tabulated value). And the parameter density only for incompressible substance. So the full definition of ideal gas substance is only to create the record of the five parameters. This way was already defined more than 35 real chemical substances in example package of this chemical library. The usage of these predefined substances data are very simple. It the parameter dialog of the chemical substance can be selected the right record with this data as shown in Figure 1.

Figure 1, Setting of the predefined chemical substance, where (s) means solid phase, (aq) means dissolved in water, and (g) means gas phase and (l) means liquid phase.

This setting is typically the most important setting of each chemical model. All equilibrium coefficients are automatically solved using the selected substance data. As a result for example the chemical reaction component need to define only the stoichiometry coefficient and from the connected substances it reach the equilibrium at the right equilibrium coefficient.

# Example of lead-acid battery

The lead-acid electrochemical cells are characterized with two chemical reactions:

|  |  |
| --- | --- |
| PbO2 + HSO4- + 3 H+ +2 e- ↔ PbSO4 + 2 H2O | (1) |
| Pb + HSO4- ↔ PbSO4 + H+ + 2 e- | (2) |

Building of one cell of lead-acid battery using the library starts with definition of three solutions. Two for electrodes and one for acid solution. This can be done by drag and drop of the library class Components.SimpleSolution class into the diagram. The first instance we called “cathode” and the second “solution” and the last “anode” as in Figure 2A. We set the parameter ElectricalGround as “false” for all of these solutions to reach possibility of non-zero voltage of the solution. Now we can to specify the chemical substances inside the chemical solutions. We drag and drop of the library class Components.Substance into the “solution” as chemical substances H2O(liquid), H+(aqueous), HSO4-(aqueous) representing the liquid aqueous solution of sulfuric acid, into the “cathode” PbSO4(solid) and PbO2(solid) representing as the elements of positive electrode, and into “anode” the substances Pb(solid) and PbSO4(solid) representing the elements of negative electrode, see Figure 2B. As was mentioned for all this substances must be selected appropriate substance data definition, e.g. Examples.Substances.Water\_liquid, Lead\_solid, Lead\_dioxide\_solid, Lead\_sulfate\_solid and so on. The last, very special substance is an electron. This class is called Components.Electrone and it must be added for each electrode as in Figure 2C to translate electron flows from the chemical reaction to the electric current. Each of these substances must be connected to the appropriate solution using solution port as expressed in Figure 2B,C. Having all substances it is possible to implement the chemical reactions. Drag and drop of library class Components.Reaction for both chemical reactions and setting they parameters into appropriate number of reactants and products and their stoichiometry allows to connect each substance with the reaction as expressed in reactions (1,2) and Figure 2D. This setting can be done using parameter dialog of the cathode chemical reaction (1) as there are four types of substrates (nS=4) with stoichiometric coefficients 1 for first and second reactant, 3 for third reactants and 2 for forth reactants (s={1,1,3,2}) and there are two types of products (nP=2) with stoichiometry 1 for PbSO4 and 2 for water (p={1,2}) following the chemical scheme of the first chemical reaction above. After setting the number of reactants and products there is possible to connect the substances with reaction. Each instance of reaction was an array of connectors for substrates and an array of connectors for products and the user must be very careful to connect each element of this array with another substance in the same order as defined stoichiometric coefficients. This means that for example the water must be connected in index 2 of products of the first chemical reaction, because the second product was stoichiometry 2 as defined for H2O. The chemical reaction (2) must be set analogically as nS=2, nP=3, p={1,1,2} with connections of substance ports of Pb to substrate[1], HSO4- to substrate[2], PbSO4 to product[1], H+ to product[2] and e- to procuct[3] as graphically represented in Figure 2D.

Figure 2, Building of one electro-chemical cell of lead-acid battery in four steps: A) adding solution components, B) adding substance components, C) adding electron components, D) adding reaction components

Now, the electrochemical cell is already implemented. However the simulation need the initial state of substances, which for the fully charged battery means that almost all elements of cathode is PbO2 and almost all elements anode is Pb. At this state can be the sulfuric acid very concentrated, what increases the effectivity of the electrochemical cell. To set this state it is possible just double-click on PbO2 and Pb and set the amount e.g. 1mol, the same as the total amount of “cathode” and “anode” components. These total amounts of solutions must be equal or greater than sum of all substances inside. To set them the user must double click on the border of solution and using the parameter dialog rewrite the value of amountOfSolution\_start also to 1mol for both electrodes. If we went to examine also mass of volume properties of electrodes, there must be set the initial values of these physical quantities. However they in this example do not have impact to generated electric currents or electric potentials, so without we can ignore them this time without loss of generality. To set the pure concentrated sulfuric acid we can set the amount of SO4- and H+ also to 1 mol, what makes the total amount of substances in liquid “solution” 2 mol (solution1. amountOfSolution\_start=2). This fully charged ideal state is ready to simulate when it is connected the electric ground into one of the electric port of the one electron.

These battery can be connected to any electrical circuit, which is slowly discharging it. For example if we only connect the simple electric resistance of 1 Ohm as expressed in Figure 3 then the simulation of discharging process during 13 hours and 45 minutes gives the results of electric current and electric potential as in Figure 4.

Figure 3, Disscharging experiment of the lead-acid battery cell

Figure 4, Discharging simulation of lead acid battery cell from of Figure 3with the initial amount of substances as described in text.

# Example of hydrogen burning engine

In contrast with oxidation-reduction reactions describing processes in lead-acid electrochemical cell is the gaseous reaction of the hydrogen burning very simple:

|  |  |
| --- | --- |
| 2 H2 + O2 <-> 2 H2O | (3) |

However this reaction generates big amount of energy, which can be used for mechanical or thermal purposes.

The building of this model using the chemical library is very easy. At first we drag and drop the library class Components.IdealGasSolution to the diagram of our new model. The instance is called “idealsGas” in Figure 5. For this solution we need to set the area of the piston (e.g. 1 dm2), where the pressure makes the force of the green mechanical port of the upper side. The next parameter is the initial volume of the gas inside (e.g. 1 liter). Then all thee substance can be added using drag and drop of library class Components.Substance. Because this model uses gases, the substanceModel must be changed to some gases substance such as ideal gas substance model prepared in library as Interfaces.IdealGasSubstanceModel. The substance data must be selected to define the appropriate substance as Examples.Substances.Hydrogen\_gas, Examples.Substances.Oxygen\_gas and Examples.Substances.Water\_gas. And the initial amount of substances can be prepared for ideal solution of hydrogen and oxygen gases in ratio 2:1 to reach the chemical equation above with naïve expectation that at the end of burning only water substance will be presented. If we want to start with zero pressure at the beginning of the simulation we must to calculate the amount of particles using ideal gas equation, so there should be (2/3)\*p\*V/(R\*T) of H2 particles and (1/3)\*p\*V/(R\*T) of O2 particles, where p=100 000 Pa, V=0.001 m3, R=8.314 J/(K.mol), T=298 K. All substances must be connected with solution using solution port of blue color expressed typicallu in the bottom side of each substance and each solution. Then the chemical reaction is inserted into the diagram of this model as library class Components.Reaction and it is set to 2 substrates (nS=2) with stoichiometry s={2,1} and one product with stoichiometry 2 (p={2}). The substances are then connected using substance connectors of violet color with appropriate indexes: H2 to substates[1], O2 to substates[2] and H2O to products[1]. Now is the model prepared to simulate in condition of not connected heat port and not connected mechanical port. This simulation reach the theoretical ideal thermally isolated (zero heat flow from/to the solution) and isobaric (zero force generated on piston) conditions.

However in the real word there is always some thermal energy flow from the solution and this cooling process can be connected using the thermal connector of Modelica Standard Library 3.2.1. For example as simple thermal conductor of thermal conductance 2W/K to constant temperature environment at 25°C as expressed in Figure 5. The mechanical power of the engine can be connected to robust mechanical model. However in our example we select only very strong mechanical spring with spring constant of 1e6 N/m to stop the motion of the piston to generate the pressure, see Figure 5. The results of this experiment is shown in Figure 6.

Figure 5, Hydrogen burning engine with the spring above the piston and the cooling to constant temperature environment.

Figure 6, Simulation of hydrogen burning experiment of Figure 5. Initial phase of explosion is very fast - temperature reaches 3600°C from 25°C, the pressure reaches 10 bar from 1 bar. This pressure and temperature is generated because of very strong spring, which allows to change the volume only about 8% during the explosion.

# Example of chloride shift

Only in last century was discovered the real function of membranes in biology. Division of solutions of organism makes different conditions for different processes. For example each cell of the body is surrounded by cellular membrane composed with special very compact lipid double-layer. Any lipophobic compound (not soluble in lipids) cannot cross it without special proteins called membrane channels. So even a water molecules must have a membrane channels (called aquaporines) to cross the cellular membrane. The chloride shift (also known as Hamburger shift) is in both directions exchanging an aqueous chloride Cl- for an aqueous bicarbonate HCO3- across the cellular membrane of blood red cells using the membrane channel “Band 3”. As each passive membrane channel it allows only to equilibrate the electro-chemical potentials of the specific permeable ions on both sides of membrane. The different electric potentials on each side of membrane causes different concentrations at equilibrium. And backwards, the equilibrium of different ions compositions of the solution on both sides of the membrane causes the measurable electric membrane potential. It is not so intuitive, because even the both solution does not have any electric charge there can be a non-zero electric potential for permeable ions. This potential for permeable ions at equilibrium is called Nernst membrane potential and in the chemical library it is a direct mathematical result of the equality of electro-chemical potential of the ion in the both solutions.

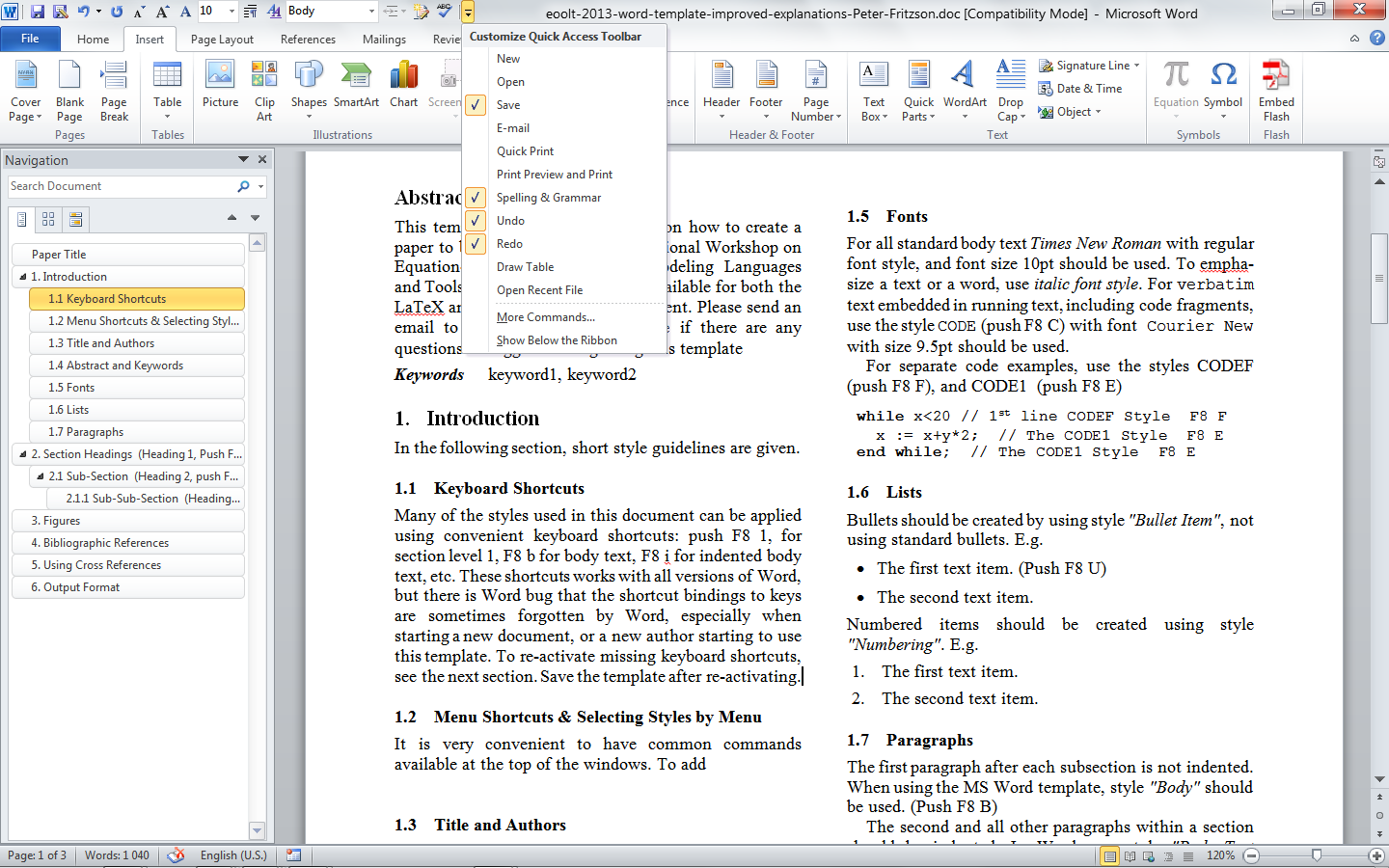
The mature red cell is the simplest cell in the body. Its primary function is transport of blood gases as oxygen O2 (from lungs to tissues) and carbon dioxide CO2 (from tissue to lungs). The chemical processes behind the gases transport is a bit complex, because the capacity of the water for only their free dissolved forms is very low. To transport sufficient amount of O2 and CO2 the gas must be chemically bound to hemoglobin as described in Matejak et al. and/or transported as different substance, which can be presented in water in much higher concentration than only free dissolved form allows. So to transport sufficient amount of CO2 it must be changed to HCO3- using chemical reaction:

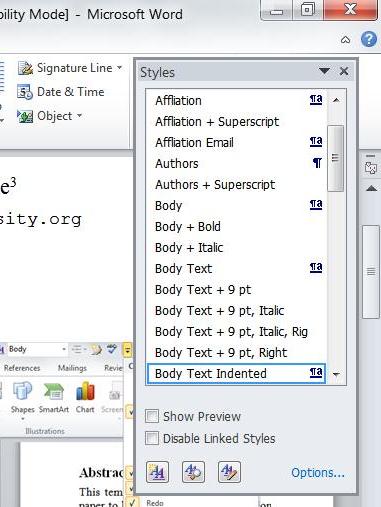
|  |  |
| --- | --- |
| CO2 + H2O <-> HCO3- + H+ | (4) |

Almost all of this change takes place inside the red cells, where is presented an enzyme called carbonic anhydrase. This effect can be simplified by slow down the reaction of in the blood plasma (KC=1e-15 mol2/(s.J)).

# Results

# Discussion





## Abstract and Keywords

The abstract should be written as one paragraph. It is not recommended to exceed 150 words.

Appropriate keywords describing the content of the paper should be supplied as a comma separated list.

Figure 1. An example of a figure that spans over two columns. To move it, click on its border, push shift, click again, move without releasing the mouse. Release shift if need to move it vertically. You can also use arrow buttons to move it.



## Fonts

For all standard body text *Times New Roman* with regular font style, and font size 10.5pt should be used. To emphasize a text or a word, use *italic font style*. For verbatim text embedded in running text, including code fragments, use the style CODE (push F8 C) with font Courier New with size 9.5pt should be used.

For separate code examples, use the styles CODEF (push F8 F), and CODE1 (push F8 E)

**while** x<20 // 1st line CODEF Style F8 F

x := x+y\*2; // The CODE1 Style F8 E

**end** **while;** // The CODE1 Style F8 E

## Lists

Bullets should be created by using style *"Bullet Item"*, not using standard bullets. E.g.

* The first text item. (Push F8 U)
* The second text item.

Numbered items should be created using style *"Numbering"*. E.g.

1. The first text item.
2. The second text item.

## Paragraphs

The first paragraph after each subsection is not indented. When using the MS Word template, style *"Body"* should be used. (Push F8 B)

The second and all other paragraphs within a section should be indented. In Word, use style *"Body Text Indented"* for these paragraphs. (Push F8 I)

# Section Headings (Heading 1, Push F8 1)

Section headings should be numbered. Words in the headings should be capitalized. The, style *"Heading 1"* should be used for the main section headings.

## Sub-Section (Heading 2, push F8 2)

Subsections are numbered and style *"Heading 2"* should be used.

### Sub-Sub-Section (Heading 3, push F8 3)

It is possible to use sub-sub-sections (style *"Heading 3"*). However, if possible, only sections and sub-sections should be used.

# Figures

Figures should be numbered and include a description text. All figures should be referenced within the body text by an insert cross reference to Figure (onlylabel+number).



Figure 2. Example of figure text.

When using MS word, style *"Caption"* could be used, as shown in the example. The style "*Figure*" should be used for the line where the picture is placed[[1]](#footnote-1).

It is recommended to copy a figure + caption such as the one below and modify the contents when creating a new figure.

The figure numbering will be automatically updated when selecting the whole article text by ctrl-A, and pushing alt/shift-U for the number updating.



Figure 3. Example of figure text.

This is another Body paragraph, followed by another Figure.



Figure 4. Example of figure text.

The following is an example table:

Table 1. Sizes of Compiler Phases, Lines of Code.

|  |  |
| --- | --- |
| *Compiler Phase* | *Lines* |
| BackEnd (from flat Modelica to sorted eq.syst.) | 29190 |
| Code generation (generated code) | 35971 |
| Code generation (template source code) | 8957 |
| FrontEnd (up to flat Modelica) | 92192 |
| OpenModelica scripting environment | 21883 |
| Template language Susan compiler | 12119 |
| Unparsing modules | 16984 |
| Utility modules | 12983 |
| *Total size (excl. generated code)* | *194218* |

# Equations

Equations should be numbered on the right side, such as:

|  |  |
| --- | --- |
|  | (5) |

There can be also more complicated formulae:

|  |  |
| --- | --- |
|  | (6) |

# Bibliographic References

The bibliographic reference list are shown at the end of the paper; starting with an unnumbered heading *"References"*. The list of references should be sorted in alphabetic order according to the first author's surname.

Citations are stated within the body text by inserting at most two author names and the publication year in parenthesis, such as (Pantelides, 1988) or (Tuffin *et al*, 2001). Two or more citations can be combined, separated by semicolon: (Pantelides, 1988; Augustin, Fineberg et al 1967)

# Using Cross References

To insert a cross reference to a heading, use menu item Insert->Cross-reference, reference type Heading, select reference to paragraph number.

To insert a cross reference to a Figure, use menu item Insert->Cross-reference, reference type Figure, select reference to Only Label and number.

To update all cross-references and re-number figures, first select all text (ctrl-A), then perform the update command by pushing 3 keys simultaneously: Alt-Shift-u

# Output Format

The paper should be submitted as a PDF-file using page size A4 (not US Letter!). All fonts should be included in the PDF-file.

Note: When using PDF-generators such as Adobe Acrobat PDF generator, remember to enable high-quality output. If this option is not enabled, figures and photos may be reduced in quality, resulting in poor quality when printed.

Acknowledgements

The templates and instructions were created by Peter Fritzson with feedback from Martin Otter, based on an earlier EOOLT workshop version created by Peter Fritzson and David Broman.

References

*References are in 9.5 pt font size, and should be ordered alphabetically according to the surname of the first author. Shortcut key to apply a reference style is F8 R. It is recommended to supply a DOI at the end of the reference.*

Donald C. Augustin, Mark S. Fineberg, Bruce B. Johnson, Robert N. Linebarger, F. John Sansom, and Jon C. Strauss. The SCi Continuous System Simulation Language (CSSL). *Simulation*, No 9, pp. 281–303, 1967.

Iain S. Duff and John K. Reid. An Implementation of Tarjan’s Algorithm for the Block Triangularization of a Matrix. *ACM Transactions on Mathematical Software*, 4(2):137–147, 1978. doi:

Constantinos C. Pantelides (1988): The Consistent Initialization of Differential-Algebraic Systems. *SIAM Journal on Scientiﬁc and Statistical Computing*, 9(2):213–231, 1988.

Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press. 2002. doi:10.1145/2593882.2593895.

B. Tuffin, D. S. Chen and K. Trivedi. Comparison of hybrid systems and fluid stochastic Petri nets. *Discrete Event Dynamic Systems*, 11 (1/2):77-95, 2001. doi:10.4271/2009-01-0746.

1. Footnotes should be numbered and located at the bottom of the column. [↑](#footnote-ref-1)