Publikované práce

Přílohy 1 – 12
Restoration of Guyton diagram for regulation of the circulation as a basis for quantitative physiological model development

Kofránek Jiří and Rusz, Jan

str. 897-908

Restoration of Guyton’s Diagram for Regulation of the Circulation as a Basis for Quantitative Physiological Model Development

J. KOFRÁNEK¹, J. RUSZ¹,²

¹Laboratory of Biocybernetics, Department of Pathophysiology, First Faculty of Medicine, Charles University in Prague, Czech Republic, ²Department of Circuit Theory, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

Received June 25, 2009
Accepted April 16, 2010
On-line June 9, 2010

Summary
We present the current state of complex circulatory dynamics model development based on Guyton’s famous diagram. The aim is to provide an open-source model that will allow the simulation of a number of pathological conditions on a virtual patient including cardiac, respiratory, and kidney failure. The model will also simulate the therapeutic influence of various drugs, infusions of electrolytes, blood transfusion, etc. As a current result of implementation, we describe a core model of human physiology targeting the systemic circulation, arterial pressure and body fluid regulation, including short- and long-term regulations. The model can be used for educational purposes and general reflection on physiological regulation in pathogenesis of various diseases.

Key words
Body fluid homeostasis • Blood pressure regulation • Physiological modeling • Guyton’s diagram

Corresponding author
J. Rusz, Department of Circuit Theory, Faculty of Electrical Engineering, Czech Technical University, Technická 2, 166 27 Prague 6, Czech Republic. E-mail: ruszjan@fel.cvut.cz

Introduction
The landmark achievement closely associated with integrative physiology development was the circulatory dynamics model published by Prof. Arthur C. Guyton and his collaborators (Guyton et al. 1972). Subsequently, its more detailed description was published in the monograph one year later (Guyton et al. 1973). This model represents the first large-scale mathematical description of interconnected physiological subsystems of the body. The model was described by a sophisticated graphic diagram with various computing blocks symbolizing quantitative physiological feedback connections. The diagram was published as a picture and the actual realization of the model was implemented in the FORTRAN language.

Although the FORTRAN implementation worked correctly, the diagram contains a number of errors that cause wrong model behavior. Moreover, FORTRAN implementation is not in agreement with this famous graphic diagram, it is almost unavailable nowadays, and contains several programming and computation-related features that require special treatment (Thomas et al. 2008). Despite the fact that the model was published over 30 years ago, it is currently used as a base for a number of research studies in the field of physiology (Montani and Van Vliet 2009, Osborn et al. 2009) and physiological modeling (Hunter et al. 2002, Bassingthwaighte 2000, 2009, Thomas et al. 2008), including research on the physiological consequences of weightlessness in manned space flight (White et al. 1991, 2003), or in a new approach to automation in medicine (Nguyen et al. 2008). In addition, the diagram is still reprinted with the original errors (Hall 2004, Bruce and Montani 2005). The overall revision of the diagram requires exhaustive search for errors and sophisticated analyses of physiological regulations system.

Here, we present a prototype of a core model of human physiology based on the original Guyton’s diagram targeting the short- and long-term regulation of
blood pressure, body fluids and homeostasis of the major solutes. This model also includes the hormonal (antidiuretic hormone, aldosterone and angiotensin) and nervous regulators (autonomic control), and the main regulatory sensors (baro- and chemoreceptors). Our complex circulatory dynamics model corresponds to the same graphic notation of the original Guyton’s diagram and adheres to its basic physiological principles. While new models are continuously being developed (Srinivasan et al. 1996, Abram et al. 2007, Hester et al. 2008), our model finally brings a fully functional modification of the original Guyton’s diagram, which is more suitable for a better and deeper understanding of the importance of physiological regulations and their use in development of many pathophysiological conditions by using simulation experiments.

The resulting model can be used as a baseline for the quantitative physiological model development designated for physicians’ e-learning and acute care medicine simulators. This model can also be used as an effective learning aid for physiological regulation systems education, connected with biomedical engineering specialization. The model is provided as an open-source and it is downloadable at <http://physiome.cz/guyton/>.

**Methods**

**Mathematical model of global physiological regulation of blood pressure**

The model consists of 18 modules containing approximately 160 variables and including 36 state variables (see Table 1 for more details). Each module represents an interconnected physiological subsystem (kidney, tissue fluid, electrolytes, autonomous nervous regulation and hormonal control including antidiuretic hormone, angiotensin and aldosterone). The model is constructed around a ‘central’ circulatory dynamics module in interaction with 17 ‘peripheral’ modules corresponding to physiological functions (Fig. 1) and complete model targeting the systemic circulation, arterial pressure and body fluid regulation, including short- and long-term regulations. A graphic presentation of the model allows a display of the connectivity among all physiological relationships. In essence, the model contains a total of approximately 500 numerical entities (model variables, parameters and constants). Members of the original Guyton’s laboratory have been continuously developing a more sophisticated version of the model, which is used for teaching (Abram et al. 2007). Although it includes about 4000 variables, this more elaborate model is less suited for our purposes than the model by Guyton et al. (1972), because of its incomplete description and physiological relationships formulation.

**Physiological regulations system analyses**

The original model represented as a sophisticated graphic diagram contains a number of errors which imply entirely incorrect physiological model behavior. The correction of these errors demanded complicated physiological regulations system analyses.
Table 1. List of state variables used in the original Guyton’s diagram with physiological significances, block numbers, and abbreviations.

<table>
<thead>
<tr>
<th>State variable in selected subsystem</th>
<th>Block number</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Circulatory dynamics</strong></td>
<td>1 - 60</td>
<td></td>
</tr>
<tr>
<td>01. Venous vascular volume</td>
<td>6</td>
<td>VVS</td>
</tr>
<tr>
<td>02. Right atrial volume</td>
<td>13</td>
<td>VRA</td>
</tr>
<tr>
<td>03. Volume in pulmonary arteries</td>
<td>19</td>
<td>VPA</td>
</tr>
<tr>
<td>04. Volume in left atrium</td>
<td>25</td>
<td>VLA</td>
</tr>
<tr>
<td>05. Volume in systemic arteries</td>
<td>31</td>
<td>VAS</td>
</tr>
<tr>
<td><strong>Vascular stress relaxation</strong></td>
<td>61 - 65</td>
<td></td>
</tr>
<tr>
<td>06. Increased vascular volume caused by stress relaxation</td>
<td>65</td>
<td>VV7</td>
</tr>
<tr>
<td><strong>Capillary membrane dynamics</strong></td>
<td>66 - 82</td>
<td></td>
</tr>
<tr>
<td>07. Plasma volume</td>
<td>71</td>
<td>VP</td>
</tr>
<tr>
<td>08. Total plasma protein</td>
<td>80</td>
<td>PRP</td>
</tr>
<tr>
<td><strong>Tissue fluids, pressure and gel</strong></td>
<td>83 - 113</td>
<td></td>
</tr>
<tr>
<td>09. Total interstitial fluid volume</td>
<td>84</td>
<td>VTS</td>
</tr>
<tr>
<td>10. Volume of interstitial fluid gel</td>
<td>101</td>
<td>VG</td>
</tr>
<tr>
<td>11. Interstitial fluid protein</td>
<td>103</td>
<td>IFP</td>
</tr>
<tr>
<td>12. Total protein in gel</td>
<td>112</td>
<td>GPR</td>
</tr>
<tr>
<td><strong>Electrolytes and cell water</strong></td>
<td>114 - 135</td>
<td></td>
</tr>
<tr>
<td>13. Total extracellular sodium</td>
<td>118</td>
<td>NAE</td>
</tr>
<tr>
<td>14. Total extracellular fluid potassium</td>
<td>122</td>
<td>KE</td>
</tr>
<tr>
<td>15. Total intracellular potassium concentration</td>
<td>131</td>
<td>KI</td>
</tr>
<tr>
<td><strong>Pulmonary dynamics and fluids</strong></td>
<td>136 - 152</td>
<td></td>
</tr>
<tr>
<td>16. Pulmonary free fluid volume</td>
<td>142</td>
<td>VPF</td>
</tr>
<tr>
<td>17. Total protein in pulmonary fluids</td>
<td>149</td>
<td>PPR</td>
</tr>
<tr>
<td><strong>Angiotensin control</strong></td>
<td>153 - 163</td>
<td></td>
</tr>
<tr>
<td>18. Angiotensin concentration</td>
<td>159</td>
<td>ANC</td>
</tr>
<tr>
<td><strong>Aldosterone control</strong></td>
<td>164 - 174</td>
<td></td>
</tr>
<tr>
<td>19. Aldosterone concentration</td>
<td>170</td>
<td>AMC</td>
</tr>
<tr>
<td><strong>Antidiuretic hormone control</strong></td>
<td>175 - 189</td>
<td></td>
</tr>
<tr>
<td>20. Degree of adaption of the right atrial pressure</td>
<td>180</td>
<td>AHY</td>
</tr>
<tr>
<td>21. Antidiuretic hormone concentration</td>
<td>185</td>
<td>AHC</td>
</tr>
<tr>
<td><strong>Thirst and drinking</strong></td>
<td>190 - 194</td>
<td></td>
</tr>
<tr>
<td><strong>Kidney dynamics and excretion</strong></td>
<td>195 - 222</td>
<td></td>
</tr>
<tr>
<td><strong>Muscle blood flow control and PO2</strong></td>
<td>223 - 254</td>
<td></td>
</tr>
<tr>
<td>22. Rate of increase in venous vascular volume</td>
<td>231</td>
<td>DVS</td>
</tr>
<tr>
<td>23. Total volume of oxygen in muscle cells</td>
<td>238</td>
<td>QOM</td>
</tr>
<tr>
<td>24. Muscle vascular constriction caused by local tissue control</td>
<td>254</td>
<td>AMM</td>
</tr>
<tr>
<td><strong>Non-muscle oxygen delivery</strong></td>
<td>255 - 272</td>
<td></td>
</tr>
<tr>
<td>25. Non-muscle venous oxygen saturation</td>
<td>260</td>
<td>OSV</td>
</tr>
<tr>
<td>26. Non-muscle total cellular oxygen</td>
<td>271</td>
<td>QO2</td>
</tr>
<tr>
<td><strong>Non-muscle, non-renal local blood flow control</strong></td>
<td>273 - 290</td>
<td></td>
</tr>
<tr>
<td>27. Vasoconstrictor effects of rapid autoregulation</td>
<td>278</td>
<td>AR1</td>
</tr>
<tr>
<td>28. Vasoconstrictor effects of intermediate autoregulation</td>
<td>285</td>
<td>AR2</td>
</tr>
<tr>
<td>29. Vasoconstrictor effects of long-term autoregulation</td>
<td>289</td>
<td>AR3</td>
</tr>
<tr>
<td><strong>Autonomic control</strong></td>
<td>291 - 320</td>
<td></td>
</tr>
<tr>
<td>30. Time delay for realization of autonomic drive</td>
<td>305</td>
<td>AU4</td>
</tr>
</tbody>
</table>
These include exhaustive revision of the complete model and its behavior validation using several simulation experiments. In this stage, the original FORTRAN code of the Guyton’s model was also used to compare the obtained simulation results. It is of course the case that the original FORTRAN code runs correctly; the errors were only in the diagram.

Because it would be beyond the scope of this paper to discuss each error in the original Guyton’s diagram, as an example of the system analyses, we describe the five most significant errors which would have the greatest role in creating the unpredictable model behavior (Fig. 2). The other errors are mostly caused by replaced mathematical operations, wrong set of normalization and damping constants, and replaced signs that determine the positive or negative feedback.

The first error is the wrong flow direction marking of blood flow in the circulatory dynamics subsystem (Fig. 2a). The rate of increase in systemic venous vascular blood volume (DVS) is the subtraction between all rates of inflows and rates of outflows. Blood flow from the systemic arterial system (QAO) means inflow and rate from veins into the right atrium (QVO) means outflow. Rate change of the vascular system filling as the blood volume changes (VBD) is calculated as the difference between the summation of vascular blood compartments and blood volume overall capacity, meaning that VBD is found in the outflow rate too. Equation (1) gives DVS:

Correct eq.: \( DVS = QAO - VBD - QVO \),
Erroneous eq.: \( DVS = QAO + VBD + QVO \). \( (1) \)

The second error is an algebraic loop in the nonmuscle oxygen delivery subsystem (Fig. 2b). There is a wrong feedback connection in venous oxygen saturation (OSV), which would cause a constant rise of OSV and the model would rapidly become unstable. Equation (2)
gives the OSV from the blood flow in non-renal, non-muscle tissues (BFN), oxygen volume in aortic blood (OVA), rate of oxygen delivery to non-muscle cells (DOB) and hematocrit (HM).

Correct eq.: 
\[
\frac{d(OSV)}{dt} = \left(\frac{BFN \times OVA - DOB}{BFN \times HM \times 5} - OSV\right) / Z7,
\]

Erroneous eq.: 
\[
\frac{d(OSV)}{dt} = \frac{BFN \times OVA - DOB}{BFN \times HM \times 5} - \frac{BFN \times OVA - DOB}{BFN \times HM \times 5 \times Z7}.
\]

(2)

Errors 3 and 4 involve simple subsystem red cells and viscosity. The third one is caused by positive feedback in the volume of red blood cells (VRC) computation (see Fig. 2c). Equation (3) gives the VRC from the red cell mass production rate (RC1) and rate factor for red cells destruction (RCK) where the product between VRC and RCK gives the red cell mass destruction rate,

Correct eq.: 
\[
\frac{d(VRC)}{dt} = RC1 - VRC \times RCK,
\]

Erroneous eq.: 
\[
\frac{d(VRC)}{dt} = RC1 + VRC \times RCK.
\]

(3)

The fourth error is caused by a missing negative feedback in the portion of blood viscosity caused by red blood cells (VIE) computation (Fig. 2d). VIE is computed from the output of integrator HM2 (HM after integration divided by the normalization parameter HKM). Without the negative feedback, HM2 would incessantly rise. Viscosity is proportionate to hematocrit and the integrator acts as a dampening element in the original Guyton’s model. From experimental data it can be derived that dependence of blood viscosity on hematocrit is not linearly proportional (Guyton et al. 1973). In equation (4), we designed a negative feedback by adding HMK constant into the feedback and by changing the HKM normalization parameter, which caused stabilized behavior of HM2,

Correct eq.: 
\[
\frac{d(HM2)}{dt} = HM - \frac{HM2}{HMK},
\]

Erroneous eq.: 
\[
\frac{d(HM2)}{dt} = HM.
\]

(4)

The fifth error is in the antidiuretic hormone control subsystem. The problem is in normalized antidiuretic hormone control computation (AHC) and computation of normalized rate of antidiuretic hormone creation (AH \times 0.3333) (Fig. 2e), when both values have a value of 1 under normal conditions. The solution emerges from the classic compartment approach. The hormone inflows into the whole-body compartment at the rate F_I and outflows at the rate F_O. Rate of its depletion is proportional to its concentration c, where F_O = k \times c, and concentration depends on overall quantity of hormone M and on capacity of distribution area V. Equation (5) gives the quantity of hormone M in whole-body compartment, which depends on the balance between hormone inflow and outflow,

\[
\frac{dM}{dt} = F_I - \frac{kM}{V}.
\]

Provided that the capacity of distribution area V is constant, we will substitute the ratio k/V with constant k_1. Guyton calculated the concentration of hormone c_0 normalized as a ratio of current concentration c to its normal value c_{norm} = c/c_0. At invariable distribution area V, ratio of concentrations is the same as a ratio of current hormone overall quantity M to overall hormone quantity under normal conditions M_{norm} = M/c_0. When we formulate the rate of flow in a normalized way (as a ratio to normal rate), under normal conditions it holds that F_I = 1, dM_{norm}/dt = 0 and after substituting it into equation (5) we get the equation (6),

\[ 1 - k_1 M_{norm} = 0. \]

The relative concentration of hormone c_0 can be formulated as equation (7),

\[
c_0 = \frac{M}{M_{norm}} = k_1 M, \]

and after final adjustments and inserting into a differential equation (7) we arrive at

Correct eq.: 
\[
\frac{dc_0}{dt} = (F_I - c_0)k_1,
\]

Erroneous eq.: 
\[
\frac{dc_0}{dt} = F_I - c_0 k_1. \]
According to equation (8), the normalized concentration of hormone \( c_0 \) is subtracted from normalized inflow of hormone \( F_I \). In original Guyton’s diagram, the normalized concentration of aldosterone and angiotensin is calculated this way, which means that normalized rate of inflows is \( F_I = AH \times 0.3333 \) and normalized concentration of hormone is \( c_0 = AHC \). As a result, AHC is represented by equation (9) instead of equation (8) in original Guyton’s diagram. Equation (10) gives the final relation of AHC represented in model:

Correct eq.:
\[
\frac{d(AHC)}{dt} = (AH \times 0.3333 - AHC) \times 0.14,
\]

Erroneous eq.:
\[
\frac{d(AHC)}{dt} = AH \times 0.3333 - AHC \times 0.14. \quad (10)
\]

Model under SIMULINK
SIMULINK is a block-based language for describing dynamic systems, and also works as a modeling and simulation platform (we used version 7.5.0.342 - R2007b, integrated with MATLAB, The MathWorks, Natick, MA, USA). It is an interactive and graphic environment dedicated to the multi-domain simulation of hybrid continuous/discrete systems. During simulations, model and block parameters can be modified, and signals can be easily accessed and monitored. In the model, numerical integration was performed using ‘ode13t’ (a MATLAB library) with a variable step size (maximum step size, auto; relative tolerance, 10^{-3}).

First, code operations and routines from the computer program were rendered into the SIMULINK graphical description, i.e. elementary blocks and subsystems were connected by appropriate signals and the graphic notation of the original Guyton’s diagram was kept as much as possible (Kofránek and Rusz 2007).

Second, subsystems were not treated as ‘atomic subunits’. This causes SIMULINK’s solver to treat each subsystem as a complete functioning model. Technically, the model works in continuous time and performs all physiological regulations as a complete unit (as the original graphic diagram was designed – the FORTRAN implementation of the model is characterized by a wide range of time scales in the different subsystems), which provides an advantage when designing control systems using principles of complex physiological regulation. All calculations were performed using only the original damping constants obtained from Guyton’s diagram.

Finally, to remove a lack of convergence due to oscillation and other run-time errors, the model has addressed the algebraic loops. Note that complex model behavior depends also on correct communication between all subsystems. In this case, it was essential to normalize some of the experimental set and damping constants and supervise model behavior. The complete model is available as open-source on <http://physiome.cz/guyton/>.

Model validation
In order to validate our corrected SIMULINK implementation of the Guyton’s diagram, we simulated four experiments described by Guyton et al. (1972) paper and compared the results with the in vivo data obtained in a series of six dogs (data adopted from Chau et al. 1979), and the original Guyton’s model implementation in the FORTRAN environment.

The first experiment is the simulation of hypertension in a salt-loaded, renal-deficient patient by decreasing the functional renal mass to ~ 30 % of normal and increasing the salt intake to about five times normal on day 0. This is a very fundamental experiment revealing the importance of the kidneys in blood-pressure control and their influence in the development of essential hypertension (Langston et al. 1963, Douglas et al. 1964, Coleman and Guyton 1969, Cowley and Guyton 1975). The duration of the whole experiment is 12 days.

The second benchmark experiment represents sudden severe muscle exercise and takes place over a much shorter time scale than other experiments (5 min). The exercise activity was increased to sixty times the normal resting level by setting the exercise activity-ratio with respect to activity at rest after 30 s, corresponding to an approximately 15-fold increase in the whole-body metabolic rate (in this case, the time constant for the local vascular response to metabolic activity was reduced by 1/40).

The third benchmark experiment simulates the progress of nephrotic edema by increasing seven-fold the rate of plasma-protein loss on day 1. After seven days, the rate of plasma-protein loss is reduced to three-times above the norm. The duration of the whole experiment is 12 days.

The fourth benchmark experiment simulates the atrioventricular fistula by opening the fistula on day 1 (the constant that represents fistula is set to 5 %) and closing the fistula on day 5. The duration of the whole experiment is 9 days.

The goodness-of-fit of the model was also
compared in terms of the chi-square ($\chi^2$) test between observed simulation results and predicted clinical data.

**Results**

Figure 3 represents the results of the simulation of hypertension (Experiment 1). The cardiac output rose at first to ~ 30% above normal but then was stabilized by the end of 12 days. The arterial pressure rises more slowly, requiring several days to reach high elevation. During the next days it remained at its new high level indefinitely, as long as the high salt intake was maintained. The simulation is quite sufficient to predict the available data with high statistical significances of $\chi^2 (11) = 1445$; $p<0.001$ for simulation of the arterial pressure, $\chi^2 (10) = 939$; $p<0.001$ for simulation of the heart rate, $\chi^2 (10) = 1388$; $p<0.001$ for simulation of the stroke volume, $\chi^2 (10) = 1189$; $p<0.001$ for simulation of the cardiac output, and $\chi^2 (10) = 1304$; $p<0.001$ for simulation of the total peripheral resistance.

Figure 4 presents the results of the muscle exercise simulation (Experiment 2). At the onset of exercise, cardiac output and muscle blood flow increased considerably and within a very short time. Urinary output fell to its minimal level, while arterial pressure rose moderately. Muscle cell and venous PO$_2$ fell rapidly. Muscle metabolic activity showed an instantaneous increase but then decreased considerably because of the development of a metabolic deficit in the muscles. When exercise was stopped, muscle metabolic activity fell below normal, but cardiac output, muscle blood flow and arterial pressure remained elevated for a while as the person was repaying the oxygen dept.

Figure 5 illustrates the results of the nephrosis simulation (Experiment 3). The principal effect of nephrosis consists of urine protein excretion that may or may not be associated with significant changes in other renal functions. A deficit of the total plasma protein reduces the oncotic pressure, resulting in a fluid redistribution from the blood to the interstitial compartment and an increase of the (mostly free) interstitial fluid volume. Another effect is a mild decrease of cardiac output and arterial pressure. The initial hypoproteinemia only slightly decreased both arterial pressure and cardiac output but induced a notable restriction of the urinary output. Thus, the fluid was retained in the organism causing the interstitial swelling, although the volume of the free interstitial fluid remained relatively unchanged until the interstitial fluid pressure stayed negative. After it reached positive values, an apparent edema occurred with a sharp drop in the arterial pressure. When the rate of renal loss of protein was increased to the point where the liver could increase the plasma protein level, the edema was relieved with high
diuresis and increased cardiac output by the end of 12 days.

Figure 6 shows the results of atrioventricular fistula simulation (Experiment 4). Opening the fistula caused an immediate dramatic change in cardiac output, total peripheral resistance and heart rate. Urinary output decreased to minimal threshold levels. As the body adapted, extracellular fluid volume and blood volume increased to compensate for the fistula with the result that after a few days arterial pressure, heart rate and urinary output were near normal levels, while cardiac output doubled and peripheral resistance halved. When the fistula was closed, a dramatic effect occurred with a rapid decrease in cardiac output, rapid increase in peripheral resistance, moderate increase in arterial pressure and moderate decrease in heart rate. Marked diuresis reduced the extracellular fluid volume and blood volume to normal or slightly below. After 9 days, the patient was nearly normal.

Discussion and Conclusion

The main goal of this paper is the implementation of the core circulatory dynamics model based on Guyton’s original diagram and its validation with real experimental data. It was shown how a model might furnish a physiological interpretation for the statistical results obtained on clinical data. We also used the output from Guyton’s experiments (Guyton et al., 1972) as a benchmark to validate our implementation. One such problem is the regulation of arterial blood pressure, as was well established by Guyton and his collaborators, since their quantitative systems models led them to a deep reorientation of the understanding of the causes of hypertension (Guyton et al., 1967, Guyton 1980, 1990). This was our rationale for adopting Guyton’s diagram as the initial demonstrator of the core model.

As an example of general reflection on physiological regulation, we further discussed the significant differences between the output of the last two simulations including nephrosis and atrioventricular fistula. Both experiments are associated with significant changes in kidneys functions; involving changes in urinary output, arterial pressure, cardiac output, and plasma or blood volume. In simulation of the circulatory changes in nephrosis, the seven-fold rate of plasma-protein loss caused a fast decrease of proteins volume in the plasma. Reduced oncotic pressure of proteins led to a transfer of water from plasma into interstitium, and a decrease of plasma volume which caused a decrease in arterial pressure. The decreased volume of plasma led also to a decrease of atrial pressure followed by a decrease of the cardiac output. As a result of decreased arterial pressure, vasoconstrictor effects of autonomic autoregulation caused a rapid decrease of urinary output. Reduced volume of plasma proteins lowered the intake of oncotic pressure of proteins in glomerular capillaries, and thus caused an increase in glomerular filtration and
Fig. 5. Benchmark experiment 3: simulation of circulatory dynamics in nephrosis. At day 1, the kidneys began to excrete large amount of plasma protein. As a consequence, the fall of the total circulating plasma protein occurred. When the plasma total protein fell below a critical level, an enormous increase in interstitial free fluid occurred. At the end of simulation, an increase in total plasma protein caused marked diuresis and beginning resorption of the edema. Total experiment time (x-axis) was 288 hours (12 days). Comparison of simulation results of our SIMULINK model (solid lines) with the original Guyton’s model implementation in FORTRAN (dashdot lines).

Fig. 6. Benchmark experiment 4: simulation of atrioventricular fistula. At day 1, the opening of the fistula caused an extreme increase in cardiac output, and decrease in total peripheral resistance. It remains until 5 day where the fistula was closed. At the end of the record, patient was nearly normal. Total experiment time (x-axis) was 9 days. Comparison of simulation results of our SIMULINK model (solid lines) with the original Guyton’s model implementation in FORTRAN (dashdot lines).

sequential diuresis. Continuous transfer of water from plasma into interstitium and a decrease in arterial pressure resulted in a slow decrease of diuresis into minimal threshold levels. Considering that a simulated patient could not lose more plasma proteins through the kidneys, the rate of plasma protein loss was reduced to three-fold of the norm after 7 days of the experiment. This effect was sufficient to stop the decrease and sequential increase of the concentration of plasma proteins in consequence of protein synthesis progress in the liver. Considering water accumulation in interstitium, the interstitial fluid pressure increased, a slight increase of proteins was sufficient to invert equilibrium on the capillary membrane, and water began to be resorbed from interstitium to plasma. This was associated with increased plasma volume and sequential diuresis. The results from the simulation are almost identical with those that occur in patients with nephrosis (Guyton et al. 1972, Lewis et
This includes the failure to develop sufficient amounts of edema until the protein concentration falls below a critically low level of about third of normal (Guyton et al. 1972). The simulation also shows the typical tendency of nephrotic patients to have a mild degree of circulatory collapse and slightly decreased plasma volume (Guyton et al. 1972). Another important effect is the changing level of urinary output, a feature that also occurs in nephrotic patients, with urinary output falling very low during those periods where large amounts of edema are being actively formed and the urinary output becoming great during those periods when edema is being resorbed (Guyton et al. 1972).

Similarly as the simulation of nephrosis, the simulation of the atrioventricular fistula was associated with an initial rapid decrease of urinary output. Opening the fistula caused a dramatic decrease of peripheral resistance and an immediate increase of cardiac output. This resulted in acute reaction of autonomic system which rapidly decreased glomerular filtration by increasing of renal vascular resistance, and thus practically stopped the urinary output. As a consequence of the stopped urinary output, the blood volume was increased, vasoconstrictor reaction in kidneys was subsided, and diuresis was re-established. Circulatory system dynamics shifted to its new dynamic equilibrium with increased cardiac output and blood volume, and decreased peripheral resistance. After a closure of the fistula, the whole process was reoriented. The kidneys rapidly excreted redundant blood volume and circulatory dynamics system returned to normal levels. The results from the simulation are almost identical with those that occur during clinical observation of the effects of closing and opening a fistula in animals (Frank et al. 1955). An important effect of fistula management was reported by Friesen et al. (2000). This simulation also shows the essential importance of renal blood volume control for maintenance of blood pressure.

Our circulatory dynamics model can also be used to simulate other experiments including simulations of development of general heart failure, effects of sympathetic nervous system blockade on circulatory function, effect of infusion of different types of substances, effects of vasoconstrictor agents acting on different parts of the circulation, effects of extreme reduction of renal function on circulatory function, and others. Created SIMULINK diagram involves tracking the values of physiological functions during simulation experiments and also disconnect the individual regulation circuits using switches. It allows tracking the importance of individual regulation circuits in progression of a number of various pathological conditions. As an example, in atrioventricular fistula experiment, when the AUM-parameter (sympathetic arterial effect on renal arteries) is returned to its normal value, the kidneys will not respond on increased autonomic system activity. Similarly, in the nephrotic experiment, when the PPC-parameter (plasma colloid osmotic pressure) is returned to its normal value, the kidneys will not increase diuresis in response to the decrease of plasma protein volume. The restored Guyton’s diagram has become an interactive educational aid that allows through model experiments, a better reflection of general physiological regulations in the pathogenesis of various diseases.

The result of this study is not only a complex functional model, but also a correction of the frequently published Guyton’s diagram, which still remains a landmark achievement. The model evolved over the years, but the core of the model and the basic concepts remained untouched and many of the principles contained in the original model have been incorporated by others into advanced models (Abram et al. 2007, Hester et al. 2008). The originality of our core model implementation is our commitment to provide a documentation for each basic module and continuous interactive modification and development of any aspect of the model parameters or equation and its documentation. The complex medicine simulator based on the quantitative physiological model will make it possible to simulate a number of pathological conditions on a virtual patient and the effect of using an artificial organ on normal physiological function can also be simulated. These include artificial heart, artificial ventilator, dialysis, and others.

**Conflict of Interest**

There is no conflict of interest.

**Acknowledgements**

This research was supported by the research programs “Studies at the molecular and cellular levels in normal and in selected clinically relevant pathologic states” MSM 0021620806, and “Transdisciplinary Research in Biomedical Engineering” MSM 6840770012, and by the grants “e-Golem: medical learning simulator of human physiological functions as a background of e-learning teaching of critical care medicine” MSM 2C06031, „Analysis and Modelling Biological and Speech Signals” GAČR 102/08/H008, and by the Grant Agency of the
Czech Technical University in Prague for the FORTRAN implementation of the original Guyton’s model. We are obliged to R. J. White for provision of

References


Guyton's Diagram Brought to Life - from Graphic Chart to Simulation Model for Teaching Physiology

Kofránek, Jiří; Rusz, Jan and Matoušek, Stanislav

Internetový sborník


7 stran.
GUYTON'S DIAGRAM BROUGHT TO LIFE – FROM GRAPHIC CHART TO SIMULATION MODEL FOR TEACHING PHYSIOLOGY

Jiří Kofránek, Jan Rusz, Stanislav Matoušek

Laboratory of Biocybernetics, Department of Pathophysiology, 1st Faculty of Medicine, Charles University, Prague, Czech Republic

Abstract
Thirty five years ago, A.C. Guyton et al. published a description of a large model of physiological regulation in a form of a graphic schematic diagram. The authors brought this old large-scale diagram to life using Matlab/Simulink. The original layout, connections and description of individual blocks were saved. However, contrary to the old system analysis diagram, the new one is also a functional simulation model by itself, giving the user a possibility to study behaviour of all the variables in time. Furthermore, obvious and less obvious errors and omissions were corrected in the new Simulink diagram.

1 Introduction

Prof. Arthur C. Guyton, T. G. Coleman and H. J. Grand published the article [6] in the Annual Review of Physiology magazine 35 years ago. It was a completely different form of article than usual physiological articles published until that time. Its fundament was a large scheme, which at first sight evoked some electrotechnical device, but there were computing blocks shown (multipliers, dividers, summators, integrators, functional blocks) instead of electrotechnical components. They symbolized mathematical operations, which were applied on physiological quantities. Connecting wires between blocks represented complicated feedback connections of physiological quantifiers? Blocks were divided to eighteen groups, which have represented separate physiological subsystems. The central subsystem symbolized circulation dynamics – to which other blocks were connected (kidney, tissue fluid, electrolytes, hormonal control and autonomous nervous regulation) via feedback connections.

2 Schematic Diagram Instead of Verbal Description

The article described a large-scale model of the circulatory system regulation in wider perspective: The respiratory system is integrated into other subsystems of the organism that influence its function. Instead of giving the reader a set of mathematic equations, the article uses fully equivalent graphical representation. This syntax graphically illustrates the mathematical relationships in the form of the above mentioned blocks. The description of the model was given in the form of a principal graphical chart only (which was, however, fully illustrative), explicatory comments and reasoning behind the given formulas were very brief, e.g.: “Blocks 266 through 270 calculate the effect of cell pO2, autonomic stimulation, and basic rate of oxygen consumption by the tissues on the actual rate of oxygen consumption by the tissues.” Such a formulation required full concentration, as well as some physiological and mathematical knowledge for the reader to understand the meaning of the formalized relationships between the physiological entities. Later, in 1973, and in 1975 Arthur C. Guyton published monographs [7,8], where he explained most of the concepts in more length.

Guyton’s model represents the first large-scale mathematical description of the body's interconnected subsystems and their functioning. It was indeed a turning point – the impetus to start
the research in the field known as integrative physiology. Using the system analysis of the physiological regulation, the model was for the first time in history able to depict the simultaneous dynamics of the circulatory, excretory, respiratory and homeostatic regulation.

The group of A. C. Guyton kept upgrading and extending the model later on, and upon request they even provided the FORTRAN source code of the model realization to the ones interested. In 1982, the “Human” model appeared [4], representing yet another milestone in the simulation model development. It gave the possibility to simulate a number of pathological conditions on a virtual patient (cardiac, respiratory, kidney failure, etc.) and the therapeutic influence of various drugs, infusions of electrolytes, blood transfusion, etc. Furthermore, the effect of the artificial organ use on normal physiological functions could have been simulated (artificial heart, artificial ventilator, dialysis, etc.). Its current interactive web implementation is available from this address http://venus.skidmore.edu/human.

The latest work results from Guyton’s colleagues and students are Quantitative Circulatory Physiology and Quantitative Human Physiology simulators [1]. Models can be downloaded from this address http://physiology.unc.edu/themodelingworkshop/.

3 Pioneer of the Systemic Approach in Physiology

Arthur C. Guyton (Fig. 1) was among the pioneers of system analysis in the inquiry of physiological regulation. He introduced many fundamental concepts regarding short and long time regulation of the circulation and its connection with the regulation of circulating volume, osmolarity and ionic composition of bodily fluids. He worked up a great many original experimental procedures – for instance, he was the first one to measure the value of pressure in the interstitial fluid. However, he was not only an innovative experimenter, but also a brilliant analyst and creative synthesizer. He was able to draw out new conclusions for the dynamics of processes in the body from the experimental
results and thus explain the physiological basis of a number of regulatory processes in the organism as a whole. Guyton’s research has shown, for example, that it is not only the heart as a pump that controls the cardiac output; but that an equally important roles are played by the regulation of tissue perfusion, dependent on the oxygen supply, as well as on the filling of the vessels and the compliance of great veins. It was A.C. Guyton who proved that the long-term regulation of blood pressure is done by kidneys [9].

When you study the dynamism of regulatory processes, verbal description and common sense are often not sufficient. Prof. Guyton realized this already in the mid sixties, when he studied the factors influencing blood pressure. Hence, he has searched a more exact way of expressing relationships; first using connected graphs and finally also computer models. He created his first computer models, together with his long-term colleague Thomas Coleman, in 1966. As an erudite physiologist and a hand-minded person at the same time, he was engaged in biomedical engineering in times, when this specialization did not yet officially exist.

Remarkably, Guyton did not intend to engage in theoretical medicine at first. His original aim was to work in the clinical field. After he graduated from Harvard University in 1943, he began his surgical internship at Massachusetts General Hospital. His surgical carrier was interrupted by war. He was called into the Navy. However, he worked in bacteriological warfare research during most of this period. After the war, he returned to the surgery, but only for a short while. In 1946, overworked, he suffered a bout of poliomyelitis that left serious consequences – paralysis of the left arm and leg had bound him either to a wheelchair or crutches for the rest of his life. However, his creative spirit did not leave him in this period of hardship, and he invented an electric wheelchair controlled by “a joystick”, as well as a special hoist for easy transfer of disabled people from bed to the wheelchair. Later, he received a Presidential Citation for his invention. The physical handicap ended Guyton’s carrier in cardiac surgery and steered him into the theoretical research. In spite of having job offers at Harvard University, he returned back to his hometown Oxford, Mississippi, where he first taught pharmacology at a two-year medical school; however, not long after that, he became head of the Department of Physiology at The University of Mississippi. He established a world famous physiological school in what used to be a rather provincial institute (on an American scale). Here, he wrote his world-famous textbook of physiology, originally a monograph that has seen its eleventh edition already, as well as more than 600 articles and 40 other books. He has trained many generations of medical students and more than 150 Ph.D. students. In 1989, he passed on the leadership of the institute to his disciple J.E. Hall and as a professor emeritus devoted himself to research and teaching. He died tragically in an automobile accident in 2003.

4 Fixing Errors in Guyton's Chart

Guyton was among the first proponents of the formalized description of physiological reality. Formalization means converting a purely verbal description of a relevant array of relationships into a description in the formalized language of mathematics. Guyton's diagram from 1972 (Fig 2) is a formalized description of results of one of the first significant systemic analysis of physiological functions.

Graphic notation for description of quantitative and structural relations in physiological systems suggested by Guyton was adopted by other authors in the seventies and eighties. For example, in 1977 [2] they used a slightly modified Guyton's notation in their monograph, covering the system analysis of interconnection between physiological regulation systems, [11] formulated, in Guyton's notation, their model of overall regulation of body fluids, etc.

Later on, means of simulation development tools were used for graphic notation of the structure of physiological regulation relations, for example; Simulink by the Mathworks company or open source free software package for teaching physiological modelling and research JSIM [16, 17] (see http://www.physiome.org/jsim/), or recently, graphic means of expression of simulation language Modelica [5].

Simulink diagrams are very similar to the thirty five year old notation used in the original model of A.C.Guyton. Therefore we decided to revive the old model by means of a modern software instrument. We tried to keep the resemblance identical as it was in the original pictorial diagram - the layout, the disposition of wires and the quantity labels are the same.
The realization of the old diagram is not as smooth as it might seem at first sight, because, there are errors and omissions in the original scheme. In a hand-drawn picture, it does not matter so much, because the overall meaning is still valid (most of the errors are present just on the paper, not in the original FORTRAN implementation). However, if we try to bring the model to life in Simulink, the errors show up. The model either behaves inadequately or even becomes unstable, values start to oscillate and the model complex collapses. There were a few errors – changed signs, a multiplier instead of a divider, changed connection between blocks, missing decimal point, wrong initial conditions, etc. – but it was enough for a wrong functioning of the model. Being acquainted with physiology and system analysis, we could have avoided the mistakes with a little effort.

An easily detectable error in the diagram is, for instance, wrong marking of flow direction in the summation block no. 5 in subsystem Circulatory Dynamics (Fig 3). It is obvious, that the rate of increase in systemic venous vascular blood volume (DVS) is the subtraction (not the summation) between all rates of inflows and rates of outflows. Inflow is a blood flow from systemic arterial system – its rate is denoted as QAO, outflow rate from systemic veins is a blood flow rate from veins into the right atrium (QVO). The rate change of filling of the vascular system as the blood volume changes (VBD) is calculated from the difference between summation of overall capacity of vascular blood compartments and blood volume – therefore VBD is the outflow rate and not the inflow rate, and in the summator it must have a negative sign.

In subsystem Non-Muscle Oxygen Delivery, there is a wrong depiction of connection in integrative block no. 260 (Fig.4). If the model was programmed exactly as depicted in the original
diagram, the value of non-muscle venous oxygen saturation (OSV) would constantly rise and the model would become unstable very quickly. Besides, there would be an algebraic loop in the model. Correction is simple, input to summator no. 258 is the value of OSV, and therefore it is sufficient to move feedback input to summator behind the integrator as it is indicated in the picture.

Small and simple subsystem Red Cells and Viscosity includes two errors (Fig 5). The first is visible at first sight. It is obvious that the rate of change of red cell mass (RCD) is the subtraction (not the summation) between red cell mass production rate (RC1) and red cell mass destruction rate (RC2). The second error is obvious as well. During calculation of a portion of the blood viscosity caused by red blood cells (VIE) from value of hematocrit (HK) according to the diagram, the viscosity would have to constantly rise, because the value of quantity HM2 would incessantly rise (HK is the input to the integrator). According to the diagram, the value of a variable HM2 is equal to 1600 - in a stable situation and under normal conditions. If we divide this value by a constant parameter HKM (=0.000920), we should arrive at a normal value VIE. Normal value VIE should be 1.5 (formulated as a ratio to viscosity of water). We can find out, by simple calculation, that it is not so, and we will arrive at the correct calculation if we multiply the value HM2 by constant HKM instead of using division. Thus it is obvious, that block no. 337 should be a multiplier unit and not a dividing unit. In order to have the value of a variable HM2 in stable situation constant (and under normal conditions equal to value 1600), the input to integrator must have zero value (block no. 336). Therefore, it is apparent, that the depiction of feedback has been omitted in the diagram. The corrected diagram is shown in picture 6 as "Corrected (A)". Viscosity is proportionate to hematocrit and the integrator acts here as a dampening element. It can be from the experimental data that dependence of viscosity of blood on hematocrit is not linear proportionate [7]. Therefore in a later realization of the model (according to source text in Fortran language) the relation between hematocrit (HK) and portion of blood viscosity was caused by red blood cells (VIE) formulated as follows:

\[
VIE = \frac{HM}{(HKM - HM)HKM}
\]
Figure 6: The error in the Antidiuretic Hormone Control subsystem.

Where:

\[ HMK = 90 \]

and

\[ HKM = 5.3333 \]

which is shown in a diagram in fig. 6, marked as Corrected (B). If we compare this picture with the original chart, the integrator no. 336 is replaced with dividing and summator units (and the value of HKM constant is quite different). Maybe, this exact structure should have been originally drawn in the original diagram, and (by mistake, the integrator was drawn instead of divider and summator) a label HKM on the left side next to integrator no. 336 indicates this situation.

An error in Antidiuretic Hormone Control subsystem is not visible at first sight (Fig. 6). According to graphic diagram, the following should hold true:

During stable conditions, according to data on the graphic diagram under normal conditions, the values should be: :

\[ 0.333 \cdot AH = 1, \]
\[ AHC = 1. \]

Then the integrator 185 will have no zero value and the system will not be in stable condition.

Where is the error?

\[ AH \cdot 0.3333 \] is a normalized rate of antidiuretic hormone creation (ratio of current rate of creation according to the norm). \[ AHC \] is a normalized concentration of this hormone (according to the norm). How is the normalized concentration of substance from normalized rate of substance creation calculated? The classic compartment approach will answer our question.

In subsystems of conducting ADH creation, aldosterone and angiotensin are calculated in the model from the rate of hormone inflow (normalized as a relative number according to the norm) and hormone concentration (again normalized as a relative number according to the norm).
We come out of a simple compartment approach - into a whole-body compartment inflow the hormone at the rate $F_i$ (it is synthesised) and outflows at the rate $F_o$. Quantity of hormone $M$ in whole-body compartment depends on the balance between inflow and outflow of the hormone.

$$F_i - F_o = \frac{dM}{dt}. \quad (2)$$

Rate of depletion of hormone $F_o$ is proportional to its concentration $c$:

$$F_o = k \cdot c. \quad (3)$$

Concentration of hormone $c$ depends on overall quantity of hormone $M$ and on the capacity of distribution area $V$:

$$c = \frac{M}{V}. \quad (4)$$

Thus after inserting:

$$F_i = \frac{k \cdot M}{V} = \frac{dM}{dt}. \quad (5)$$

Provided that the capacity of distribution area $V$ is constant, we will substitute ratio $k/V$ for constant $k_1$:

$$k_1 = \frac{k}{V}. \quad (6)$$

We arrive at:

$$F_i - k_1 \cdot M = \frac{dM}{dt}. \quad (7)$$

In the model, Guyton calculated the concentration of hormone $c_0$ normalized as a ratio of current concentration $c$ to its normal value $c_{norm}$:

$$c_0 = \frac{c}{c_{norm}}. \quad (8)$$

At invariable distribution area $V$ ratio of concentrations is the same as a ratio of current overall quantity of hormone $M$ to overall quantity of hormone under normal conditions $M_{norm}$:

$$c_0 = \frac{c}{c_{norm}} = \frac{M}{M_{norm}}. \quad (9)$$

If we formulate the rate of flows in a normalized way (as a ratio to normal rate), then under normal conditions:

$$F_i = 1,$$

$$\frac{dM_{norm}}{dt} = 0.$$ 

Therefore:

$$1 - k_1 \cdot M_{norm} = 0. \quad (10)$$

Normal quantity of hormone $M_{norm}$ will be:

$$M_{norm} = \frac{1}{k_1}. \quad (11)$$

Hence, the relative concentration of hormone $c_0$ can be formulated:

$$c_0 = \frac{M}{M_{norm}} = k_1 \cdot M. \quad (12)$$

Thus:

$$M = \frac{c_0}{k_1}. \quad (13)$$
After inserting into differential equation we arrive at:

\[ F_i - k_0 \frac{c_0}{k_1} = \frac{dc_0}{dt} \],

i.e.:

\[ F_i - c_0 = \left( \frac{1}{k_1} \right) \frac{dc_0}{dt} \].

Thus:

\[ (F_i - c_0)k_1 = \frac{dc_0}{dt} \].

According to this equation, the normalized concentration of the hormone \( c_0 \) is calculated from the normalized inflow of the hormone \( F_i \). In original Guyton's chart, the normalized concentration of aldosterone and angiotensin is calculated in this way. In case of ADH, there is an error in the chart.

The normalized rate of inflow in case of ADH:

\[ F_i = 0.3333 \, AH \].

The normalized concentration of the hormone is:

\[ c_0 = AH/HC \].

Coefficient \( k_1 = 0.14 \).

Instead of \( (F_i - c_0)k_1 = \frac{dc_0}{dt} \), there is a graphic representation of relation \( F_i - c_0 k_1 = \frac{dc_0}{dt} \).

Correct relation in case of ADH should be:

\[ (0.3333 \, AH - AH/HC)0.14 = \frac{dAH/HC}{dt} \].

This relation corresponds to a correct part of diagram shown in fig. 6.

Quoted examples of errors in the original graphic depiction of Guyton's model do not mean at all that the actual implementation of the model did include the above-mentioned errors. The model was implemented in Fortran language and it functioned flawlessly. What was incorrect was only the graphic depiction of the mathematical relations that did not correspond to the model.

If somebody implemented the model exactly according to the depiction, without thinking over and understanding the meaning of mathematical relations between physiological quantities, then such a model would not function correctly on a computer.

It is interesting, that this complicated schematic diagram was many times overprinted in several publications and nobody made an effort to fix these errors. After all, at the time, when picture schemes were created, no appropriate application had existed yet – pictures arose like a complicated drawing – and to handle redoing such a complicated drawing isn’t so easy. Maybe the authors didn't even want to correct the errors – the ones who took the pain over the analysis of the model easily uncovered the diagrams mistakes, the ones who just wanted to blindly copy, failed.

After all, at that time, the authors even used to send round the program source files in Fortran language, so if somebody wanted to just test the behaviour of the model, s/he did not have to program anything (at the most they had to routinely convert the Fortran program into other programming languages).

5 Results

After the correction of errors in the original Guyton's chart, we realized its Simulink implementation. In the Simulink diagram, we tried to maintain the same distribution of all the individual elements, as in the original diagram.
The only difference is in the graphic shapes of the individual elements - e.g. in Simulink, the multiplier/divider is represented as a square unlike the "piggy" symbol in Guyton's notation (See Fig. 7). The integrator does not have the sign of integral on itself but the expression "\(1/s\)" (being related to the transcription of Laplace transformation).

In the Simulink model, we also used switches, by which we could couple or uncouple individual subsystems and control loops.

Resultant chart of Simulink model is depicted in Fig 8.

We can transform individual physiological subsystems of the model into the form of Simulink subsystems. The graphic chart of the whole model looks rather better arranged (Fig. 9). Then the diagram of the model resembles the interconnected network of electronic chips – instead of electric signals; however, there is a flow of information in individual conductors - data of the model.

Physiological subsystems are represented by "simulation chips" – conductors with input data are connected to their individual input, and signals with information about the value of individual physiological quantities are distributed from their output "pins" to other "simulation chips".

Models formulated through the network of "simulation chips" are also the appropriate tools for team collaboration between branches of study [13]. Such a chart is much more legible also for an experimental physiologist who does not have to understand the complicated mathematical structure of a computational network inside a "simulation chip", however s/he understands the structure and functions of physiological relations. S/he can study the behaviour of a model in individual simulation chips on virtual displays and oscilloscopes, which are standard components of the Simulink environment.

In fig. 10, there is a Simulink implementation of a Guyton-Coleman model from 1986, formulated with the help of interconnected "simulation chips". When the reader compares it to a previous picture, s/he can imagine how the model has expanded in the past 14 years.

Figure 7: The pictorial block scheme of the original A.C. Guyton's model on the left and the model block diagram in the Simulink software tool. Analogically positioned and numbered blocks represent the same mathematical operations. Multipliers and dividers: blocks 255, 257, 259, 261, 263, 268, 272, 270; sum blocks: 256, 258, 262, 264, 266, 269; integrator blocks: 260 a 271; function blocks (cubic function): 265 a 267; high level saturation: between blocks 272 and 286, low level saturation: between blocks 265 and 180. The switches can either be set to receive the input values from other subsystems, or directly from the user, thus disconnecting the block from the rest of the model.
Figure 8: Guyton's overall regulation model of Circulation - implementation in Matlab/Simulink. The layout and block numbering is exactly the same as in the original Guyton's scheme (Fig. 2). The difference is, that this scheme is also a fully functional simulation model.
Simulink implementation of the (corrected) Guyton's model made by us is available for download from the address [http://physiome.cz/Guyton](http://physiome.cz/Guyton) to anyone interested. At the same address, our Simulink implementation of a much complex sequel of the model from 1986 can be found too.

Further, there is also a detailed description of all mathematical relations used with their reasoning (however, for the present time it is in the Czech language only).

6 From Simulink Diagram to Simulation Games During Physiological Teaching

We use Simulink implementation of Guyton's model as an educational tool to teach physiology to undergraduate and postgraduate students at the Czech Technical University (ČVUT). This structure of Simulink diagram (in a form of "simulation chips") is however, too abstract for medical students. It is ideal, if their teaching models have the form of schematic pictures to which they are accustomed, for example from the Atlas of Physiology [18]. Unlike the book, these pictures can be interactive, and
models running in the background can enable students to "play" with this physiological subsystem and monitor its response to various inputs.

Simulation models in the background of teaching programs are therefore very effective educational tools that facilitate the comprehension of complex regulation relations in the human organism and pathogenesis of their malfunction.

From the pedagogical perspective it is advantageous, according to our experiences, if we allow disconnection of individual regulation loops temporarily, and study reaction of the individual subsystems separately, which contributes to better understanding of the dynamics of physiological regulations [12].

During the creation of teaching applications with the use of simulation games, it is necessary, on one hand, to resolve the creation of the simulation model, and on the other hand the creation of our own simulator. These are two different tasks, whose effective solution facilitates the use of various developmental tools [14].

During the creation of simulation models it is advantageous to use developmental tools designated for creating and identification of simulation models – for example Matlab/Simulink from the Mathworks company. In this environment, we have also created a special library of physiological models - Physiology Blockset for Matlab/Simulink, open source software library. 1st Faculty of Medicine, Charles University, Prague, available at http://physiome.cz/simchips.

Creating simulation models is closely related to issues of creating formalized description of biological reality, which is the content of the worldwide PHYSIOME project [3, 10].

Creating our own teaching simulators is done in the environment of classic developmental tools for computer programmers (for example Microsoft Visual Studio, etc.) and tools facilitating the creation of interactive animated pictures, used in user interface of teaching programs (for example Adobe Flash, Adobe Flex). The future probably lies in simulators available on the web and on the accessibility of e-learning educational environment [15, 19]..
7 From Simulation Games to Medical Simulators

Thirty five years ago, when A. C. Guyton et al. published his large-scale model, the only possibility to study the behaviour of the model was on large computers that often occupied an entire room. Nowadays it is possible to run even very sophisticated models on a PC. Moreover, today’s technology allows us to add on a graphical attractive user-friendly interface to these models.

From the technological standpoint there are no obstacles that would prevent PCs from running learning simulators for practicing medical decision-making. The basis for a pilot’s simulator during training pilots is the model of the plane. Similarly, one of the prerequisites when creating a medical simulator is the extensive simulation model of a human organism. This simulation model must include all significant physiological subsystems – circulation, respiration, kidney function, water, osmotic and electrolyte homeostasis, acid-base regulation, etc. – which have to be interconnected into the model. Therefore, now is the time of a renaissance in the formation of large integral models of human organism, and of the concept of integrative physiology, that Guyton came up with years ago. At the present time the practical fulfillment is being achieved.

For example, at the present time, Thomas Coleman, one of the co-authors of the legendary article by prof. Guyton from 1972 [6], together with Guyton’s disciples, created a simulator Quantitative Human Physiology (QHP), whose theoretical basis is a new mathematical model of integrative human physiology which contains more than 4000 variables of biological interactions. A review edition of this simulator is freely available for download at http://physiology.umc.edu/themodelingworkshop/.

The simulator consists of two software packages.

The first is the equation solver, named QHP 2007.EXE. This is the executable file, prepared for the Windows operating system (2000, xp, Vista).

The second is an XML document that defines the model, the solution control and the display of results. This document is distributed over a large number of small files in the main folder and several subfolders. The XML schema used is described in a preliminary fashion in another section of this modeling workshop.

The XML document is parsed at program startup. Parsing progress is displayed in the status bar at the bottom of the program’s main window.

All of the XML files are both machine and human readable. You only need a text editor (such as Notepad, WordPad).

Unfortunately, the orientation in the structure of such a large model is difficult, due to a large number of variables (more than 4000).

Standardized notation of the model structure in XML is easily understandable for the machine, but for a human it is necessary to provide a graphic depiction of the structure of the physiological regulation relations.

Thus, the suggestions that prof. Guyton et al. sparked, thirty five years ago, by his legendary article (the concept of integrative physiology, the creation of large-scale models of physiological subsystems interconnected in an integrative way, and an effort to graphically depict the structure of physiological regulation relations), nowadays return in a new form and with new possibilities.

References

Acknowledgement

This research was supported by MŠMT aid grant No. 2C06031 and BAJT servis s.r.o company.

Jiří Kofránek, M.D., Ph.D.,
Laboratory of Biocybernetics,
Dept. of Pathophysiology,
1st Faculty of Medicine,
Charles University, Prague
U Nemocnice 5, 128 53 Praha 2,
Czech Republic
e-mail: kofranek@email.cz
Základní struktura matematického modelu fyziologických funkcí člověka (modifikace modelu A. C. Guytona a spol. z roku 1986 ve formě “simulačních čipů”, podrobný popis rovnic simulačního modelu a odladěná schemata v Simulinku)

Kofránek, Jiří

Publikováno v elektronickém sborníku konference Technical Computing Prague 2007 (http://dsp.vscht.cz/konference_matlab/MATLAB07/) jako příloha k práci:

*Kofránek, J., Rusz, J., Matoušek, S.: Guyton’s Diagram Brought to Life - from Graphic Chart to Simulation Model for Teaching Physiology*
Základní struktura matematického modelu fyziologických funkcí člověka

(modifikace modelu A.C.Guytona z roku 1986 ve formě simulačních čipů)

podrobný popis rovnic simulačního modelu
a odladěná schémata modelu v Simulinku

Všechny jednotlivé bloky a rovnice modelu jsou rozebrány na následujících stránkách.
MODUL HEMODYNAMIKA

Základní smyčka náplni, průtoku a tlaků v jednotlivých částech krevního řečiště

Krevní řečiště je rozděleno do pěti objemových kompartmentů – aorta a velké artérie (VAS), systémové vény (VVS), pravá síň (VRA), plicní artérie (VPA), plicní žíly a levá síň (VLA).

Při změně objemu krve (VB) z vnějšího části modulu krevního řečiště – se tato změna (VBD), počítaná v každém časovém kroku jako rozdíl mezi aktuální hodnotou objemu krve (přicházející jako vnější vstup do modulu) a součtem objemů krve ve všech pěti částech krevního řečiště:
VBD = VAS-VVS-VRA-VPA-VLA \quad (HD \ 01)

rozdistribuuje mezi VAS, VVS, VRA, VPA a VLA v následujících poměrech:

\[
\begin{align*}
DVBD\_VAS &= 0.261 \cdot VBD \\
DVBD\_VVS &= 0.3986 \cdot VBD \\
DVBD\_VRA &= 0.0574 \cdot VBD \\
DVBD\_VPA &= 0.155 \cdot VBD \\
DVBD\_VLA &= 0.128 \cdot VBD
\end{align*}
\] \quad (HD \ 02-06)

Objem v aortě a velkých artériích (VAS) závisí na rozdílu (DAS) mezi přitokem krve z levého srdce – QLO [l/min] a odtokem krve z aorty a velkých artérií - QAO [l/min] a rychlosti změny objemu krve z vnějšího cívního systémových rozdistrubuovanou na arteriální část krevního řečiště (DVBD\_VAS):

\[
DAS = QLO - QAO \quad (HD \ 07)
\]

\[
VAS = \int (DAS + DVBD\_VAS) \, dt \quad (HD \ 08)
\]

Objem krve, který napíná aortu a velké artérie (VAE) je rozdíl celkového objemu krve v artériích (VAS) a objemu naplňujícím aortu a velké artérie při nulovém tlaku (VAS0)

\[
VAE = VAS - VAS0 \quad (HD \ 09)
\]

Střední arteriální tlak je pak úměrný objemu napínajícím aortu a nepřímo úměrný poddajnosti arteriálního systémového řečiště:

\[
PA = VAE / CAS \quad (HD \ 10)
\]

Tlakový arterio-venózní gradient v systémové cirkulaci (PGS) je rozdíl mezi tlakem v aortě (PA) a středním tlakem ve velkých systémových žilách (PVS):

\[
PGS = PA - PVS \quad (HD \ 11)
\]

Tok krve zkrze nesvalové části krevního řečiště s výjimkou ledvin (BFN) se počítá z tlakového gradientu v systémovém řečišti mezi aortou a tlakem ve velkých žilách a odporu nesvalové části krevního řečiště (nezahrnujících ledviny) – RSN:

\[
BFN = PGS / RSN \quad (HD \ 12)
\]

Průtok krve svalovinou (BFM) se rovná arterio-venóznímu tlakovému gradientu v systémové cirkulaci a rezistencí v krevního řečiště ve svalech (RSM):

\[
BFM = PGS / RSM \quad (HD \ 13)
\]

Celková renální rezistence (RR) se počítá v modulu "ledvina" a v modulu hemodynamiky je vstupním parametrem. Renální průtok (RBF) vypočítáme z gradientu v mezi arteriálním a systémovým žilním tlakem (PGS) a renální rezistencí (RR).

\[
RBF = PGS / RR \quad (HD \ 14)
\]

Rychlost výtoku krve z arteriálního řečiště se rovná toku krve svalovinou (BFM), toku krve ledvinami (RBF), toku krve zbylou nesvalovou části krevního řečiště nezahrnujícího ledviny (BFN), a případně i průtoku krve uměle vytvořenými arteriovenózními spojky (FISFLO):
\[ QAO = BFM + BFN + RBF + FISFLO \]  
(HD 15)

Průtok umělým arteriovenózním zkratem je přímo úměrný tlakovému gradientu mezi arterií (PA) a pravou sníčí (PRA), a vodivosti zkratu (FIS).

\[ FISFLO = (PA - PRA) \times FIS \]  
(HD 16)

Objem v systémových žilách - VVS závisí na rozdílu (DVS) mezi přítokem krve do systémových žil (QAO) a odtokem krve do pravé sníčí (QVO) a rychlosti změny objemu krve z vnějšího systému rozdistribuovanou na venózní část krevního řečiště (DVBD_VVS):

\[ DVS = QAO - QVO \]  
(HD 17)

\[ VVS = \int (DVS + DVBD_VVS) \, dt \]  
(HD 18)

Objem krve, napínající venózní systémové řečiště VVE je počítán jako rozdíl aktuální hodnoty objemu krve ve velkých žilách (VVS) a hodnoty maximálního objemu krve ve venózním řečišti při nulovém tlaku, tj. takového objemu při kterém se naplňují žíly ale nestoupá tlak (VVS0):

\[ VVE = VVS - VVS0 \]  
(HD 19)

Hodnota maximální náplně venózního řečiště, které nemění tlak (VVS0) je součet několika objemů – rezidualního objemu, nezávislého na působení ostatních faktorů (VVR), objemu měnícího se díky tahové relaxaci svaloviny žil způsobené tlakem v žilní stěně (VV6 a VV7), dodatečného objemu vzniklého reflexně zpětnovazebnou relaxací tahovými receptory ze sníčí (ATRRVFB – když se rozepnou sníčí, roztáhnou se i velké žíly) a nárůstu objemu díky působení angiotenzinu – (VV_ANU), kde ANU je nerenální účinek angiotenzinu, vyjádřený jako relativní hodnota vzhledem k normální hladině angiotenzinu, a ANY je citlivost velkých žil na angiotenzin:

\[ VV_ANU = (ANU - 1) \times ANY \]  
(HD 20)

\[ VVS0 = VVR + VV_ANU + VV6 + VV7 + ATRVFB \]  
(HD 21)

Průměrný tlak ve venózním systému (PVS) je úměrný objemu napínajícímu venózní řečiště (VVE) a nepřímý úměrný poddajností venózního systémového řečiště (CV):

\[ PVS = VVE / CV \]  
(HD 22)

Tlakový gradient pro tok krve z venózního systému do pravé sníčí (PGV) se rovná rozdílu mezi průměrným tlakem v systémovém žilním řečišti (PVS) a odtokovým tlakem ve velkých žilách v hrudníku (PR1):

\[ PGV = PVS - PR1 \]  
(HD 23)

Rezistence průtoku krve zkze žilního systému (RVG) se rovná konstantní hodnotě, závislé na viskozitě krve (VIM) dělené průměrným tlakem v žilách (PVS):

\[ RVG = 2.738 \times VIM / PVS \]  
(HD 24)

Rychlost toku krve z venózního řečiště do pravého srdce (QVO) se rovná tlakovému gradientu v žilním systému (PGV) děleném rezistenci toku krve zkze žilního systému (RVG):

\[ QVO = PGV / RVG \]  
(HD 25)

1 Tyto závislosti jsou realizovány v bloku "Unstressed venous volume"
Objem krve v pravé síně - VRA závisí na rozdílu (DRA) mezi přítokem krve z velkých žil do pravé síně (QVO) a odtokem krve (QVO) a rychlostí změny objemu krve z vnějšího cévního systému rozdistribuovanou na část krevního řečiště v pravé síně (DVBD_VRA):

\[
DRA = QVO - QRO \quad \text{(HD 26)}
\]

\[
VRA = \int (DRA + DVBD_VRA) \, dt \quad \text{(HD 27)}
\]

Objem krve, napínající krevní řečiště v pravé síně (VRE) je počítán jako rozdíl okamžité aktuální hodnoty objemu krve v pravé síně (VRA) po odečtení konstantní hodnoty objemu krve v pravé síně za podmínek konstantní hodnoty reprezentující reziduální objem krve v pravé síně v případě nulového tlaku (VRA0=0.1 litru):

\[
VRE = VRA - VRA0 \quad \text{(HD 28)}
\]

Tlak v pravé síně (PRA) se rovná objemu krve napínající krevní řečiště v pravé síně (VRE) dělené konstantou představující poddajnost pravé síně (CRA=0.005 torr/l):

\[
PRA = \frac{VRE}{CRA} \quad \text{(HD 29)}
\]

Výpočet tlaku ve velkých oddílovcích žílách v hrudníku (PR1) je využíván při výpočtu odtoku krve ze žilního systému (viz rovnice HD 23-25). Tento tlak se rovná tlaku v pravé síně (PRA). Pokud je ale tlak pravé síně je negativní, velké žíly v hrudníku kolabují a tlak v nich proto nepoklesne pod určitou minimální hodnotu (PR1LL):

když \( PRA > PR1LL \), pak: \( PR1 = PRA \) \quad \text{(HD 30a)}

v opačném případě: \( PR1 = PRLL \) \quad \text{(HD 30b)}

Závislost mezi odtokem krve z pravého srdce (QRN) na tlaku v pravé síně (PRA) u normalizovaného zdravého srdce (Starlingova křivka) je aproximována proložením uzlových bodů následující tabulky:\(^2\):

\[
QRN = \text{function StarlingRNorm(PRA)} \quad \text{(HD 31)}
\]

<table>
<thead>
<tr>
<th>PRA [mmHg]</th>
<th>QRN [l/min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; -6</td>
<td>0</td>
</tr>
<tr>
<td>-6</td>
<td>0</td>
</tr>
<tr>
<td>-3</td>
<td>0.75</td>
</tr>
<tr>
<td>-1</td>
<td>2.6</td>
</tr>
<tr>
<td>0</td>
<td>5.0</td>
</tr>
<tr>
<td>2</td>
<td>9.8</td>
</tr>
<tr>
<td>4</td>
<td>12.1</td>
</tr>
<tr>
<td>8</td>
<td>13.5</td>
</tr>
<tr>
<td>&gt; 8</td>
<td>13.5</td>
</tr>
</tbody>
</table>

\(^2\) Do budoucna bude lépe definovat Starlingovu křivku jako závislost StrokeWork na preloadu /eliminace vlivu afterloadu/.
Výpočet odtoku krve z pravého srdce (QRO) na základě faktoru účinnosti pumpy (HPEF) a odtoku z normalizovaného zdravého srdce (QRN):

\[ QRO = QRN \cdot HPEF \] (HD 32)

Objem krve v plicním arteriálním řečišti - VPA závisí na rozdílu (DPA) mezi přitokem krve z pravého srdce (QRO) a odtokem krve z plicního arteriálního řečiště do plicního venózního řečiště (QPO) a rychlosti změny objemu krve z vnějšího cévního systému rozdistribuovanou na část plicního krevního řečiště (DVBD_VPA):

\[ DPA = QRO - QPO \] (HD 33)

\[ VPA = \int (DPA + DVBD_VPA) \, dt \] (HD 34)

Objem krve, napínající plicní arteriální krevní řečiště (VPE) je počítán jako rozdíl okamžité aktuální hodnoty objemu krve v plicním arteriálním řečišti (VPA) po odečtení hodnoty reziduálního objemu krve v plicním arteriálním řečišti který řečiště zcela naplňuje, ale ještě nezvyšuje tlak z nulové hodnoty (VPA0 = 0.30625 litru):

\[ VPE = VPA - VPA0 \] (HD 35)

Tlak v plicnici (PPA) se rovná objemu krve napínající arteriální plicní krevní řečiště (VPE) dělené konstantou představující poddajnost plicních artérií (CPA = 0.0048 l/torr):

\[ PPA = VPE / CPA \] (HD 36)

Tlakový gradient mezi středním tlakem v plicních artériích a tlakem v plicních žilách (PGL) počítáme jako rozdíl mezi tlakem v plicnici (PPA) a tlakem v levé síni (PLA):

\[ PGL = PPA - PLA \] (HD 37)

Odtok krve z plicního arteriálního do plicního venózního řečiště (QPO) je úmerný tlakovému gradientu mezi arteriálním a venózním plicním řečištěm (PGL) a nepřímo úmerný rezistenci v plicním oběhu (RPT):

\[ QPO = PGL / RPT \] (HD 38)

Objem krve v levé síni (a v plicních žilách) - VLA závisí na rozdílu (DLA) mezi přitokem krve z plicního arteriálního řečiště (QPO) a odtokem krve z levé síni do levé srdce a komorou do systémového arteriálního řečiště (QLO). K tomu je zapotřebí připočíst ještě rychlost změny objemu krve z vnějšího cévního systému rozdistribuovanou na část plicního krevního řečiště (DVBD_VPA):

\[ DLA = QPO - QLO \] (HD 39)

\[ VLA = \int (DLA + DVBD_VLA) \, dt \] (HD 40)

Objem krve který roztahuje a napíná levou síň (a plicní žily) – VLE je počítán jako rozdíl mezi celkovým objemem krve v levé síni a plicních žilách (VLA) a reziduálním objemem krve v plicních žilách a levé síni (VLA0 = 0.4 l), který tuto část krevního řečiště zcela zaplňuje ale ještě nezvyšuje tlak:

\[ VLE = VLA - VLA0 \] (HD 41)
Tlak v levé síní (PLA) se rovná okamžité hodnotě objemu napínajícím plicní žíly a levou sín (VLE) dělenou poddajností levé síně a plicních žíl (CLA=0.01 l/torr):

\[ PLA = \frac{VLE}{CLA} \]  

(ZD 42)

Závislost mezi odtokem krve z pravého srdce (QLN) na plnícím tlaku v levé síní (PLA) u normalizovaného zdravého srdce (Starlingova křivka) je aproximována proložením uzlových bodů následující tabulky:

\[ QLN = \text{function StarlingLNorm(PLA)} \]  

(HD 43)

<table>
<thead>
<tr>
<th>PLA [mmHg]</th>
<th>QLN [l/min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-4.5</td>
<td>0</td>
</tr>
<tr>
<td>-4</td>
<td>0.01</td>
</tr>
<tr>
<td>-1</td>
<td>3.6</td>
</tr>
<tr>
<td>3</td>
<td>9.4</td>
</tr>
<tr>
<td>6</td>
<td>11.6</td>
</tr>
<tr>
<td>10</td>
<td>13.5</td>
</tr>
<tr>
<td>&gt;10</td>
<td>13.5</td>
</tr>
</tbody>
</table>

Výtok krve z levé komory (QLO) je ovlivňován řadou faktorů kterými se násobí tok krve z levé komory normalizovaného zdravého srdce (QLN). Mezi ně patří – tlakové zatížení srdce (LVM), faktor změny účinnosti při abnormální změně stažlivosti levé komory (HSL), faktor zvyšující stažlivost díky hypertrofii levé komory (HPL), faktor snižující stažlivost myokardu při nízkém průtoku krve (HMD) a faktor měnící stažlivost díky zvýšení nebo snižení autonomní stimulace (AUH):

\[ QLO = LVM \times QLN \times AUH \times HSL \times HMD \times HPL \]  

(HD 44)

Výpočty rezistencí v systémové cirkulaci

Princip výpočtu rezistencí spočívá v zahrnutí vlivu nejvzácnějších okolností na změnu bazální hodnoty rezistence. Tyto vlivy jsou vyjádřeny jako multiplikační faktory kterými se násobí (nebo dělí) bazální hodnota rezistence3.

Péřiferní odporové systémové řečiště je v modelu uvažováno ve třech agregátech: ve svalech, v ledvinách a v ostatních měkkých tkáních. Péřiferní odpor a příslušný krevní průtok v ledvinách bude popsán dále v samostatném modulu ledvin. V modulu hemodynamiky se počítají rezistence v krevním řečišti svalů (RSM) a rezistence v ostatních "nesvalových" a "neledvinných" tkání (RSN).

3 Je vhodné se do budoucna zaměřit na otestování možnosti využití fuzzy algebry pro nahravení pouhého násobení multiplikátorů.
Zvýší-li se tlak v cévním řečišti, a neuvažujeme-li další mechanismy, pak zvýšení tluaku vede k rozšíření (distenzi) cévy a snižuje tím cévní odpor. Vliv hodnoty arteriálního tlaku (PA) na změnu rezistence díky arteriální distenzi je vyjadřován jako poměrný multiplikační faktor (PAM), kterým se dělí bazální hodnota rezistence. Normální hodnota tohoto faktoru je 1. Vliv arteriálního tlaku na tento faktor je exponenciální – počítá se z poměru aktuální hodnoty středního arteriálního tlaku (PA) k normální hodnotě (PA_norm = 100 torr) přes exponenciální faktor (PAEX=1.41):

\[ PAM = \left( \frac{PA}{PA_{\text{norm}}} \right)^{PAEX} = \left( \frac{0.01 \cdot PA}{100} \right)^{1.41} \quad \text{(HD 45)} \]

Výpočet společného kumulativního multiplikačního faktoru (R1) ovlivňujícího jak systémovou arteriální rezistenci ve svalech tak i rezistenci v krevních cévách (nezahrnujících ledviny). Jako vstupní faktory (vyjádřené jako poměr k normě) se zde uplatňují: úroveň autonomní sympatické stimulace (AUM), vliv viskozity (VIM), (nerenální) vliv angionenzinu (ANU), vliv antituiretického hormonu (AHMR) na rezistenci (proto je vasopresin synonymem pro ADH). Dalším faktorem je zpětnovazebný vliv tachových receptorů v cévách (ATRRFB), který vede k vasodilataci, a proto je multiplikační faktor ve jmenovateli stejně jako i distenční faktor vyjadřující přímé působení arteriálního tlaku na roztažení cévy (PAM):

\[ R1 = AUM \cdot VIM \cdot ANU \cdot AHMR / PAM \cdot ATRRRFB \quad \text{(HD 46)} \]

Výpočet rezistence v systémovém řečišti svalů (RSM) je počítán jako vliv multiplikačních faktorů na bazální hodnotu rezistence (RAM). K těmto faktorům patří kumulativní multiplikační faktor (R1) z rovnic HD 46 a dále multiplikační faktor vyjadřující lokální autoregulační vliv svaloviny (AMM), a konečné faktor vyjadřující myogenní autoregulaci (MYOGRS), zvýšující rezistenci při zvýšení tenze v arteriálním řečišti.

\[ RSM = R1 \cdot AMM \cdot MYOGRS \cdot RAM \quad \text{(HD 47)} \]

Systémová rezistence v "nesvalových" a "neledvinových" měkkých tkáních (RSN) se skládá ze složky arteriální, která je způsobena zvýšenou rezistenci cevního řečiště (RSNA) a ze složky venální (resp. venulózní) části rezistence (RSNV).

\[ RSN = RSNA + RSNV \quad \text{(HD 48)} \]

Rezistence v arteriální části cevního řečiště měkkých tkání (RSNA) ("nesvalových" a "neledvinových") se vypočítává na základě modifikace bazální hodnoty rezistence (RAR) v těchto tkáních kumulativním multiplikačním faktorem (R1) z rovnic HD 46 a autoregulačním vlivem mikrocirkulace v těchto tkáních (ARM):

\[ RSNA = R1 \cdot ARM \cdot RAR \quad \text{(HD 49)} \]

Pro výpočet venulózní části rezistence v "neledvinových" a "nesvalových"měkkými tkáněmi musíme nejprve znát její bazální hodnotu (RV1), která zohledňuje nastavení napětí vodního čerstvých cévních, která je závislým faktorem na stavu arteriálního řečiště a mimo jiné závisí na stavu na kapilárních arterií, na kapilárním tlaku (PC) a na bazálním systémovém venálním multiplikačním faktoru (RVSM) – za normálních okolností rovného 1.0. Poznámka: hodnota bazálního tlaku (PA) se vypočítává do jisté míry iterativně , protože pro stanovení hodnoty PC potřebujeme znát mimo jiné hladinu RVS, která je závislá na RA1, který ale závisí na RVS, která je závislá na RV1, který se vypočítá do jisté míry iterativně, a protože pro stanovení hodnoty PC potřebujeme znát mimo jiné hladinu RVS, která je závislá na RA1, který ale závisí na hodnotě PC (viz rovnicu HD 54). Tlumivý zpětnovazebný faktor v rovnicí HD 50a-b zabraňuje oscilacím v systému:

\[ DCN3 = (((PC - 17.0) \cdot CN7 + 17.0) \cdot CN2 - CN3) \cdot 0.1 \quad \text{(HD 50a)} \]

\[ CN3 = \int DCN3 \, dt \quad \text{(HD 50b)} \]
Výpočet aktuální hodnoty venózní rezistence (RVS) závisí na modifikaci bazální hodnoty rezistence ve venózním řečišti (RV1) pozměňovacími multiplikátory vyjadřujícími vliv autonomního nervového systému (AVE), vliv účinku angiotenzinu na tonus venózní část řečiště (ANUVN) a vliv viskozity krve (VIM) na venózní rezistenci:

\[ RVS = AVE \times VIM \times ANUVN \times RV1 \]

Venózní část rezistence v "neledvinných" a "nesvalových" měkkými tkáněmi vypočítáváme pomocí aktuální hodnoty venózní rezistence (RVS) vynásobené proporčním faktorem 1.79⁴ (vypočítáváme tím tu část celkové venózní rezistence, která přispívá na "neledvinové" a "nesvalové" tkáň – minutový průtok je přibližně 1.79 x větší než je průtok "neledvinovými a nesvalovými tkáňemi"), a zároveň zde uplatníme možnost uplatnění myogenní autoregulace prostřednictvím faktoru (MYOGRS) - tak jako i v arteriální části rezistence (RSNA), kde proměnná MYOGRS byla "schována" v proměnné R1:

\[ RSNV = RVS \times 1.79 \times MYOGRS \]

Výpočet průměrného kapilárního tlaku v tkáních (PC). Kapilární tlak je úmerný rezistenci průtoku krve malými vénami (RVS) a průtoku krve v měkkých "neledvinných a nesvalových" tkání (BFN) vynásobeným koeficientem 1.79 korigujícím průtok "neledvinných a nesvalových" tkání na celkový průtok zahrnujících i průtok svalem a ledvinou plus tlak ve velkých žilách (PVS):

\[ PC = RVS \times 1.79 \times BFN + PVS \]

Celkovou systémovou periferní rezistenci (jako výstupní parametr z hemodynamického modulu) vypočteme z gradientu arteriálního (PA) a pravého atriaálního tlaku (PRA) a z odtoku krve z arteriálního řečiště:

\[ RTP = \frac{(PA - PRA)}{QAO} \]

**Výpočet rezistencí v plicní cirkulaci**

Celková rezistence v plicích (RPT) je počítaná jako součet arteriolární a venózní složky rezistence.

Výpočet rezistence průtoku zkrze plicní arterioly RPA je založen na hodnotě plicního arteriálního tlaku (předpokládáme autoregulaci rezistence na tlaku v plicních arteriích). Tato závislost je nelineární. Rovnice 56b ohraničuje hodnoty mezivýsledku. Rovnice 57 počítá exponentiální závislost vodivosti plicních arteriol, a rovnice 58 počítá rezistenci jako převrácenou hodnotu vodivosti:

\[ PP1 = 0.026 \times PPA \] (HD 56a)
\[ když PP1 < 0.00001, pak PP1 = 0.00001 \] (HD 56b)
\[ CPA = PP1^{0.5} \] (HD 57)
\[ RPA = 1.0 / CPA \] (HD 58)

Výpočet venózní plicní rezistence souvisí s hodnotou tlaku v levé síně (PLA) – vyšší tlak vede k disenzii plicních žil a poklesu jejich rezistence:

---

⁴ Tuto část bude potřeba podrobit revizi – koeficient 1.79 nebudou konstantní při redistribucích krve v šokových stavech apod.
\[ PL_1 = PLA + 20.0 \]  
\[ RPV = 1.0 / PL_1 / 0.0357 \]  

Celková plicní rezistence je pak součet arteriolární a venózní rezistence:

\[ RPT = RPV + RPA \]

Můžeme ještě vypočítat tlak v plicních žilách (PVP). Z odtoku z plicních arterií do kompartmentu plicních žil a levého artia (QPO) a z odporu plicních žil (RPV) spočítáme příslušný tlakový spád a tuto hodnotu odečteme od tlaku v plicní arterii (PPA):

\[ PVP = PPA - QPO / RPV \]

**Výpočet výkonnosti čerpací funkce levého srdce**

Výpočet multiplikačního faktoru PA2 v závislosti na třech faktorech ovlivňujícím výkonnost levé srdeční pumpy. Prvním z nich je stimulace levého srdce autonomním nervovým systémem (AUH), ovlivnění činnosti myokardu saturací arteriální krve kyslíkem (OSA) a ovlivnění čerpací funkce komory tlakovým zatížením závisícím na hodnotě arteriálního systémového tlaku (AP):

\[ PA2 = PA / (AUH * OSA) \]

Funkční křivka počítá závislost kumulativního parametru charakterizujícím vliv tlakového zatížení levého srdce (LVM) na koeficientu PA2 (koeficient LVM je následně použit v rovnici HD 44). Empirická funkce je aproximována proložením uzlových bodů následující tabulky:

<table>
<thead>
<tr>
<th>PA2</th>
<th>LVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0</td>
<td>1.04</td>
</tr>
<tr>
<td>0</td>
<td>1.04</td>
</tr>
<tr>
<td>70</td>
<td>1.025</td>
</tr>
<tr>
<td>125</td>
<td>0.97</td>
</tr>
<tr>
<td>160</td>
<td>0.98</td>
</tr>
<tr>
<td>200</td>
<td>0.59</td>
</tr>
<tr>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>&gt;240</td>
<td>0</td>
</tr>
</tbody>
</table>

**Výpočet výkonnosti čerpací funkce pravého srdce**

Multiplikační faktor (PP2) kumulativně vyjadřuje schopnost pravého srdce pracovat proti zvýšené zátěži reprezentované zvýšeným tlakem v plicnicí (PPA). Tuto schopnost ovlivňuje aktivita sympatické stimulace.
(AUH) a dostupnost kyslíku v koronárních cévách – proto je dalším faktorem arteriální saturace hemoglobinu kyslíkem:

\[ PP2 = \frac{PPA}{(AUH \times OSA)} \]  

(Funkční křivka počítá závislost kumulativního parametru charakterizujícím vliv tlakového zatížení pravého srdce (RVM) na koeficient PP2 (koeficient RVM je následně použit v následující rovnici HD 66). Empirická funkce je aproximována proložením uzlových bodů následující tabulky:

\[ RVM = \text{function rightHeartLoading}(PP2) \]  

\[
\begin{array}{|c|c|}
\hline
\text{PA2} & \text{RVM} \\
\hline
<0 & 1.06 \\
0 & 1.06 \\
10 & 1.03 \\
20 & 0.97 \\
24 & 0.93 \\
30 & 0.88 \\
38 & 0.46 \\
45 & 0 \\
>45 & 0 \\
\hline
\end{array}
\]

Výpočet kumulativního faktoru čerpací účinnosti pumpy (HPEF), kterým se násobí normalizovaný odtok krve z pravého srdce v rovnici (HD 31). Na tento koeficient má (zhruba z 60%) vliv kontrakční činnost levého srdce – proto se výhodný koeficient vlivu levého srdce (QRF=0.6) násobí poměrem okamžité hodnoty srdečního výdeje levé komory (QLO) k normalizované hodnotě (QLN) – normalizované vzhledem k plnícímu tlaku v levé sini – viz rovnici (HD 43). Další část kumulativně počítá vliv faktorů na pravou komoru – vliv tlakového zatížení pravého srdce, vyjádřený faktorem (RVM) – z předchozí rovnice, vliv autonomní stimulace na inotropii pravé komory (AUH), vliv inotropie pravého srdce vyjádřený poměrem k normě (HSR), vliv možného poškození myokardu při šoku a jiných faktorech, vyjádřený koeficientem (HMD) a inotropní vliv hypertrofie pravého srdce (HPR):

\[ HPEF = (1 - QRF) \times RVM \times AUH \times HSR \times HMD \times HPR + QRF \times QLO \times QLN \]  

(Výpočet vlivu myogenní stimulace)

Myogenní stimulace spočívá ve zvýšení rezistence arteriál periferních tkání velkého oběhu při zvýšení tlaku. Myogenní stimulace závisí na arteriálním tlaku (PA) a na tlaku v kapilárách (PC). Rychlost myogenní odpovědi ovlivňuje časová konstanta MYOGTAU (normálně je 240 min). Rozsah vlastní odpovědi je aproximována splinovou funkcí prokládající experimentální data změny vodivosti (obrácená hodnota rezistence) navržená Colemanem. Multiplikační faktor TENSIGN určuje zesílení myogenní odpovědi (normálně TENSIGN=1).
\[ PDIFF = (PA + PC) - PADAPT \] (HD 68)

\[ DPADAPT = PDIFF / MYOGTAU \] (HD 69)

\[ PADAPT = \int (DPADAPT) \, dt \] (HD 70)

\[ MYOGRS1 = function \, myogenResp(PDIFF) \] (HD 71)

\[ MYOGRS = TENSIGN / MYOGRS1 \] (HD 72)

*Implementační poznámka:*

Při realizaci komplexního modelu v prostředí Simulink pro zabránění algebraické smyčky zařadíme funkci \textit{myogenResp}(PDUFF) (rovnice HD71) až za integrátor počítající \textit{PADPT} (v rovnici 70). Proto abychom dostali zpátky hodnotu PDUFF zderivujeme hodnotu \textit{PADPT} a výsledek vynásobíme konstantou \textit{MYOGTAU}:

\[ MYOGRS1 = function \, myogenResp(PDIFF) = function \, myogenResp((dPADAPT/dt) \ast MYOGTAU) \]
Původní a upravená implementace myogenické regulace pro odstranění algebraické smyčky

Zavedení tlumení pro numerickou stabilitu při implementaci modelu

Z důvodů zlepšení numerické stability simulačního při změnách průtokových objemů modelu je vhodné při implementaci modelu zavést tlumené faktory pro plnění pravé a levé síně. Proto v každém integračním kroku, těsně před výpočtem změn objemů krve plicních artérií a v levé síní včetně plicních žil (viz rovnice HD 33-34, HD 39-40) korigujeme aktuálně vypočtenou hodnotu odztoku krve z plicních artérií do plicních žil a levé síně (QPO) s ohledem na momentální hodnotu odztoku krve z levé síně (QLO):

\[ QPO = (QPO - QLO) \times 0.25 + QLO \]  
(HD 73)

Obdobně, těsně před výpočtem objemu krve v systémových žilách a v pravé síně (viz rovnice HD7-8 a HD17-18) korigujeme aktuálně vypočtenou hodnotu odztoku krve ze systémových žil do pravé síně (QVO) s ohledem na momentální hodnotu odztoku krve z pravé síně (QRO):

\[ QVO = (QVO - QRO) \times 0.25 + QRO \]  
(HD 74)

Srdeční frekvence a systolický objem
Vstupem do modulu je multiplikační koeficient vlivu autonomní stimulace na srdeční frekvenci (AUR), přímý vliv tlaku v pravé síni (PRA) na zvýšení srdeční frekvence – vzestup tlaku o 5 torů vede ke zvýšení frekvence o 10 tepříčků, multiplikátor (HMD) vyjadřující efekt poškození srdece - depresivní vliv hypoxie, šoku a dalších faktorů na srdece (u nepoškozeného srdece HMD=1, při poškození HMD>1):

\[ HR = (32. + 40. * AUR + PRA * 2.) * ((HMD - 1.)*5. + 1.) \]  

(Systolický objem je počítán z toku krve z levého srdece (QLO) děleného frekvencí (HR):

\[ SVO = QLO / HR \]

\[ \text{Účinek tahové relaxace velkých žil na reziduální venózní objem (VVS0)} \]

Při zvýšení náplné venózního řešitě vlivem relaxace svaloviny velkých žil dochází ke zvětšení reziduálního objemu (naplňujícího venózní řešitě při nulovém tlaku). V modelu jsou uvažovány dva mecha

\[ DVV6 = \frac{(VVE - 0.3) * SR2 - VV6}{SRK2} \]  

\[ VV6 = \int (DVV6) \, dt \]

\[ DVV7 = \frac{(VVE - 0.3) * SR - VV7}{SRK} \]  

\[ VV7 = \int (DVV7) \, dt \]

\[ \text{Zpětnovazebný vliv volumoreceptorů v pravé srdeční síní} \]

Stimulace volumoreceptorů v pravé srdeční síní ovlivňuje (tlumič) výdej antidiuretického hormonu – tento příspěvek stimulace atriálních receptorů na sekreci ADH je vyjádřen proměnnou AH7. Stimulace atriálních volumoreceptorů má dále vliv na reflexní relaxaci svaloviny velkých žil, což se projeví zvětšením reziduálního objemu žilního systému (ATRFVB – viz rovnice HD 21) a snížením rezistence v nesvalové i svalové části řešitě (ATRRFB – má vliv na koeficient R1 viz rovnice HD 46).

Nejprve se počítá nelineární závislost mezifaktoru AHZ na tlaku v pravé síně (PRA), pomocí funkční polynomické aproximace, s uvažováním citlivostního koeficientu (AH9=1.0):

\[ AH10 = 0.333 \]  

\[ AHZ2 = (\text{abs}(PRA))^{AH10} \]

\[ \text{když PRA < 0, pak AHZ2 = -AHZ2} \]  

\[ \text{Nejprve se počítá nelineární závislost mezifaktoru AHZ na tlaku v pravé síně (PRA), pomocí funkční polynomické aproximace, s uvažováním citlivostního koeficientu (AH9=1.0):} \]
Výstupní hodnota vlivu podráždění atriálních volumoreceptorů (AH7) se počítá jako odchylka mezihodnoty AHZ od momentálně nastavené hodnoty klidového natažení volumoreceptorů (AHY).

Nastavení "klidové aktivity" volumoreceptorů (AHY) se postupně adaptuje na děletrvající natažení – v modelu je to vyjádřeno integračním členem. Rychlost adaptace závisí na časová konstantě adaptéace volumoreceptorů (AH11=1000):

$$DAHY = (AHZ-AHY)/AH11$$  \hspace{1cm} (HD 82)

$$AHY = \int (DAHY) \, dt$$  \hspace{1cm} (HD 83)

$$AH7 = (AHZ-AHY)$$  \hspace{1cm} (HD 84)

Poznámka: V původním klasickém Guytonově modelu z roku 1972 se hodnota AHZ počítala jako lineárně závislá na PRA (AHZ=0.2*PRA) a hodnota časově konstanty AH11 byla 10000/7.

Uplatnění vlivu volumoreceptorů v pravé síní na změnu reziduálního objemu systémového venózního řečiště (ATRVFB – víz rovnice HD 21) se počítá z hodnoty vlivu podráždění volumoreceptorů (AH7) v závislosti na citlivostním koeficiéntu (ATRVFBM). Hodnotu citlivostního koeficiéntu Guyton v roce 86 uvádí u normálních jedinců jako nulovou – tj. prakticky se tento vliv normálně neuplatní. Z důvodů utlumení kmitání v systému při výpočtu hodnoty ATRVFB ještě zavádíme třímučí integrační člen.

$$ATRVFB0 = ATRVFBM \times AH7$$  \hspace{1cm} (HD 85a)
**DATRVFB = (ATRVFB0 - ATRVFB) * 0.1**  
(HD 85b)

**ATRVFB = \int (DATRVFB) \, dt**  
(HD 85c)

Obdobně, uplatnění vlivu atriálních volumoreceptorů na reflexní změnu rezistence systémového cévního řečiště nesvalových tkání (ATRRFB – viz rovnice HD 46) závisí na citlivostním koeficientu (ATRRFBM). Jeho hodnotu ale také Guyton v roce 86 uvádí u normálních jedinců jako nulovou – prakticky se tedy tento vliv normálně neuplatní. Hodnota multiplikačního koeficientu ATRRFBM v rovnici HD 46 je ve jmenovateli a nesmí být záporná, proto zde ještě zavádíme kontrolu na podkročení hodnoty 0.1 (kdě, viz rovnice HD 46 vlivem reflexu nesmí se zmenšit rezistence více než desetkrát):

**ATRRFB = ATRRFBM * AH7 + 1.0**  
(když ATRRFB < 0.1 pak ATRRFBM = 0.1)  
(HD 86a)

Uplatnění hypertrofie srdce a poškození srdce tkáňovou hypoxií

Inotropní vliv hypertrofie pravého srdce (HPR), vyjádřený jako relativní poměr k normě ( který se uplatňuje v rovnici HD 67) závisí především na dlouhodobém vlivu tlaku v plicnici (PPA), resp. na jeho poměru k normě, na bazální inotropii pravého srdce, vyjádřené jako poměr k normě (HSR) – čím větší je bazální inotropie, tím ménší bude sklon k hypertrofii (HSR je vstupní parametr modelu). Dalším faktorem, který se zde uplatňuje je minutový průtok srdeční (QAO), přesněji řečeno, jeho poměr k normě. Multiplikaci těchto relativních faktorů dostaneme:

**PP3 = (PPA/15) * (QAO/5) / HSR = PPA * QAO / HSR / 75**  
(HD 87)

Cílový stupeň hypertrofie (HPR1) je pak určován na citlivostním exponentem (Z13 = 0.625):

**HPR1 = PP3^Z13**  
(HD 88)

Hypertrofie se rozhvíjí pomalu, časová konstanta je zde 57600 min:

**DHRP = (HPR1 - HPR) / 57600**  
(HD 89)

**HPR = \int (DHRP) \, dt**  
(HD 90)

Obdobně, inotropní vliv hypertrofie levého srdce (HPL), vyjádřený jako relativní poměr k normě ( který se uplatňuje v rovnici HD 44) závisí především na dlouhodobém vlivu systémového arteriálního tlaku (PA), resp. na jeho poměru k normě, na bazální inotropii levého srdce, vyjádřené jako poměr k normě (HSL) – čím větší je bazální inotropie, tím ménší bude sklon k hypertrofii (HSL je vstupní parametr modelu). Dalším faktorem, který se zde uplatňuje je minutový průtok srdeční (QAO), přesněji řečeno, jeho poměr k normě. Multiplikaci těchto relativních faktorů dostaneme:

**PA4 = (PA/100) * (QAO/5) / HSL = PA * QAO / HSL / 500**  
(HD 91)

Cílový stupeň hypertrofie (HPL1) je pak určován na citlivostním exponentem (Z13 = 0.625):

**HPL1 = PA4^Z13**  
(HD 92)

Hypertrofie se rozhvíjí pomalu, časová konstanta je zde 57600 min:
DHPL = (HPL1 - HPL) / 57600  

HPL = \int (DHPL) \, dt

Čerpací funkce myokardu může být ovlivněna hypoxií – depresivní vliv tkáňové hypoxie na čerpací funkci srdce je vyjadřena multiplikačním faktorem HMD (viz rovnice HD 44, HD 67, HD 74), jeho hodnota je normální 1, při hypoxii může být menší než 1. Pokud tenze kyslíku v nesvalových tkáních (POT) poklesne pod 5 torr, hodnota HMD se bude snižovat – integrační člen v rovnici HD 96 je ohraničen shora jedničkou (a zdola nulou):

DHM = (POT - 5) * 0.0025

HMD = \int (DHM) \, dt

když HMD > 1, pak HMD = 1
když HMD < 0, pak HMD = 0

**Výpočet středního cirkulačního plnícího tlaku a počátečních hodnot objemů jednotlivých částí krevního řečiště**

Nejprve spočteme celkový reziduální objem roztahující cévy, ale ještě nezvyšující tlak (total unstressed volume) (VT0) jako součet reziduálních objemů krve v plicních žilách a levé síně (VLA0), v plicním arteriálním řečišti (včetně pravé komory) (VPA0), v pravé síně (VRA0) v žílním systémovém řečišti (VVS0) a v systémovém arteriálním řečišti (včetně levé komory) (VAS0):

VT0 = VLA0 + VPA0 + VRA0 + VVS0 + VAS0  

Z celkového objemu krve (VB) pak spočítáme celkový objem krve napínající cévy (VE0):

VE0 = VB - VT0

Z celkového objemu roztahující cévy a z poddajností levé síně a plicních žil (CLA), z poddajnosti plicních artérií (CPA), z poddajnosti pravé síně (CRA), z poddajnosti venálního systémového řečiště (CV) a z poddajnosti systémového arteriálního řečiště (CAS) můžeme vypočítat střední cirkulační plnící tlak (mean circulatory pressure), který se ustanoví v krevním řečišti při zástavě oběhu (tato rovnice je výsledkem řešení soustavy rovnic (HD 98) a (HD 99) a rovnic MCP = VEi / Ci, kde VEi a Ci jsou příslušné objemy náplni jednotlivých částí krevního řečiště roztahující cévy a příslušné poddajnosti):

MCP = VE0 / (CLA + CPA + CRA + CV + CAS)

Nakonec vypočítáme objemy krve v jednotlivých částech krevního řečiště po zástavě oběhu, když se v celém krevním řečišti uskuteční střední cirkulační plnící tlak a krev se rozdistribuuje do jednotlivých částí krevního řečiště podle příslušných poddajností. Vypočítáme počáteční objem krve v arteriálním systémovém řečišti, včetně levé komory (VSA0), ve venálním systémovém řečišti (VVS0) a v pravé síně (VRA0), v plicních artériích včetně pravé komory (VPA0) a v levé síně a plicních žilách (VLA0).

Tyto hodnoty použijeme jako počáteční hodnoty v integračních člencích rovnice (HD 08, HD 18, HD 27, HD 34, HD 40)5:

5 Toto nastavení počátečních hodnot vzchází z podmínek nulového minutového průtoku, kde všechy tlaky v cévním řečišti jsou rovny střednímu cirkulačnímu plnícímu tlaku MCP. Je možná i druhá alternativa – nastavení k hodnotám normativních tlaků.
<table>
<thead>
<tr>
<th>Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{VAS}_{\text{init}} = \text{VAS}_0 + \text{MCP} \times \text{CAS}$</td>
<td>(HD 100)</td>
</tr>
<tr>
<td>$\text{VV}_{\text{Sinit}} = \text{VV}_S + \text{MCP} \times \text{CV}$</td>
<td>(HD 101)</td>
</tr>
<tr>
<td>$\text{VRA}_{\text{init}} = \text{VRA}_0 + \text{MCP} \times \text{CRA}$</td>
<td>(HD 102)</td>
</tr>
<tr>
<td>$\text{VPA}_{\text{init}} = \text{VPA}_0 + \text{MCP} \times \text{CPA}$</td>
<td>(HD 103)</td>
</tr>
<tr>
<td>$\text{VLA}_{\text{init}} = \text{VLA}_0 + \text{MCP} \times \text{CLA}$</td>
<td>(HD 104)</td>
</tr>
</tbody>
</table>
Circulatory Dynamics - Flows And Pressures

**INPUTS:**
- **VB**: blood volume [l]
- **QRO**: actual right ventricular output [l/min]
- **QLO**: output of left ventricle [l/min]
- **VVS0**: unstressed venous systemic vascular volume [l]
- **VVE**: excess volume (stressed volume) in systemic veins [l]
- **VAS0**: unstressed volume in systemic arteries [l]
- **VRA0**: unstressed right atrial volume [l]
- **VPA0**: unstressed volume in pulmonary arteries [l]
- **VLA0**: unstressed volume in left atrium and pulmonary veins [l]
- **CAS**: capacitance of systemic arteries [l/torr]
- **CV**: capacitance of venous systemic volume [l/torr]
- **CRA**: capacitance of right atrium [l/torr]
- **CPA**: capacitance of pulmonary arteries [l/torr]
- **CLA**: capacitance of left atrium and pulmonary veins [l/torr]
- **RSN**: vascular resistance in non-muscle and non renal tissues [torr min /l]
- **RSM**: vascular resistance in muscles [torr min /l]
- **RPT**: pulmonary vascular resistance [torr min /l]
- **RR**: total renal resistance [torr min /l]
- **VIM**: blood viscosity (ratio to normal blood)
- **FIS**: conductance fot the fistula [l/min/torr]

**OUTPUTS:**
- **PA**: arterial (aortic) pressure [torr]
- **PVS**: average venous pressure [torr]
- **PRA**: right atrial pressure [torr]
- **PPA**: pulmonary arterial pressure [torr]
- **PLA**: left atrial pressure [torr]
- **VAS**: volume in systemic arteries [l]
- **VVS**: systemic venous vascular volume [l]
- **VVE**: excess blood volume in the veins [l]
- **VRA**: right atrial volume [l]
- **VPA**: volume in pulmonary arteries [l]
- **VLA**: volume in left atrium [l]
- **QAO**: blood flow in the systemic arterial system [l/min]
- **QVO**: rate of blood flow from veins into right atrium [l/min]
- **QPO**: rate of blood flow in non muscle and non renal tissues [l/min]
- **RBF**: renal blood flow [l/min]
- **RBF**: renal blood flow through a fistula [l/min]
- **RTP**: total systemic peripheral resistance [torr min /l]
- **VTO**: total unstressed blood volume [l]
- **VEO**: total excess volume of blood in the vasculature [l]
- **MCP**: mean circulatory pressure [torr]

Circulatory Dynamics - Flows And Pressures
UNSTRESSED VOLUME IN SYSTEMIC VENOUS TREE

**Inputs:**
- VVR - normal maximum volume of blood in the venous system at zero pressure (l)
- ANU - nonrenal effect of angiotensin [ratio to normal]
- ANY - sensitivity of large veins to effect of angiotensin [normal value = -0.2 l/unit of angiotensin]
- VV6, VV7 - changes in basic volume of venous system caused by stress relaxation (l)
- ATRVFB - change in basic volume of venous system caused by atrial volume receptor feedback

**Output:**
- VVS0 - the maximum volume of venous system at zero volume (so called unstressed venous volume) (l)

Unstressed volume in systemic venous tree
RIGHT HEART PUMPING

INPUTS:
- PRA - right atrial pressure [torr]
- PPA - pulmonary arterial pressure [torr]
- AUH - autonomic stimulation of heart [ratio to normal]
- OSA - oxygen hemoglobin saturation
- HSR - basic strength or right ventricle [ratio to normal]
- HPR - hypertrophy effect of heart [ratio to normal]
- HMD - cardiac depressant effect of hypoxia, shock and other factors [ratio to normal]
- QLO - output of left ventricle [l/min]
- QLN - normalised output of the left heart [l/min]

OUTPUTS:
- QRO - actual right ventricular output [l/min]
- QRN - normalised right ventricular output [l/min]
- RVM - depressing effect on right ventricle of pulmonary arterial pressure [ratio to normal]

1-D T(u) function rightHeartLoading(PP2)
cubic spline interpolation
x=[-6,-3,-1,0,2,4,8]
y=[0,0.75,2.6,5.0,9.8,12.1,13.5]

1-D T(u) function StarlingRNorm(PLA)
cubic spline interpolation
x=[-6,-3,-1,0,2,4,8]
y=[0,0.0,75,2.6,5.0,9.8,12.1,13.5]
PLA - left atrial pressure [torr]
PA - systemic arterial pressure [torr]
AUH - autonomic stimulation of heart [ratio to normal]
OSA - oxygen hemoglobin saturation
HSL - basic strength or left ventricle [ratio to normal]
HPL - hypertrophy effect of left heart [ratio to normal]
HMD - cardiac depressant effect of hypoxia, shock and other factors [ratio to normal]

QLO - actual left ventricular output [l/min]
QLN - normalised left ventricular output [l/min]
LVM - depressing effect on left ventricle of pulmonary arterial pressure [ratio to normal]
RESISTANCES IN THE SYSTEMIC CIRCULATION

**INPUTS:**
- PA - systemic arterial pressure [torr]
- RSM - basic vascular resistance of muscle [torr min/l]
- RSN - basic vascular resistance of non-muscle and non-renal tissues [torr min/l]
- MYOGRS - myogenic autoregulation effect on vascular resistance in muscle and in non-renal tissue [multiplier, ratio to normal]
- AUM - sympathetic vasoconstrictor effect on arteries in muscle and non-renal tissues [multiplier factor, ratio to normal]
- VM - blood viscosity [ratio to normal]
- ANU - nonrenal effect of angiotensin [ratio to normal]
- AMM - antidiuretic hormone effect to non renal vascular resistance, multiplier [ratio to normal effect]
- ATRRFB - multiplier factor for the effect on muscle and non-renal vascular resistance of feedback from the atrial stretch receptors [multiplier, ratio to resting state]
- ARM - muscle vascular constriction caused by local tissue control [multiplier, ratio to resting state]
- RV1 - vascular resistance in small veins [torr min/l]

**OUTPUTS:**
- RSN - vascular resistance in non-muscle and non-renal tissues [torr min/l]
- RSM - vascular resistance in muscles [torr min/l]
- RVS - resistance in small veins [torr min/l]

**Resistances in the Systemic Circulation**
Resistances in the Pulmonary Circulation and Pulmonary Venous Pressure

**Inputs:**
- **PPA** - Pulmonary arterial pressure [torr]
- **PLA** - Left atrial pressure [torr]
- **QPO** - Rate of blood flow into pulmonary veins and left atrium [l/min]

**Outputs:**
- **RPA** - Pulmonary arterial resistance [torr min/l]
- **RPV** - Pulmonary venous resistance [torr min/l]
- **RPT** - Total pulmonary vascular resistance [torr min/l]
- **PVP** - Pulmonary venous pressure [torr]
MYOGENIC AUTOREGULATION

INPUTS:
- PA - systemic arterial pressure [torr]
- PC - capillary pressure [torr]
- MYOGTAU - time delay factor of myogenic response (in normal condition TENSTC= 240 min)
- TENSIGN - factor of effectiveness of myogenic response

OUTPUT:
- MYOGRS - myogenic autoregulation effect on vascular resistance in muscle and in non-renal tissue [multiplier, ratio to normal]

CURVILINEAR
- Interpolates function value using linear (if x<=0 or x>x2) or cubic spline (if x0>x<x2) interpolation

INPUTS:
- x - value of independent variable
- xySlope - vector of coordinates and slope of 3 knots:
  - x[0], y[0], slope0,
  - x[1], y[1], slope1,
  - x[2], y[2], slope2

OUTPUTS:
- y - interpolated function value

function MyogenResp(PDIFF) - spline interpolation

-[30.0,1.5,0.0,0.0,1.0,-0.02,80.0,0.5,0.0] - data from Coleman (1992)
HEART RATE AND STROKE VOLUME

INPUTS:
- AUR - autonomic stimulation of heart rate [ratio to normal]
- HMD - cardiac depressant effect of hypoxia, shock and other factors [ratio to normal]
- PRA - right atrial pressure [torr]
- QLO - output of left ventricle [l/min]

OUTPUTS:
- HR - heart rate [beats/min]
- SVO - stroke volume [l]

Heart Rate and Stroke Volume
Effect of stress relaxation on basic venous volume

**VENOUS STRESS RELAXATION**

**INPUTS:**
- VVE - excess blood volume in the veins [l]

**OUTPUTS:**
- VV6 - increased venous vascular volume caused by long time stress realxation [l]
- VV7 - increased venous vascular volume caused by short time stress relaxation [l]

Effect of stress relaxation on basic venous volume

**EFFECT OF STRESS RELAXATION ON BASIC VENOUS VOLUME**
The volume receptor feedback mechanism

INPUTS:
- PRA - right atrial pressure [torr]
- ATRVFBM - sensitivity controller of volumereceptor feedback effect on change of basic of venous system
  \[ ATRVFBM = AH7 \times ATRVFBM, \text{no effect} = 0 \]
- ATRRFBM - sensitivity controller of volumereceptor feedback effect on nonmuscle arterial resistance
  \[ ATRRFBM = AH7 \times ATRRFBM, \text{no effect} = 0 \]

OUTPUTS:
- AH7 - effect of right atrial volume receptor reflex on ADH secretion [relative additive factor, normal value = 0]
- ATRVFB - change in basic volume of venous system caused by atrial volume receptor feedback
  \[ ATRVFB = AH7 \times ATRVFBM, \text{no effect} = 0 \]
- ATRRFB - multiplier factor for the effect on muscle and non-renal vascular resistance of feedback from the atrial stretch receptors [multiplier, ratio to resting state]
Heart hypertrophy or deterioration

**INPUTS:**
- PA - systemic arterial pressure [torr]
- QAO - blood flow in the systemic arterial system [l/min]
- PPA - pulmonary arterial pressure [torr]
- HSR - basic strength or right ventricle [ratio to normal]
- HSL - basic strength or left ventricle [ratio to normal]
- POT - Non muscle cells PO2 [torr]

**OUTPUTS:**
- HPR - hypertrophy effect of right heart [ratio to normal]
- HPL - hypertrophy effect of left heart [ratio to normal]
- HMD - cardiac depressant effect of hypoxia, shock

HEART HYPERTROPHY OR DETERIORATION

**Formulas:**
- $HPR_1 = PP3^{Z_{13}}$
- $HPL_1 = PA4^{Z_{13}}$
- $HMD_0$
DODÁVKA KYSLÍKU DO TKÁNÍ

Tkáně jsou z hlediska dodávky kyslíku rozděleny na svalové a nesvalové a dodávka kyslíku do každé z nich je počítána zvlášť. Dodávky kyslíku jsou počítány z celkových koncentrací kyslíku v arteriální krvi, z průtoků a ze spotřeby kyslíku která může být utlumována nízkou tkáňovou tenzí kyslíku.

Dodávka kyslíku do nesvalových tkání

Tok kyslíku v arteriální krvi se počítá z toku krve nesvalovou tkání (BFN) z celkové koncentrace kyslíku v arteriální krvi (OVA):
 Tok kyslíku odtékající z nesvalových tkání počítá z toku kyslíku přítěkajícím v arteriální krvi (O2ART) po odečtení rychlosti dodávaného kyslíku do tkání (DOB):

\[ O2\text{ART}=BFN\times OVA \quad (OD\ 01) \]

Z toku kyslíku ve venózní krvi a z krevního průtoku zjistíme koncentraci kyslíku ve venózní krvi odtékající z nesvalových tkání

\[ O2\text{VENN}=O2\text{ART}-DOB \quad (OD\ 02) \]

Poddělíme-li tuto koncentraci hematokritem (HK, vyjádřeným v procentech), získáme koncentraci kyslíku na litr erytrocytů, a podělením kyslíkovou kapacitou krve můžeme přibližně zjistit saturaci hemoglobinu kyslíkem (v procentech) v odtékající venózní krvi od nesvalových tkání OSV1 – při tomto výpočtu Guyton zanedbával rozpuštěný kyslík:

\[ OSV1=O2\text{VENN}/BFN/\text{HM}/5 \quad (OD\ 03) \]

Pro prevenci oscilací při prudkých změnách saturace se zavádí tlumení pomocí integrace a tlumivého koeficientu (Z7) – hodnota tohoto koeficientu Z7=5:

\[ DOSV=(OSV1-OSV)/Z7 \quad (OD\ 04) \]

Výpočet parciálního tlaku kyslíku v odtékající venózní krvi (POV) je počítán podle linearizovaného sklonu počáteční části saturační křivky hemoglobinu:

\[ POV=OSV\times 57.14 \quad (OD\ 05) \]

Gradient parciálních tlaků kyslíku mezi venózní (nebo odtékající kapilární) krvi a tkáněmi (PGRN) se v modelu bere jako rozdíl mezi tenzí kyslíku ve venózní krvi (POV) a tenzí kyslíku v buňkách (POT):

\[ PGRN=POV-POT \quad (OD\ 06) \]

Dodávka kyslíku do tkání (DOB) je přímo úměrná gradientu parciálních tlaků kyslíku (PGRN) a difúzní ploše kapilár – ta je úměrná krevnímu průtoku (BFN). Koeficient (PN5=964) vyjadřuje vliv difúzní plochy kapilár, hustoty kapilární sítě apod.. Dodávka kyslíku je nepřímo úměrná rezistenci toku kyslíku mezi kapilárami a buňkami (v této rezistenci je skryta i difúzní dráha):

\[ DOB=PGRN\times BFN\times PN5/RDO \quad (OD\ 07) \]

Rychlost změny množství kyslíku v tkáních se rovná rozdílu mezi přítokem kyslíku z krve (DOB) a jeho metabolickou spotřebou (MO2):

\[ DO2N=DO2-MO2 \quad (OD\ 08) \]

Celkové množství kyslíku (QO2) v buňkách nesvalové tkáně pak bude:

\[ QO2=\int DO2N\ dt \quad (OD\ 09) \]

Parciální tlak kyslíku v nesvalových tkáních (POT) se počítá z celkového množství rozpuštěného kyslíku v tkáních:
Výpočet rezistence toku kyslíku z kapilár do buněk (RDO) – rezistence v sobě zahrnuje i difúzní dráhu. Rezistence je zhruba kubicky závislá na tkáňovém pO₂, protože se při výpočtu se bere v úvahu i zvýšení denzity kapilár při nízkých hodnotách tenze kyslíku (POT), kdy dochází k otevření většího množství kapilár, sníží se proto difúzní dráha a tím i hodnota rezistence toku kyslíku. Sensitivita je určována exponentem (EP=3) a hraniční hodnotou (RDOMin=50):

\[ RDO = POT^{EP} \]  
\[ \text{když } RDO < RDOMin, \text{ pak } RDO = RDOMin \]

Spotřeba kyslíku v nesvalových tkáních (MO₂) je ovlivňována stimulací sympatikem a může být utlumována nízkou tenzí kyslíku v tkáních. Při výpočtu spotřeby kyslíku v nesvalových tkáních (MO₂) proto multiplikační faktor vyjadřující tuto závislost (AOM) spolu s multiplikačním faktorem vyjadřujícím příkon vliv nízkého tenza kyslíku na spotřebu kyslíku (PDO) násobí bazální spotřebu kyslíku v nesvalových tkáních (O₂M):

\[ MO₂ = AOM \times PDO \times O₂M \]

Spotřeba kyslíku v nesvalových tkáních klesá, pokud tenze kyslíku v tkání poklesne pod určitou prahovou hodnotu (v modelu je uvažováno 8 torrů). Multiplikační faktor (PDO), vyjadřující vliv nízké tkáňové tenze kyslíku (POT) na spotřebu kyslíku se při tenzích kyslíku nad 8 torrů rovná jedné, při po poklesu kyslíku postupně klesá k nule:

\[ \text{když } (POT > 8.0) P1O = 8.0 \text{ jinak } P1O = POT \]

\[ PDO = 1 - (8.0001 - P1O)^{3}/512.0 \]
**Inputs:**
- OVA - oxygen content of arterial blood [ml O2 STPD/l blood]
- BFN - blood flow in non-muscle and non-renal tissues [l/min]
- HM - hematocrit [%]
- AOM - autonomic effect on tissue oxygenation [ratio to normal]
- Q2M - basic oxygen utilisation in non-muscle cells [ml O2 STPD/min]

**Outputs:**
- OSV - non-muscle venous oxygen saturation [%]
- POV - non-muscle venous PO2 [torr]
- POT - non-muscle cell PO2 [torr]
- DOR - rate of oxygen delivery to non-muscle cells [ml O2 STPD/min]
- MO2 - rate of oxygen utilisation by non-muscle tissues [ml O2 STPD/min]
- QO2 - non-muscle total cellular oxygen [ml O2 STPD]

**Delivery of Oxygen to the Non-muscle Tissues**

\[ \text{QO2} = 2.8 \times \text{BFN} \]

\[ \text{POT} = \text{POT}^3 \]

\[ \text{DOB} = 7.35 \]

\[ \text{MO2} = 164.5 \]

\[ \text{QO2} = 2.87 \times \text{DO2N} \]
Době plánování kyslíku

O2VENM - oddělit kyslíku ze svalů venozní krvi

OVM - koncentrace kyslíku ve venozní krvi svalů

PVO - PO2 v odtékající venozní krvi ze svalů

PGRM - gradient PO2 mezi krvi a svalovou buňkou

RDOM – celková rezistence toku kyslíku z krve do buněk

PMO - PO2 ve svalových buňkách

tlumivý vliv nízkých PO2 na spotřebu kyslíku ve svalech

OMM – bazální spotřeba kyslíku ve svalech

O2ARTM - přítok kyslíku do svalů arteriální krvi

OVA – celková koncentrace kyslíku v arteriální krvi

BFM - průtok krve valovou tkání

Diffúzní plocha kapilár otevřených pro průtok

RMO - dodávka kyslíku kyslíku do svalových tkání

QOM - celkové množství kyslíku ve svalových buňkách

MMO - spotřeba kyslíku ve svalech

AOM – vliv sympatiku na spotřebu kyslíku ve svalech

EXC – vliv svalové práce na spotřebu kyslíku ve svalech
Tok kyslíku v arteriální krvi do svalů (O2ARTM) se počítá z toku krve svalovou tkání (BFM) z celkové koncentrace kyslíku v arteriální krvi (OVA):

\[ O2ARTM = BFM \times OVA \quad \text{(OD 17)} \]

Tok kyslíku odtékající ze svalových tkání (O2VENNM) počítá z toku kyslíku přítékající v arteriální krvi (O2ARTM) po odečtení rychlosti dodávaného kyslíku do svalů (RMO):

\[ O2VENM = O2ARTM - RMO \quad \text{(OD 18)} \]

Z toku kyslíku ve venózní krvi a z krevního průtoku zjistíme koncentraci kyslíku ve venózní krvi odtékající z svalův:

\[ OVM = O2VENM / BFM \quad \text{(OD 19)} \]

Podělíme-li tuto koncentraci hematokritem (HK, vyjádřeným v procentech), získáme koncentraci kyslíku na litr erytrocytů, a podělením kyslíkovou kapacitou krve můžeme přibližně zjistit saturaci hemoglobinu kyslíkem (v procentech) v odtékající venózní krvi od svalův OVS1 – při tomto výpočtu Guyton zanedbával rozpuštěný kyslík:

\[ OVS1 = O2VENM / BFM / HK / 5 \quad \text{(OD 20)} \]

Pro prevenci oscilací při prudkých změnách saturace se zavádí tlumení pomocí integrace a tlumivého koeficientu (Z6) – hodnota tohoto koeficientu Z6=5:

\[ DOVS = (OVS1 - OVS) / Z6 \quad \text{(OD 21)} \]

\[ OSV = \int DOVS \, dt \quad \text{(OD 22)} \]

Výpočet parciálního tlaku kyslíku v odtékající venózní krvi (PVO) je počítán podle linearizovaného sklonu počáteční části saturační křivky hemoglobinu:

\[ PVO = OVS \times 57.14 \quad \text{(OD 23)} \]

Gradient parciálních tlaků kyslíku mezi venózní (nebo odtékající kapilární) krvi a svalovou tkání (PGRM) se v modelu bere jako rozdíl mezi tenzí kyslíku ve venózní krvi (PVO) a tenzí kyslíku ve svalových buňkách (PMO):

\[ PGRM = PVO - PMO \quad \text{(OD 24)} \]

Dodávka kyslíku do svalů (RMO) je přímo úměrná gradientu parciálních tlaků kyslíku (PGRM) a difúzní ploše kapilár – ta je úměrná krevnímu průtoku (BFM). Koeficient (PM5=125) vyjadřuje vliv difúzní plochy kapilár, hustoty kapilární sítě apod. Dodávka kyslíku je nepřímo úměrná rezistenci toku kyslíku mezi kapilárami a svalovými buňkami (RDOM) - v této rezistenci je skryta i difúzní dráha, rezistence je nižší než v nesvalové tkáni:

\[ RMO = PGRM \times BFM \times PM5 / RDOM \quad \text{(OD 25)} \]

Rychlost změny množství kyslíku ve svalech se rovná rozdílu mezi přítokem kyslíku z krve (RMO) a jeho metabolickou spotřebou (MMO):

\[ DQOM = RMO - MMO \quad \text{(OD 26)} \]

Celkové množství kyslíku (QOM) ve svalovových buňkách pak bude:
Parciální tlak kyslíku v svalových buňkách (PMO) se počítá z celkového množství kyslíku ve svalech (QOM). Při výpočtu je nutno uvažovat vazbu kyslíku na myoglobin. Vazebná křivka je aproximována výrazem s koeficienty (PK1=2500 a PK2=800):

\[ PMO = \frac{PK2}{PK1 - QOM} \] (OD 28)

Výpočet rezistence toku kyslíku z kapilár do svalových buněk (RDOM) – rezistence v sobě zahrnuje i difúzní dráhu. Rezistence je zhruba kvadraticky závislá na tkáňovém pO2, protože se při výpočtu se bere v úvahu i zvýšení denzity kapilár při nízkých hodnotách tenze kyslíku (PMO), kdy dochází k otevření většího množství kapilár, sníží se proto difúzní dráha a tím i hodnota rezistence toku kyslíku. Sensitivita je určována exponentem (PK3=2), ohraničicím koeficientem (PM4=1) a hraniční hodnotou (PM3=0,001):

\[ RDOM = PM1^{PK4} - PM4 \]

(když PMO<PM3, pak PM1=PM3 jinak PM1=PMO) (OD 29a)

\[ RDOM = PM1^{PK4} - PM4 \]

(když PMO<PM3, pak PM1=PM3 jinak PM1=PMO) (OD 29b)

Spotřeba kyslíku v svalech (MMO) je ovlivňována stimulací sympatikem a může stoupat vlivem tělesné zátěže. Při výpočtu spotřeby kyslíku proto bazální spotřebu ve svalech (OMM) násobíme multiplikačním faktorem vyjadřujícím vliv sympatiku (AOM), faktorem vyjadřujícím vliv svalové práce na zvýšení bazální kladové spotřeby kyslíku (EXC). Při poklesu PO2 ve svalech pod hraniční hodnotu 8 torrach dochází k útlumu spotřeby kyslíku, proto výraz ještě násobíme multiplikačním faktorem vyjadřujícím třumivý vliv nízkých tenzí kyslíku na spotřebu kyslíku (PDOM):
MMO = AOM * EXC * PDOM * OMM

Spotřeba kyslíku v svalovu klesá, pokud tenze kyslíku v tkáně poklesne pod prahovou hodnotu 8 torrů. Tato "kritická" hodnota tenze (P2O) pak ovlivňuje spotřebu kyslíku. Je-li nad 8 torrů, pak je spotřeba kyslíku řízena pouze metabolickými nároky buněk a tenze PO2 na ní nemá vliv, pokud poklesne pod tuto hodnotu, spotřeba kyslíku ve svalu se sníží. Tato závislost je aproximována kubickým polynomem. Výsledkem je multiplikační faktor (PDOM), vyjadřující tlumivý vliv nízké tkáňové tenze kyslíku (PMO) na spotřebu kyslíku. Při tenzích kyslíku nad 8 torrů se tento faktor rovná jedné, při poklesu kyslíku postupně klesá k nule (tlumivý vliv aporximujeme stejnou závislostí jako v nesvalové tkáni):

\[
\begin{align*}
\text{když} & \quad (PMO>8.0) \quad P2O=8.0 \quad jinak \quad P2O=PMO \\
PDOM &= 1 - (8.0001-P2O)^{3/512.0} \quad (OD 31a) \quad (OD 31b)
\end{align*}
\]
DELIVERY OF OXYGEN TO THE MUSCLES

**Inputs:**
- OVA - oxygen content of arterial blood [ml O2 STPD/l blood]
- BFM - blood flow in muscles [l/min]
- HM - hematocrit [%]
- AOM - autonomic effect on tissue oxygenation [ratio to normal]
- OMM - basic oxygen utilisation in muscles [ml O2 STPD/min]
- EXC - Effect of exercise on the metabolic usage of oxygen by the muscles [ratio to resting state]

**Outputs:**
- OVS - muscle venous oxygen saturation [%]
- PVO - muscle venous PO2 [torr]
- PMO - muscle cell PO2 [torr]
- RMO - rate of oxygen delivery to muscles [ml O2 STPD/min]
- MMO - rate of oxygen utilisation by muscles [ml O2 STPD/min]
- QOM - muscle total cellular oxygen [ml O2 STPD]

**Delivery of Oxygen to the Muscles**
Cirkulační systém je z hlediska autoregulačního řízení průtoku rozdělen na tři oddělené části: na ledviny ( které jsou zahrnuty do samostatného modulu), na autoregulaci průtoku krve ostatními nesvalovými (a neledvinovými) tkáněmi a na autoregulaci průtoku krve ve svalové tkáni.

**Autoregulace lokálního průtoku krve v nesvalové tkáni**

Vlastní regulátor je sestaven ze tří integračních komponent s různou rychlostí regulace: rychlou, střední a pomalou. Všechny tři závisí na hladině kyslíku v tkáních. První dvě složky odpovídají rychlým metabolickým změnám, první je téměř okamžitá, druhá nastupuje v průběhu dvaceti minut až hodiny. Třetí pomalá složka reprezentuje strukturální změny objevující se po týdnu a déle a je důsledkem vasokonstrikce nebo vasodilatace, které nastávají v cívním řečišti v důsledku dvou pních krátkodobých metabolických odpovědí.

V původním Guytonově modelu z roku 1972, se regulační odchylka počítala z tenze kyslíku ve venální krvi nesvalových tkání (POV). V pozdějších modifikacích Guyton počítal regulační odchylku z tkáňové tenze kyslíku (POT). V tomto modelu je proto regulační odchylka lokálního řízení krevního průtoku v nesvalové tkáni pro všechny tři regulátory počítána z rozdílu mezi okamžitou hodnotou tenze kyslíku v buňkách nesvalové tkání (POT) a její náležitou hodnotou (POR=8 torr):

\[ \text{POD}=\text{POT}-\text{POR} \]  

(CITIVOST Rychlé regulační komponenty je řízena proměnnou (POK), za normálních okolností POK=0.1. Výstupem z integračního tlumivého členu, ovlivňovaného koeficientem (Z=5) je proměnná POB:

\[ \text{DPOB}=\left(\text{POK} \times \text{POD}+1-\text{POB}\right)/\text{Z} \]  

\[ \text{POB}=\int \text{DPOB} \, dt \]  

Výstup POB je zdola ohraničen hodnotou 0.4 (což znamená, že tato složka regulátoru reaguje na pokles POT<2 torr stejně jako kdyby POT=2 torr) a poté je vstupem do integračního zpožďovacího členu s časovou konstantou (A1K), jejíž hodnota je 1 min. Výstupem je rychlá regulační složka AR1 (díky ohraničení POB je její nejmenší možná hodnota 0.4):

\[ \text{když} (\text{POB}<0.4) \text{ pak } \text{POB}=0.4 \]  

\[ \text{DAR1}=\left(\text{POB}-\text{AR1}\right)/\text{A1K} \]  

\[ \text{AR1}=\int \text{DAR1} \, dt \]  

Citlivost střední regulační komponenty je řízena proměnnou (PON), za normálních okolností PON=0.1, je tlumena integračním členu ovlivňovaným, jako i předchozím případě rychlé komponenty, koeficientem (Z=5). Výstupem z integračního členu je proměnná POA.

\[ \text{DPOA}=\left(\text{PON} \times \text{POD}+1-\text{POA}\right)/\text{Z} \]  

\[ \text{POA}=\int \text{DPOA} \, dt \]  

Výstup POA je zdola ohraničen hodnotou 0.5 (což znamená že tato složka regulátoru už dále nemění intenzitu své odpovědi pokud POT poklesne pod 3 torry, tj. při POT<3 torr reaguje stejně jako by POT=3 torr. To také znamená, že regulační složka AR2 má nejmenší možnou hodnotu 0.5). Po kontrole mezí je POT vstupem do integračního zpožďovacího členu s časovou konstantou (A2K=20 min). Výstupem je střední regulační složka AR2:
když \( POA<0.5 \) pak \( POA=0.5 \) 
\[ DAR2=(POA-AR2)/A2K \] 
\[ AR2=\int DAR2 \, dt \]

Citlivost pomalé regulační komponenty je řízena proměnnou \( POZ \), za normálních okolností \( POZ=16 \).

Výstup \( POC \) je zdola ohraničen hodnotou \( 0.3 \). To znamená, že pokud \( POT \) poklesne na nižší hodnotu než 7.9562, pak \( POC \) již nebude klesat pod hodnotu 0.3 (a proto následně i pomalá regulační složka \( AR3 \) má nejmenší možnou hodnotu 0.3). Po kontrole mezi je proměnná \( POC \) vstupem do integračního zpožďovacího členu s časovou konstantou \( (A3K=10 \, 000 \, \text{min}, tj. zhruba 7 \, \text{dní}) \). Výstupem je dlouhodobá regulační složka \( AR3 \):

\[ POC=POZ*POD+1.0 \] 
\[ když \ (POC<0.3) \ \text{pak} \ \ POC=0.3 \] 
\[ DAR3=(POC-AR3)/A3K \] 
\[ AR3=\int DAR3 \, dt \]

Souhrnný vliv všech tří regulačních složek, tj. rychlé \( (AR1) \), střední \( (AR2) \) a pomalé \( (AR3) \) je počítán jako jejich součin – výsledkem je proměnná \( ARM1 \):

\[ ARM1=AR1*AR2*AR3 \]

Ze souhrnného vlivu všech tří regulačních složek \( (ARM1) \) je nakonec vypočítávána souhrnná regulační odpověď cévního řešitě na tkáňovou tenzí kyslíku v nesvalových měkkých tkáních \( (ARM) \). Výsledná odpověď je souhrnně ověřena citlivostí, charakterizovanou koeficientem \( AUTOSN \) (jeho hodnota je za normálních okolností rovna 0.9):

\[ ARM=(ARM1-1)*AUTOSN+1 \]
**NON MUSCLE LOCAL BLOOD FLOW CONTROL**

**INPUT:**
POT - non-muscle cells PO2 [torr]

**OUTPUT:**
ARM - vasoconstrictor effect of local blood flow autoregulation in non-muscle tissues [ratio to normal]

Non-muscle Local Blood Flow Control

**NON MUSCLE BLOOD FLOW CONTROL**

**INPUT:**
POT - non-muscle cells PO2 [torr]

**OUTPUT:**
ARM - vasoconstrictor effect of local blood flow autoregulation in non-muscle tissues [ratio to normal]

Non-muscle Blood Flow Control
Autoregulace lokálního průtoku krve ve svalové tkáni

Na rozdíl od regulace v nesvalové tkáni je regulátor lokálního průtoku krevního řečiště ve svalech sestaven pouze ze dvou integračních komponent s různou rychlostí regulace: komponenty s rychlou odezvou, která umožňuje okamžité přizpůsobení krevního průtoku požadavkům metabolismu svalů a pomalou, dlouhodobou komponentu reprezentující strukturální adaptační změny objevující se řádově po týdnu poškození vyvolávajícího stimulu. Guyton ve svém původním klasickém modelu z roku 1972 uvažoval pouze rychlou regulaci komponentu.

Krom toho, v původním Guytonově modelu z roku 1972, se regulační odchylka počítala z tenze kyslíku ve venční krvi odtékající ze svalů (PVO). V pozdějších modifikacích Guyton počítal regulační odchylku z tenze kyslíku ve svalových buňkách (POM) a do regulátoru přidal i dlouhodobou komponentu. Regulační odchylka lokálního řízení krevního průtoku ve svalové tkáni je pro oba regulátory počítána z rozdílu mezi okamžitou hodnotou tenze kyslíku v buňkách svalů (POM) a její následnou hodnotou (PORM=8 torrů):

\[ PDO = POM - PORM \]  
(AR 18)

Citlivost rychlé regulační komponenty je řízena proměnnou (POM), za normálních okolností POM=0.2. Řídící výstupní tlak, vstupující do regulátoru, je uložen v proměnné POE:

\[ POE = POM \times(PDO + 1) \]  
(AR 19)

Výstup POE je zdola ohraničen hodnotou 0.005 (což znamená, že tato složka regulátoru reaguje na pokles POM<3.025 torr stejně jako kdyby POM=3.025 torr) a poté je vstupem do integračního zpožďovacího členu s časovou konstantou (A4K), jejíž hodnota je 10 min (v roce 72 Guyton uvažoval extrémně rychlou adaptaci s časovou konstantou 0.025 min). Výstupem je rychlá regulační složka AMM1 (díky ohraničení POM je její nejmenší možná hodnota 0.005):

\[ \text{když } (POE < 0.005) \text{ pak } POE = 0.005 \]  
(AR 20)

\[ DAMM1 = (POE - AMM1) / A4K \]  
(AR 21)

\[ AMM1 = \int DAMM1 \, dt \]  
(AR 22)

Vstup do komponenty pomalé složky (POF) je počítán z regulační odchylky a citlivostního koeficientu (POM2=16).

\[ POF = POM \times PDO + 1 \]  
(AR 23)

Poměrná POF je vstupem do integračního zpožďovacího členu s časovou konstantou (A4K2=10 000 min, tj. zhruba 7 dny). Výstupem je dlouhodobé regulační složka AMM2. Poměrná POF je zdola ohraničena hodnotou 0.3. Znamená to, že pokud POM poklesne na nižší hodnotu než 7.9562, pak POF již nebude klesat pod hodnotu 0.3 (a proto následně i pomalá regulační složka AR3 má nejmenší možnou hodnotu 0.3). Pokud by nebylo zavedeno ohraničení zdola, proměnná AMM2 by při poklesu POM pod hodnotu 7.9375 mohla nabývat záporných hodnot.

\[ \text{když } (POF < 0.3) \text{ pak } POF = 0.3 \]  
(AR 24)

\[ DAMM2 = (POF - AMM2) / A4K2 \]  
(AR 25)

\[ AMM2 = \int DAMM2 \, dt \]  
(AR 26)
Vliv obou regulačních složek, tj. rychlé (AMM1), pomalé (AMM2) ne celkovou regulační odpověď je počítán jako jejich součin – výsledkem je souhrnná regulační odpověď cévního češtiny na tkáňovou tenzi kyslíku v nesvalových měkkých tkáních (AMM).

\[ AMM = AMM1 \times AMM2 \]  

(AR 27)
MUSCLE LOCAL BLOOD FLOW CONTROL

INPUT:
PMO - muscle cells PO2 [torr]

OUTPUT:
AMM - vasoconstrictor effect of local blood flow autoregulation in muscles [ratio to normal]

Muscle Local Blood Flow Control

MUSCLE BLOOD FLOW CONTROL

INPUT:
PMO - muscle cells PO2 [torr]

OUTPUT:
AMM1 and AMM2 - autoregulation in muscles

Long term autoregulation

Short term autoregulation

PMO

7.996215

PMO

POE

POE

POE

POE

POE

POE

PMO

PMO
**AUTONOMNÍ ŘÍZENÍ CIRKULACE**

Vliv PO2 na stupeň aktivace autonomního systému: vstupem je tenze kyslíku v nesvalových tkáních (POT). Vstupní hodnota je ohraničena zhora a zdola (8 a 4 torry):

\[
\text{když } (\text{POT}<4) \{\text{POQ}=4\} \text{ jinak } \{\text{POQ}=\text{POT}\} \quad (\text{AU 01})
\]

\[
\text{když } (\text{POT}>8) \text{ pak } \{\text{POQ}=8\} \quad (\text{AU 02})
\]

Výpočet faktoru který je hlavní určující složkou autonomního řídicího tlaku (PA1), vypočítaného v rovnici (AU 07) podle hodnoty arteriálního systémového tlaku a podle korigované tenze kyslíku v nesvalové tkání (POQ):

\[
\text{PAPQ}=PA*\text{POQ}/8.0 \quad (\text{AU 03})
\]

Výpočet korekčního faktoru (EXE1), vyjadřujícího vliv tenze kyslíku ve svalech (PMO) na cirkulaci: autonomní systém je ovlivňován tenzí kyslíku ve svalech pouze tehdy když tenze kyslíku ve svalových buňkách poklesne pod kritickou hodnotu 8 torrů, pak také snižuje spotřebu kyslíku (viz modul Dodávka kyslíku do tkání, rovnici OD 31a). Proto se nejprve vypočte korigovaná tenze kyslíku ve svalových buňkách, která může ovlivnit metabolismu a autonomní systém (P2O) a vyjadřující faktor počítáme z velikosti poklesu pod kritickou hodnotu 8 torrů. Citlivostní koeficient (EX1=3) reprezentuje citlivost autonomního systému na deficit tenze kyslíku ve svalech. Poznámka: rovnice AU 04a je identická s rovnici OD 31a, Guyton proto zde má vstupní proměnnou do modulu přímo P2O, my ale díky zvýraznění modularity jsme raději zvolili jako vstupní hodnotu aktuální tenzi kyslíku ve svalových buňkách (PMO) a hodnotu P2O si zde znovu spočítáme:

\[
\text{když } (\text{PMO}>8.0) \text{ pak } \text{P2O}=8.0 \text{ jinak } \text{P2O}=\text{PMO} \quad (\text{AU 04a})
\]

\[
\text{EXE1}=(8-\text{P2O})*\text{EX1} \quad (\text{AU 04b})
\]

Výpočet korekčního faktoru EXE2 vyjadřujícího vliv tělesné zátěže na autonomní systém. Vstupem je zde koeficient (EXC) vyjadřující intenzitu svalové zátěže vyjádřenou jako podíl bazální spotřeby kyslíku ve svalech při cvičení vzhledem ke klidovému stavu. Citlivostní koeficient (Z12=0.5) zde vyjadřuje sílu vlivu cvičení na autonomní systém – Guyton v původním modelu z roku 1972 uváděl normální hodnotu tohoto koeficientu 1.24, později v roce 1986 jeho hodnotu snížil na 0.5:

\[
\text{EXE2}=(\text{EXC}-1)*\text{Z12} \quad (\text{AU 05})
\]

Sumarizační vliv ovlivnění autonomního systému pro řízení cirkulace sníženou tenzi PO2 ve svalových buňkách (EXE1) a zvýšením tělesné zátěže (EXE2):

\[
\text{EXE}=\text{EXE1}+\text{EXE2} \quad (\text{AU 06})
\]

Výpočet autonomního řídicího tlaku vlivem faktorů vyjadřujících vliv arteriálního systémového tlaku a tenze kyslíku v nesvalových tkáních (PAPQ), faktoru vyjadřujících vliv zvýšení bazální spotřeby kyslíku a tělesné zátěže (EXE) a faktoru vyjadřujícího přímý vliv angiotenzinu na vasomotorické centrum (ANUBR):

\[
\text{PA1}=\text{PAPQ}+\text{EXE-ANUBR} \quad (\text{AU 07})
\]

Vliv periferních chemoreceptorů na vasomotorické centrum: z autonomního řídicího tlaku (PA1) se vypočítá řídící vliv periferních chemoreceptorů na vasomotorické centrum (AUC). Je-li hodnota PA1 větší než 80 torrů (jako je tomu i za normálních okolností), pak je hodnota AUC nulová. Lineárně se
zvyšuje z nuly na 1.2 při poklesu PA1 z 80 na 40 torrů. Při poklesu PA1 pod 40 torrů setrvává na konstantní úrovni 1.2.

\[
\begin{align*}
\text{když } (PA1 > 80), & \text{ pak } AUC = 0 \quad \text{(AU 08a)} \\
\text{když } (40 \leq PA1 \leq 80), & \text{ pak } AUC = 0.03 \times (80 - PA1) \quad \text{(AU 08b)} \\
\text{když } (PA1 < 40), & \text{ pak } AUC = 1.2 \quad \text{(AU 08c)}
\end{align*}
\]

Závislost hodnot AUC na PA1 počítaná podle rovnice AU 08

Vliv baroreceptorů na vasomotorické centrum: z autonomního řídícího tlaku (PA1) se vyvozuje řídící vliv baroreceptorů na vasomotorické centrum (AUB). Je-li hodnota PA1 větší než 170 torrů, pak je hodnota AUC nulová. Při poklesu PA1 ze 170 na 40 torrů se lineárně zvyšuje na svou maximální hodnotu (1.85718) a při dalším poklesu PA1 pod 40 torrů na této maximální hodnotě setrvává (při normální hodnotě PA1=40, AUB=0):

\[
\begin{align*}
\text{když } (PA1 > 170), & \text{ pak } AUB = 0 \quad \text{(AU 09a)} \\
\text{když } (170 \leq PA1 \leq 40), & \text{ pak } AUB = 0.014286 \times (170 - PA1) \quad \text{(AU 09b)} \\
\text{když } (PA1 < 40), & \text{ pak } AUB = 1.85718 \quad \text{(AU 09c)}
\end{align*}
\]
Závislost hodnot AUB na PA1 počítaná podle rovnice AU 09

Tato lineární závislost je dále nelineárně modifikována, z hodnoty AUB je počítána výsledná hodnota A1B vyjadřující řídící vliv baroreceptorů na vasomotorické centrum i s ohledem na citlivost barorecepce, reprezentovanou koeficientem (AUX), normální hodnota AUX=3…

když (AUB<1), pak $A1B = AUB^{AUX}$ (AU 10a)

když (AUB>=1), pak $A1B = (AUB-1)AUX+1$ (AU 10b)
Závislost hodnot $A1B$ na $AUB$ počítaná podle rovnice AU 10

Závislost řídicího vlivu baroreceptorů ($A1B$) na autonomním řídicím tlaku ($PA1$) je pak v dolní části křivky (při $PA1>100$ a následném $A1B<1$) nelineární. Sklon křivky závisí na citlivosti barorecepce, vyjádřenou koeficiecentem $AUX$ (normálně $AUX=3$).
Závislost hodnot A1B na PA1 při různých hodnotách AUX (počítáno podle AU 09 a AU 10)

Poslední část výpočtu zahrnuje adaptaci baroreceptorů na změněný tlak. Výsledná hodnota řídícího vlivu baroreceptorů s uvažováním adaptace (AU6) je počítána jako rozdíl aktuální hodnoty A1B a historicky kumulované odchylky od normy (AU4) počítané v integračním členu. Rychlost adaptace závisí na koeficientu (AUK) jehož hodnota zde reprezentuje převracenou hodnotu časové konstanty AUK=0.005=1/(2000 min). Po ustání adaptace je hodnota AU6 rovna 1.

\[ AU6 = A1B - AU4 \]  \hspace{1cm} (AU 11)

\[ DAU4 = AUK \times (AU6 - 1) \]  \hspace{1cm} (AU 12)

\[ AU4 = \int DAU4 \, dt \]  \hspace{1cm} (AU 13)

Z hlediska numerických výpočtů je zde nutné v počátečním časovém kroku nastavit počáteční hodnotu AU4. Tato iniciální hodnota (AU4init) je nulová:

\[ AU4init = 0 \]  \hspace{1cm} (AU 14)

Příkladní vliv mozkové ischémie na vasomotorické centrum: z autonomního řídícího tlaku (PA1) se vypočítává řídící vliv ischémie vlastního vasomotorického centra (AUN). Je-li hodnota PA1 větší než 50 torrů (jako je tomu i za normálních okolností), pak je hodnota AUN nulová. Lineárně se zvyšuje z nuly na 6 při poklesu PA1 z 50 na 20 torrů. Při poklesu PA1 pod 20 torrů setrvává na konstantní úrovni 6:

když (PA1>50), pak AUN=0  \hspace{1cm} (AU 15a)

když (20<=PA1<=50), pak AUN=0.2*(50-PA1)  \hspace{1cm} (AU 15b)

když (PA1<20), pak AUB=6  \hspace{1cm} (AU 15c)
Výpočet řídícího vlivu vasomotorického centra (DAU) je realizován jako součet všech tří řídících vlivů, které toto centrum ovlivňují – působení vstupů z periferních chemoreceptorů (AUC), z baroreceptorů (AU6) a přímý vliv ischémie v centrálním nervovém systému na vasomotorické centrum (AUN):

$$DAU = AUC + AU6 + AUN$$  \hspace{1cm} (AU 16)

Za normálních okolností (při PA1=100 torrů a normální oxygenace svalové a nesvalové tkáně) je hodnota AU6=1, ostatní dva vstupy v předchozí rovnici jsou nulové (AUC=0, AUN=0), DAU se proto rovná jedné. Při změnách vstupních parametrů dochází k reakci autonomního centra a následné změně hodnot DAU. Pro prevenci oscilací Guyton se zde vkládá dva tlumivé členy. První tlumivý člen (viz rovnice AU 17-18) se objevil ve verzi modelu z roku 1986, má tlumivé koeficienty (Y=1 a Z=5), druhý tlumivý člen (rovnice AU 19-20) byl již v původním modelu z roku 1972 (hodnota tlumivého koeficientu Z8=1). Hodnota integračního členu AUJ je zde ohraničena zdola hodnotou 10⁻⁶. V počáteční hodnoty integrátorů AUJ1 a AUJ2 se v počátečním časovém kroku nastavují počáteční hodnotu 1.

$$DAUJ1 = (DAU - AUJ1) / Y / Z$$  \hspace{1cm} (AU 17)

$$AUJ1 = \int DAUJ \, dt$$  \hspace{1cm} (AU 18)

$$DAUJ = (AUJ - DAUJ1) * 6.0 / Z8$$  \hspace{1cm} (AU 17)

$$AUJ = \int DAUJ \, dt$$  \hspace{1cm} (AU 18a)

když (AUJ<0) AUJ=0.000001  \hspace{1cm} (AU 18b)
Přepočet řídícího vlivu vasomotorického centra po uplatnění tlumivých vlivů omezujících oscilaci v systému (AUJ) na multiplikační koeficient vyjadřujícím všeobecný vliv aktivity sympatického autonomního nervového systému (AU) na cirkulaci (vyjádřený jako poměr vzhledem k normě). Senzitivita vlivu aktivity vasomotorického centra (AUJ) na výslednou hodnotu aktivity autonomního systému (AU0) je určována citlivostním koeficientem (AUZ). Normální hodnota citlivostního koeficientu: AUZ=1. Jeho vliv je různý, pokud hodnota AUJ je větší nebo menší než jedna:

\[ \text{když } (AUJ \geq 1), \text{ pak } AU0 = (AUJ - 1) \cdot AUZ + 1 \]  
\[ \text{když } (AUJ < 1), \text{ pak } AU0 = AUJ \cdot AUZ \] (AU 19a)  
(AU 19b)

Závislost hodnot AU0 na AUJ při různých hodnotách AUZ

Aktivitu autonomního systému ovlivňuje také fyzická aktivita. Stupěň fyzické aktivity (EXC) je vnějším vstupem do systému. Tento vstupní parametr vyjadřuje aktuální stupeň fyzické aktivity jako poměr bazální spotřeby kyslíku ve svalech ke klidové hodnotě spotřeby kyslíku. Přes exponent EXC1=0.35 tento vstup modifikuje aktivitu autonomního systému (AU0). Výsledkem je sumární aktivita autonomního systému (AU), podle níž se pak dále počítají multiplikační koeficienty vyjadřující specifický vliv autonomního nervového systému na jednotlivé díly subsystémy organismu. Poznámka: Guyton ve svém původním modelu z roku 1972 používal hodnotu AU k ovlivňování srdeční frekvence, spotřeby kyslíku ve svalech a vlivu na tvorbu ADH, v tomto modelu se k tomuto účelu používají specializované multiplikátory, pro jejich výpočet je však základem hodnota AU:

\[ AU = AU0 \cdot \text{EXC}^{EXC1} \]  
(AU 20)
Hodnoty AU počítané podle rovnice AU 20 při různých hodnotách EXC1

Jako základ pro výpočet příslušných multiplikátorů, vyjádřujících vliv autonomního nervového systému na další subsystémy organismu vypočítáme odchylku (AUO) celkové aktivity autonomního nervového systému (AU) od normy:

\[ AUO = AU - 1.0 \]  
(AU 21)

Výpočet multiplikátoru, vyjádřujícího vliv autonomního nervového systému na inotropii srdece (AUH), tento koeficient ovlivňuje systolický objem levé a pravé komory. Příslušný citlivostní koeficient (AUV=0.3) vyjadřuje sílu, kterou se odchylka od normálního tonu sympatiku (vyjádřená hodnotou proměnné AUO) projeví na změně hodnoty multiplikátoru ovlivňujícího minutowý výdej levé a pravé komory:

\[ AUH = AUO \times AUV + 1 \]  
(AU 22)

Výpočet multiplikátoru, vyjádřujícího vliv autonomního nervového systému na srdeční frekvenci (AUR). Příslušný citlivostní koeficient (AUS=1) vyjadřuje sílu, kterou se odchylka od normálního tonu sympatiku (vyjádřená hodnotou proměnné AUO) projeví na změně hodnoty multiplikátoru ovlivňujícího srdeční frekvenci:

\[ AUR = AUO \times AUS + 1.0 \]  
(AU 23)

Výpočet multiplikátoru, vyjádřujícího vliv autonomního nervového systému na exkreci ADH a na řízení některých cirkulačních funkcí (AUP). Příslušný citlivostní koeficient (AUQ=1) vyjadřuje sílu, kterou se odchylka od normálního tonu sympatiku (vyjádřená hodnotou proměnné AUO) projeví na změně hodnoty
multiplikátoru ovlivňujícího exkreci ADH a některé další cirkulační funkce (na hodnotě multiplikátoru AUP závisí i další multiplikátory (AOM, VVR, AUM a AVE) počítané v dále:

\[ AUP = AUO \times AUQ + 1.0 \]  
(AU 24)

Výpočet multiplikátoru, vyjadřujícího vliv autonomního nervového systému na spotřebu kyslíku ve svalech a nesvalové tkání (AOM). Příslušný citlivostní koeficient (O2A=0.15) vyjadřuje sílu, kterou se odchylka od normálního tonu sympatiku (vyjádřená hodnotou AUP -1) projeví na změně hodnoty multiplikátoru AOM:

\[ AOM = (AUP - 1.0) \times O2A + 1.0 \]  
(AU 25)

Výpočet vlivu autonomního systému na reziduální objem krve v cévním systémovém venózním řečišti, který neroztažuje cívě (maximální objem při nulovém tlaku). Zvýšení aktivity sympatiku tento reziduální objem (VVR) sníží. Od základního objemu, který je dán anatomickými vlastnostmi individuálně pro každého člověka (VV9=3.16) se odcítí snížení tohoto objemu působením napětí cévní venální štěny ovlivňované tonem sympatického nervového systému (VVRDM). Toto snížení závisí na odchylce od bazálního tonu sympatiku (AUO) a citlivostního koeficientu (AUL=0.21). Zvýšení tonu sympatiku se tak projeví zvýšením hodnoty VVRDM a snížením hodnory VVR:

\[ VVRDM = 0.21 \times AUO \times AUL \]  
(AU 26)

\[ VVR = VV9 - VVRDM \]  
(AU 27)

Výpočet multiplikátoru, vyjadřujícího vliv autonomního nervového systému na arteriální vasokonstrukci a následné zvýšení rezistence v arteriálním (a arteriolárním) řečišti (AUM):

\[ AUM = 0.85 \times AUP + 0.15 \]  
(AU 28)

Výpočet multiplikátoru, vyjadřujícího vliv autonomního nervového systému na venální vasokonstrukci a následné zvýšení rezistence ve venálním řečišti (AVE), citlivostní koeficient (AUY=0.25) vyjadřuje citlivost venální vasokonstrukční stimulace vzhledem ke změnám v autonomní stimulaci arteriálního řečiště:

\[ AVE = (AUM - 1.0) \times AUY + 1.0 \]  
(AU 29)
**Autonomic Control of the Circulation**

**INPUTS:**
- PA - systemic arterial pressure (torr)
- POT - non-muscle cell PO2 (torr)
- EXC - effect of exercise on the metabolic usage of oxygen by the muscles (ratio to resting state)
- ANUBR - direct effect of angiotensin on vasomotor center ("virtual" torr)
- AUZ - overall sensitivity of autonomic control (ratio to normal)

**OUTPUTS:**
- AU - overall activity of autonomic system (ratio to normal)
- AUH - autonomic multiplier effect on heart output (ratio to normal)
- AUR - autonomic multiplier effect on heart rate (ratio to normal)
- VVR - basic volume of venous tree (maximum volume at zero pressure) [l]
- AUP - autonomic multiplier effect on ADH hormone excretion etc. (ratio to normal)
- AOM - autonomic multiplier effect on tissue oxygen utilisation (ratio to normal)
- AUM - autonomic multiplier effect on arterial resistance (ratio to normal)
- AVE - autonomic multiplier effect on venous resistance (ratio to normal)

**Formulas:**
- \( \text{AU} = (\text{AUJ} - 1) \times \text{AUZ} + 1 \)
- \( \text{AUH} \) calculation:
  - \( \text{PA} < 50 \): \( \text{AUH} = 6 \)
  - \( 20 > \text{PA} < 50 \): \( \text{AUH} = 0.2 \times (50 - \text{PA}) \)
  - \( \text{PA} \geq 50 \): \( \text{AUH} = 0 \)
- \( \text{AUR} \) calculation:
  - \( \text{PA} < 50 \): \( \text{AUR} = 1.85718 \)
  - \( 40 > \text{PA} < 170 \): \( \text{AUR} = 0.014286 \times (170 - \text{PA}) \)
  - \( \text{PA} \geq 170 \): \( \text{AUR} = 0 \)
- \( \text{VVR} \) calculation:
  - \( \text{PA} < 80 \): \( \text{VVR} = 2.95 \)
  - \( \text{PA} \geq 80 \): \( \text{VVR} = 0 \)

**Variables:**
- \( \text{PA} \) - arterial pressure
- \( \text{POT} \) - non-muscle cell PO2
- \( \text{PMO} \) - muscle cells PO2
- \( \text{EXC} \) - effect of exercise on metabolic usage of oxygen
- \( \text{ANUBR} \) - direct effect of angiotensin on vasomotor center
- \( \text{AUZ} \) - overall sensitivity of autonomic control
- \( \text{AU} \) - overall activity of autonomic system
- \( \text{AUH} \) - effect on heart output
- \( \text{AUR} \) - effect on heart rate
- \( \text{VVR} \) - basic volume of venous tree
- \( \text{AUP} \) - effect on ADH hormone excretion etc.
- \( \text{AOM} \) - effect on tissue oxygen utilisation
- \( \text{AUM} \) - effect on arterial resistance
- \( \text{AVE} \) - effect on venous resistance

**Diagrams:**
- Flowchart for the computation of \( \text{AU}, \text{AUH}, \text{AUR}, \) etc.
- Block diagrams for the calculation of \( \text{PA1AUN} \) and \( \text{PA1AUC} \) with input conditions.
ERYTROCYTY A VISKOZITA KRVE

Regulace tvorby a zániku erytrocytů

Celková velikost objemu erytrocytů v organismu (VRC) je dána integrací rozdílu (RCD) mezi rychlostí jejich tvorby (RC1) a rychlostí jejich zániku (RC2):

\[ RCD = RC1 - RC2 \]  
\[ VRC = \int RCD \, dt \]

Doba životnosti erytrocytů je průměrně 120 dní. Tento model však dobu životnosti erytrocytů zatím neuvádí. Rychlost zániku erytrocytů (RC2) je v původním klasickém modelu dle Guytona z roku 1972 počítána přímo úměrně z celkového objemu erytrocytů. Koeficient úměrnosti (RKC) má hodnotu 0.0000058. Guyton v roce přidal ještě další parametr – viskozitu krve, jako relativní hodnotu relativní k normě (VIM=1), čím vyšší je viskozota krve, tím rychleji zanikají erytrocyty:

\[ RKC = 58 \times 10^{-7} \]  
\[ RC2 = RKC \times VIM \times VRC \]

Rychlost tvorby erytrocytů, tj. erytropoezy (RC1) závisí na tvorbě erytropoetinu. Erytropoetin se tvoří v převážné části v ledvinách. Regulačním stimulem je parciální tlak kyslíku v tkání ledvin – ten závisí na koncentraci kyslíku v arteriální krvi a průtoku krve ledvinami (tj. na přítoku kyslíku do ledvin) a na jeho spotřebě v ledvinách. Protože spotřeba kyslíku v ledvinách klesá při poklesu krevního přítoku, pak hlavním regulátorem bude celková koncentrace kyslíku v arteriální krvi.

V původním Guytonově modelu byla rychlost tvorby erytrocytů závislá na parciálním tlaku kyslíku v nesvalové tkáni (POT) – normální hodnota je 8 torr, počítá se rozdíl (DPOT) od konstantní hodnoty (POT1=8.25 torr). Čím nižší je parciální tlak kyslíku, tím je hodnota DPOT vyšší. Hodnota DPOT je ohraničena zdola i zhora, a rychlost tvorby erytrocytů je úměrná hodnotě DPOT, koeficient úměrnosti, reprezentující citlivost POY=

\[ DPOT = POT - POT1 \]
\[ když \, DPOT < 0.2375 \, pak \, DPO = 0.2375 \]
\[ RC1 = DPOT \times POY \]

Guyton v modelu z roku 1986 učinil rychlost tvorby erytrocytů úměrnou odchylce saturace erytrocytů v arteriální krvi (OSA) od normální hodnoty (0.97) a nepřímo úměrné průtoku erytrocytů v ledvinách, tj. průtoku krve ledvinami (RFN) a hematokritu (HM), vyjádřeného zde v procentech. Dalším multiplikačním koeficientem je podíl normálně fungující části ledvin (REK=1), aby bylo možno modelovat poruchu ledvin. Koeficient úměrnosti je 0.0003568. Výslednou hodnotou rychlosti tvorby erytrocytů je ohraničena zdola i shora:

\[ RC1 = 0.0003568 \times (1 + 0.9 \times (0.97 - OSA)) \times REK \times RFN \times HM \]
\[ když \, RC1 < 0.000004 \, pak \, RC1 = 0.000004 \]
\[ když \, RC1 > 0.00014 \, pak \, RC1 = 0.00014 \]
Celkový objem erytrocytů v organismu (VRC) je určována rozdílem mezi rychlostí tvorby (RC1) a zániku (RC2) erytrocytů:

\[ RCD = RC_1 - RC_2 \]  

\[ VRC = \int RCD \, dt \]  

Z celkového objemu erytrocytů (VRC) a objemu plazmy (VP) vypočítáme celkový objem krve (VB) a hematokrit vyjádřený jako poměrné číslo od 0-1 (HM1), nebo v procentech (HM):

\[ VB = VP + VRC \]  

\[ HM_1 = VRC/VB \]  

\[ HM = HM_1 \times 100 \]

Tento modul kinetiky erytropoézy a zániku erytrocytů nepočítá se zpožděním účinku erytropoetinu, ani s délku života erytrocytů. V budoucnu bude nutno tuto část modelu v tomto smyslu zásadně přepracovat.

Závislost viskozity (VIM) na hematokritu (HM)

Z hematokritu (HM), vyjádřeném v procentech je pomocí koeficientů (HMK=90) a (HKM=0.53333) v empirické rovnici vypočítána hodnota viskozity krve připadající na erytrocyty (vyjádřená poměrem viskozity k vodě):

\[ VIE = HM/((HMK-HM) \times HKM) \]  

\[ VIE = HM/((HMK-HM) \times HKM) \]  

\[ VIE = HM/((HMK-HM) \times HKM) \]  

\[ VIE = HM/((HMK-HM) \times HKM) \]  

\[ VIE = HM/((HMK-HM) \times HKM) \]
Z viskozity erytrocytů (VIE) a konstanty reprezentující viskozitu plazmy je spočítána viskozita krve (VIB) jako číslo poměrně k viskozité vody.

\[ VIB = VIE + 1.5 \quad \text{(RC 16)} \]

Nakonec je počítána normalizovaná viskozita krve (VIM), jako poměr viskozity krve (VIB) k normální hodnotě viskozity:

\[ VIM = \frac{VIB}{3.0} \quad \text{(RC 17)} \]
BLOOD VISCOSITY

INPUT:
- HM - hematocrit [%]
- VIM - blood viscosity (ratio to normal)

OUTPUT:
- VIM - blood viscosity (ratio to normal)

RED CELLS PRODUCTION AND DESTRUCTION

INPUTS:
- OSA - arterial oxygen saturation (ratio to full saturation)
- VP - plasma volume [l]
- VIM - blood viscosity (ratio to normal blood)
- RFN - renal blood flow (if kidney is not damaged) [l/min]
- REK - fraction of normal kidney mass (ratio to normal)

OUTPUTS:
- VRC - volume of red blood cells [l]
- VB - volume of blood [l]
- HM - hematocrit [%]
- RC1 - red cells production rate [l/min]
- RC2 - red cells destruction rate [l/min]

Red cells production and destruction
DYNAMIKA KAPILÁŘ, INTERSTICIÁLNÍ TEKUTINY, PLAZMATICKÝCH 
A TKÁŇOVÝCH PROTEINŮ

Výpočet středního kapilárního tlaku

Průtok krve malými vénami měkkých neledvinných (a nesvalových) tkání (BFN) vynásobený 
koefficitem 1,79 zvětší tento průtok o normální průtok ledvinami a svalovou tkání, vynásobíme-li ho pak 
rezistenci malých vén (RSV) dostaneme tlakový spád mezi tlakem ve velkých žilách (PVS) a středním 
kapilárním tlakem (PC) se rovná průtoku krve malými vénami vynásobeným rezistenci malých vén (RVS):

\[ PC - PVS = RVS \times 1.79 \times BFN \]  

CP 01

Na tomto vztahu je založen výpočet středního kapilárního tlaku, který se počítá v rovnici (HD 54) v 
kapitole věnované výpočtu rezistencí v systémové cirkulaci.
Systemic Capillary Pressure

**Inputs:**
- PVS - average of systemic venous pressure [torr]
- BFN - blood flow in non-renal and non-muscle tissues [l/min]
- RVS - resistance in small veins [torr.min/l]

**Output:**
- PC - systemic capillary pressure [torr]
Starlingova rovnováha na kapiláre

Tlakový gradient mezi kapilárou a intersticiální tekutinou (PCGR) se rovná rozdílu mezi tlaky, které ženou tekutinu z kapiláry do intersticia a protitlaky, které nasávají tekutinu z tkání do kapiláry – tj. rozdíl mezi středním tlakem v kapiláře (PC) a koloidně-osmotickým tlakem v tkáňovém gelu (PTC), protáhujícími tekutinu z kapiláry a plazmatickým koloidně-osmotickým tlakem (PPC) a hydraulickým protitlakem tkáňového gelu (PGH), nasávajícími tekutinu do kapiláří:

\[ PCGR = PC - PPC - PGH + PTC \]  
(CP 02)

Rychlost kapilární filtrace (CFILTR) je úměrná tlakovému gradientu mezi kapilárou a intersticiem (PCGR), koeficientem úměrnosti je zde kapilární filtrační koeficient (CFC), zahrnující v sobě rezistenci kapilární stěny a celkový povrch kapiláří:

\[ CFILTR = PCGR \times CFC \]  
(CP 03)

Celková rychlost přestupu tekutiny z kapiláry do intersticiálního prostoru ze systémových kapilár (VTC) se rovná kapilární filtrace (CFILTR) plus průsaku tekutiny zkrze "děrává kapiláry" (VTCPL):

\[ VTC = CFILTR + VTCPL \]  
(CP 04)

Průsak plazmy z kapiláry do tkáňového intersticia zkrze póry v kapilární emmbráně závisí na tlakovém gradientu (PRCD) který se rovná rozdílu mezi středním tlakem v kapiláře (PC) a kritickém kapilárním tlaku, při jehož překročení se kapilární póry otevírají (PCR=15). Pokud je střední kapilární tlak nižší než kritický kapilární tlak, k žádnému průsaku plazmy nedochází (a příslušný tlakový gradient PRCD se rovná nule):

\[ PCR = 15.0 \]  
(CP 05a)

\[ PRCD = PC - PCR \]  
(CP 05b)

\[ když \ (PRCD < 0.0) \ \text{a} \ \text{PCR} = 15.0 \]  
(CP 05c)

Rychlost průsaku plazmy skrze kapilární póry (DCP) je závislá na výše vypočteném tlakovém gradientu (PRCD) dle empirického vztahu, kde konstanty CPK znamenají vodivost a konstanta PCE (za normálních okolností ale rovná 1) vyjadřuje možnou nelinearitu tohoto vztahu:

\[ VTCPL = (PRCD \times CPK) \times PCE \]  
(CP 06c)

Rychlost změny objemu plazmy (VPD) se rovná rozdílu mezi celkovým tokem lymfy do plazmy (VTL), rychlostí přítoku vody z GIT nebo odjinud zvětšením tkáního organismu (TVD) a odtokem tekutiny ze systémových kapilár do systémového intersticia (VTC), odtokem tekutiny z plicních kapilár do plicního intersticia (DFP) a diurézou (VUD):

\[ VPD = VTL + TVD - VUD - VTC - DFP \]  
(CP 07)

Celkový objem plazmy (VP) pak odbíráme integrací rychlosti změny tohoto objemu (VPD):

\[ VP = \int VPD \, dt \]  
(CP 08)

Relativní objem intersticia (VPREL) vzhledem k normě (VPN) pro dané individuum bude:

\[ VPREL = VP / VPN \]  
(CP 09)
Rychlost změny objemu intersticiální tekutiny v systémovém intersticiálním prostoru (VTD) se rovná celkové rychlosti přestupu tekutiny ze systémových kapilár do intersticia (VTC) minus odtok tekutiny z intersticia do krve přes lymfu (VTL) minus přestup vody z intersticia do buněk (VID):

\[ V_{TD} = V_{TC} - V_{TL} - V_{ID} \]  \hspace{1cm} (CP 10)

Celkový objem systémového intersticiálního prostoru (VTS) pak odbíráme integrací rychlosti změny tohoto objemu (VTD):

\[ V_{TS} = \int V_{TD} \, dt \]  \hspace{1cm} (CP 11)

Relativní objem systémového intersticiálního prostoru (VTSREL) se počítá vzhledem k normě pro individuum dané tělesné konstituce:

\[ V_{TSREL} = \frac{V_{TS}}{V_{TSN}} \]  \hspace{1cm} (CP 12)
Dynamika plazmatických proteinů

Konzentrace proteinů v plazmě (CPP) závisí na množství proteinů v plazmě (PRP) a objemu plazmy (VP):

\[ CPP = \frac{PRP}{VP} \]  

(CP 13)

Koloidně osmotický tlak proteinů v plazmě (PPC) nelineárně závisí na koncentraci proteinů v plazmě (CPP):

\[ PPC = 0.28 \times CPP + 0.0019 \times CPP^2 \]  

(CP 14)

Tok proteinů z plazmy do intersticia (DPC) se skládá z toku proteinů díky průsahu plazmy přes kapilární pory (PLPRL) závislém na rychlosti průsahu plazmy (VTPCL) a koncentraci proteinů v plazmě (CPP) a z toku proteinů přes kapilární membránu (PLPRDF) závislém na koncentračním gradientu hladiny proteinů mezi plazmou (CPP) a intersticiální tekutinou (CPI):

\[ PLPRL = VTPCL \times CPP \]  

(CP 15)

\[ PLPRDF = (CPP - CPI) \times 0.00104 \]  

(CP 16)

\[ DPC = PLPRL + PLPRDF \]  

(CP 17)

Rychlost destrukce proteinů v játrech je nelineárně závislá na koncentraci plazmatických proteinů prostřednictvím faktoru (CPP) vyjadřujícím rozdíl mezi koncentrací proteinů v plazmě (CPP) a kritické hodnoty hladiny proteinů v plazmě (CPR=40 g/dl). Je li rozdíl záporný (tj. koncentrace plazmatických proteinů je nižší než kritická, pak faktor CPPD je nulový:

\[ CPR = 40 \]  

(CP 18a)

\[ CPPD = CPP - CPR \]  

(CP 18b)

když (CPPD<0.0) pak CPPD = 0.0  

(CP 18c)

Výpočet rychlosti destrukce plazmatických proteinů v játrech (LPPRDS) v závislosti na výše uvedeném faktoru (CPPD) je prováděn pomocí empirické rovnice v níž (LPK) a (LPDE) jsou empiricky zjištěné koeficienty na základě porovnání s měřenými daty:

\[ LPK = 0.2728 \times 10^{-13} \]  

(CP 19a)

\[ LPDE = 8.0 \]  

(CP 19b)

\[ LPPRDS = LPK \times CPP^{LPDE} \]  

(CP 19c)

Celková čistá rychlost vyměny proteinů mezi plazmou a játry - (DPL), tj. rychlost syntézy minus rychlost destrukce plazmatických proteinů v játrech, závisí na rozdílu mezi rychlostí syntézy plazmatických proteinů (LPPR=0.03), která je zadaná jako konstantní vstuú a rychlostí destrukce plazmatických proteinů (LPPRDS), počítanou v rovnici CP 19:

\[ LPPR = 0.03 \]  

(CP 20a)

\[ DLP = LPPR - LPPRDS \]  

(CP 20b)

Rychlost změny množství proteinů v plazmě (DPP) se rovná rychlosti syntézy proteinů v játrech (DLP) a rychlosti návratu proteinů do plazmy přes lymfotok (DPL) mínus rychlost přestupu proteinů ze systémových kapilár do intersticia (DPC) minus rychlost toku proteinů z plicních kapilár (PPD) minus rychlost ztrát proteinů z plazmy za patologických okolností (DPR) – tj. ztráty ledvinami nebo při popáleninách.

\[ DPP = DLP + DPL - DPC - PPD - DPR \]  

(CP 21)

Celkové množství proteinů v plazmě (PRP) pak odbražíme integrací rychlostí změny tohoto množství (DPP):
Nelineární závislost celkové rychlosti tvorby/destrukce proteinů v játrech (DLP) na jejich plazmatické koncentraci (CPP)
PLASMA PROTEINS DYNAMICS

INPUTS:

- VP - plasma volume [l]
- VTCLP - rate of leakage of whole plasma through the capillary membrane [l/min]
- DPL - rate of return of protein to the circulation through the lymph [g/min]
- CPI - concentration of protein in the interstitium [g/l] [torr]
- LPPR - rate of production of protein by the liver [g/min]
- PPD - rate of loss of protein through the pulmonary capillary membrane [g/min]
- DPR - rate of loss of protein through the kidney [g/min]

OUTPUTS:

- PPC - plasma colloid osmotic pressure [torr]
- DPC - rate of influx of protein into interstitial fluid from plasma [g/min]
- CPP - plasma protein concentration [g/l]
- DPL - net rate of protein exchange between liver and plasma [g/min]

Plasma Protein Dynamics

\[ \text{CPP} = 0.28 \times \text{CPP} + 0.0019 \times \text{CPP} \]

\[ \text{CPPD1} = \begin{cases} 0 & \text{if } \text{CPPD1} < 0 \\ \text{CPPD1} & \text{else} \end{cases} \]
Dynamika intersticiální tekutiny, intersticiálního tkáňového gelu a intersticiálních proteinů

Voda v intersticiální tekutině se nachází vázaná v gelu a volná. Poměr objemu tekutiny v gelu a volné se mění při změnách celkového objemu intersticia. Při zvětšování celkového objemu intersticiální tekutiny nelineárně stoupá objem gelu a reálnivně přibývá více objemu volné tekutiny. Objem gelu (VG) počítáme v závislosti z celkového objemu intersticiální tekutiny (VTS) podle empirického vztahu aproximující výsledky experimentálních dat pomocí splinů:

\[ VG = \text{function interstitiisGelVolume(VTS)} \]  

(CP 23)

<table>
<thead>
<tr>
<th>VTS</th>
<th>VG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12.0</td>
<td>11.4</td>
</tr>
<tr>
<td>15.0</td>
<td>14.0</td>
</tr>
<tr>
<td>18.0</td>
<td>16.0</td>
</tr>
<tr>
<td>21.0</td>
<td>17.3</td>
</tr>
<tr>
<td>24.0</td>
<td>18.0</td>
</tr>
</tbody>
</table>
Závislost objemu tkáňového gelu (VG) na celkovém objemu intersticia (VTS)

Objem volné tekutiny v intersticiu (VIF) počítáme jako rozdíl celkového objemu (systémového) intersticiálního prostoru (VTS) a objemu tkáňového gelu (VG):

\[
VIF = VTS - VG
\]

(CP 24)

Celkový tlak v tkáních systémového intersticia PTT závisí nelineárně na celkovém objemu intersticia podle empirické rovnice:

\[
VTSF = 6.0
\]

když \( VTS < VTSF \) pak \( PTT = 0 \)

jinak \( PTT = \frac{(VTS - VTSF)}{VTSF}^2 \)

(CP 25a)  
(CP 25b)  
(CP 25c)
Celkový tlak v tkáních intersticia (PTT) je vyšší, než hydraulický tlak uvnitř tkáňového gelu (PPGH), kterým gel např. působí jako tkáňový protitlak při hydraulické rovnováze tekutiny na kapiláře. Příčinou snížení tlaku je hyaluronová kyselina hojně přítomná v tkáňovém gelu. Hyaluronová kyselina působí jako natažená pružina, která snižuje celkový tlak v tkáňovém gelu (koeficient uměrnosti PGHF=3.7). Hydraulický tlak tkáňového gelu (PGH) se proto počítá ze složky celkového tlaku v intersticiu (PTT) od něhož se odečítá "pružinový" tlak (TPGHF) závislý na koncentraci kyseliny hyaluronové (CHY):

\[
TPGHF = CHY \times 3.7 \\
PGH = PTT - TPGHF
\]  

Konzentrace kyseliny hyaluronové v intersticiu (CHY) nelineárně závisí na celkovém množství kyseliny hyaluronové v intersticiu (HYL) a celkovém objemu intersticia (koeficient CMPTSS=2 charakterizuje stupeň této nelinearity):

\[
CMPTSS = 2 \\
CHY = \frac{HYL}{VTS/5.0}^{CMPTSS}
\]
Koloidně osmotický tlak v tkáňovém gelu způsobený hyaluronovou kyselinou (POSHYL) závisí na koncentraci kyseliny hyaluronové:

\[ \text{POSHYL} = \text{CHY} \times 2.0 \]  

(ČP 29)

Proteiny přítomné v tkáňovém gelu jsou ekvilibrovány mezi volnou tektinou v intersticiu a tektinou v tkáňovém gelu (z volné tektiny jsou do gelu "nasávány"). I když je koncentrace proteinů uvnitř tkáňového gelu nižší než ve volné intersticiální tektině, jejich osmotická aktivita (potencovaná kyselinou hyaluronovou) je vyšší. Proto při počítání tlakového rozhraní mezi volnou intersticiální tektinou a tektinou v tkáňovém gelu předpokládáme z hlediska proteinů mezi volnou tektinou a gelem koloidně osmotickou ekvilibraci. Z hlediska vlivu na osmolaritu proto proteinu zanedbáváme, a jediným osmoticky aktivním faktorem ovlivňujícím hydraulické tlaky na rozhraní volná tektina – tkáňový gel počítáme koloidně osmotický tlak kyseliny hyaluronové (POSGYL). Tlak ve volné intersticiální tektině (PIF) se tedy rovná hydraulickému tlaku intersticiálního tkáňového gelu minus nasávaný koloidní osmotický tlak kyseliny hyaluronové v tkáňovém gelu (POSHYL):

\[ \text{PIF} = \text{PGH} - \text{POSHYL} \]  

(ČP 30)

Tlak pevných tkání v intersticiu (PTS) se rovná celkovému tlaku v tkáni (PTT) minus tlak ve volné intersticiální tektině (PIF)

\[ \text{PTS} = \text{PTT} - \text{PIF} \]  

(ČP 31)

Z intersitiálního prostoru je tektina (a v ní rozpuštěné látky včetně proteinů) odváděna lymfou. Lymfatické cévy mají chloupné a proto lymfa nemůže z nich vytéká. Čím vyšší je tlak volné intersticiální tektiny, tím vyšší je tlakový gradient mezi vlastní intersticiální tektinou a lymfatickou cévou. Při odtoku lymfy z tkání se ale musí překonat protitlak intersticia, který lymfatické cévy stlačuje. Tlakový gradient který způsobuje tok lymfy z voně intersticiální tektiny do lymfatických cév (PLD) proto počítáme jako rozdíl součtu

![Diagram Nelineární závislost koncentrace kyseliny hyaluronové v intersticiální tektině (CHY) na celkovém objemu intersticia (VTS)](attachment:image)
konstanty (PLDF=5.9) a tlaku ve volné intersticiální tekutině (PIF) minus celkový tkáňový tlak (PTT). Výsledný tlak je shora ohraničen maximální možnou hodnotou 7.0 torr:

\[
\text{PLDF} = 5.9 \quad \text{(CP 32a)} \\
\text{PLD = PIF + PLDF - PTT} \quad \text{(CP 32b)} \\
\text{když (PLD > 7.0) pak PLD = 7.0} \quad \text{(CP 32c)}
\]

Celkový tok lymfy (VTL) se rovná tlakovému gradientu pro lymfotok (PLD) násobeného konstantou reprezentující vodivost lymfatických cév. Vypočtený tok lymfy je ohraničen zdola hodnotou 0 (lymfa neteče z lymfatických cév do tkání):

\[
\text{VTL} = \text{PLD} \times 0.02 \quad \text{(CP 33a)} \\
\text{když (VTL < 0.0) pak VTL = 0.0} \quad \text{(CP 33b)}
\]

V intersticiu jsou obsaženy proteiny, které jsou ekvilibrovány jak ve volné intersticiální tekutině, tak ve tkáňovém gelu. Pro zjednodušení zde nerozlišujeme mezi koncentracemi proteinů v gelu a ve volné intersticiální tekutině a pro celé intersticium počítáme s jedinou koncentrací proteinů. Koncentrace proteinů v intersticiu (CPI) proto jednoduše počítáme z celkového množství proteinů v tkáňovém intersticiu (TSP) a celkového objemu systémového intersticia (VTS):

\[
\text{CPI} = \frac{\text{TSP}}{\text{VTS}} \quad \text{(CP 34)}
\]

Koloidně-osmotický tlak ve volné tekutině intersticia (PTCPR) se počítá z koncentrace proteinů v intersticiu (CPI) podle stejné empirické rovnice, jakou jsme počítali koloidně osmotický tlak proteinů v plazmě (viz rovnici CP 14):

\[
\text{PTCPR} = 0.28 \times \text{CPI} + 0.0019 \times \text{CPI}^2 \quad \text{(CP 35)}
\]

Jestliže na rozhraní gel-volná tekutina v intersticiu jsme mohly osmotický vliv tkáňových proteinů zanedbat, pak na rozhraní krevní kapiláry – tkáňový gel to již neplatí a při počítání celkového koloidně-osmotického tlaku intersticiálního gelu musíme s proteiny počítať. Osmotický vliv proteinů a kyseliny hyaluronové na celkový koloidně-osmotický tlak tkáňového gelu se vzájemně potenciují, proto vliv osmotického tlaku kyseliny hyaluronové a osmotického tlaku plasmatických proteinů nepočítáme jako součet ale součin. Celkový osmotický tlak intersticiálního tkáňového gelu (PTC) se pak rovná osmotickému tlaku kyseliny hyaluronové v tkáňovém gelu (POSHYL) násobeném koloidně osmotickým tlakem způsobeným plasmatickými proteiny ve volné tekutině intersticia (PTCR), které jsou v ekvilibriu s proteiny v tkáňovém gelu. Konstanta úmernosti je zde (GCOPF = 0.7):

\[
\text{GCOPF} = 0.7 \quad \text{(CP 36a)} \\
\text{PTC} = \text{POSHYL} \times \text{PTCPR} \times \text{GCOPF} \quad \text{(CP 37b)}
\]

Do tkáňového intersticia se proteiny dostávají z plazmy (viz rovnice CP 15-17), a z tkáňového intersticia jsou odváděny zpět do cirkulace lymfou. Rychlost návratu plasmatických proteinů do plazmy prostřednictvím lymfy (DPL) je úmerná koncentraci proteinů v systémovém intersticiu (CPI) a rychlosti toku lymfy z intersticia do plazmy (VTL):

\[
\text{DPL} = \text{CPI} \times \text{VTL} \quad \text{(CP 38)}
\]

Bilance proteinů v intersticiu závisí na rozdílu mezi rychlostí přítoku proteinů z plazmy (DPC), počíataném v rovině CP 17 a rychlostí jejich odvodu z intersticia lymfou (DPL). Rychlost změny celkového množství proteinů v systémovém intersticiu (DPI) se proto rovná:

\[
\text{DPI} = \text{DPC} - \text{DPL} \quad \text{(CP 39)}
\]
Celkové množství proteinů v tkáňovém intersticiu (TSP) pak odbíráme integrací rychlosti změny tohoto množství (DPI):

\[
TSP = \int DPI \, dt
\]

(CP 40)
DYNAZIKA TEKUTIN A PLYNŮ V PLICÍCH

Dynamika tekutin a proteinů v plicích

Jedná se o velmi zjednodušenou anylyzu dynamiky plicních tekutin. Gelová součást plicní tekutiny je zanedbána, takže je plicní tekutina (VPP) aproximována jako tekutina volně mobilní. Obdobně tlakově-objemové křivky plicního intersticia jsou vysoce zjednodušeny, stejně jako i tok lymfy v plicích.

Plicní kapilární tlak (PPA) je počítán jako součet 45% plicního arteriálního tlaku (PPA) a 55% levého plicního atrialního tlaku (PLA):

\[ PCP = 0.45 \times PPA + 0.55 \times PLA \]  
(PD 01)

Tlakový gradient na plicní kapiláry (PGRPCM), působící ve směru filtrace tekutiny do plicního intersticia je počítán z plicního kapilárního tlaku (PCP) a koloidně osmotického tlaku plicní intersticiální tekutiny (POS) a protitlaku – tlaku plicní intersticiální tekutiny (PPI) a koloidně osmotického tlaku plazmatických bílkovin (PPC), nasávajícího tekutinu do kapiláry:

\[ PGRPCM = PCP - PPI + POS - PPC \]  
(PD 02)

Z tlakového gradientu na plicní kapiláře (PGRPCM) a z kapilárního filtracího koeficientu (CPF) je počítána rychlost filtrace tekutiny do intersticia plic (PFI):

\[ PFI = PGRPCM \times CPF \]  
(PD 03)

Rychlost změny objemu plicní tekutiny v intersticiu (DFP) se rovná rozdílu mezi rychlostí filtrace z plicních kapilár (PFI) a rychlosti návratu tekutiny z plic do cirkulace lymfy (PLF):

\[ DFP = PFI - PLF \]  
(PD 04)

Integrací rychlosti změny objemu plicní intersticiální tekutiny (DFP) dostaneme objem volné tekutiny v plicním intersticiu (VPF):

\[ VPF = \int DFP \, dt \]  
(PD 06)

Na základě empirického vztahu odpovídající experimentálně naměřené závislosti počítáme hydraulický tlak tekutiny v plicním intersticiu (PPI) z objemu plicního intersticia (VPF):

\[ PPI = 2.0 - 0.150 / VPF \]  
(PD 07)
Závislost hydraulického tlaku v plicním intersticiu (PPP) na celkovém objemu plicního intersticia (VPF)

Rychlost toku lymfy v plicích (PLF) závisí na tlaku intersticiální plicní tekutiny (PPI) podle empiricky zjištěného vztahu odpovídajícího naměřeným závislostem:

\[ PLF = (PPI + 11.0) \times 0.0003 \]  

PD 08

Z toku lymfy (PLF) a koncentrace proteinů v plicním interstici (CPN) můžeme spojit rychlost návratu proteinů z plicního intersticia do cirkulace prostřednictvím lymfy (PPO):

\[ PPO = PLF \times CPN \]  

PD 09

Z rozdílu koncentrací proteinů v plazmě (CPP) a plicním interstici (CPN) počítáme rychlost průsaku proteinů z plazmy do plicního intersticia (PPN):

\[ PPN = (CPP - CPN) \times 0.000225 \]  

PD 10

Rychlost změny obsahu proteinů v interstici plic (PPD) se rovná rozdílu mezi rychlostí příssunu proteinů do intersticia průsakem z plicních kapilár (PPN) a rychlosti jejich odstraňování z intersticia lymfou (PPO):

\[ PPD = PPN - PPO \]  

PD 11

Integraci rychlosti této změny (PPD) odbrzíme celkové množství proteinů v plicním interstici (PPR):

\[ PPR = \int PPD \, dt \]  

PD 12
PULMONARY FLUID DYNAMICS

INPUTS:
- PPA - pulmonary arterial pressure [torr]
- PLA - left atrial pressure [torr]
- PPC - plasma colloid osmotic pressure [torr]
- CPF - pulmonary capillary filtration coefficient [l/min/torr]

OUTPUTS:
- DFP - rate of change of free fluid in the lungs [l/min]
- VPF - volume of free fluid in the pulmonary interstitium [l/min]
- PCP - pulmonary capillary pressure [torr]
- POS - osmotic pressure in pulmonary interstitial fluid pressure [torr]
- PPI - pulmonary interstitial fluid pressure [torr]
- PFI - rate of fluid filtration out of pulmonary capillary [l/min]
- PLF - pulmonary lymph flow rate [l/min]
- CPN - concentration of protein in pulmonary interstitial fluid [g/l]
- PPN - protein leakage rate through the pulmonary capillary membrane [g/min]
- PPR - total quantity of protein in pulmonary interstitial fluid [g]

Pulmonary Fluid Dynamics
Přenos kyslíku v plicích

Celkovou spotřebu kyslíku (O2UTIL) vypočteme jako součet spotřeby kyslíku v nesvalových tkáních (DOB) a ve svalech (RMO):

\[ O2UTIL = DOB + RMO \]  \hspace{1cm} (PD13)

Výpočet saturace kyslíku v arteriální krvi (OSA) z parciálního tlaku kyslíku ve vdechovaném vzduchu (PO2AMB=150 torr), celkové metabolicke spotřeby kyslíku (O2UTIL), normalizovaného difúzního koeficientu pro kyslík přes aleveolokapilární membránu (PO2DEF=1) a volné tekutiny v plicním intersticiu (VPF), která může zhoršit přenos kyslíku při plicním edému:

\[ OSA = \frac{(PO2AMB - (O2UTIL \cdot 0.0266667/PO2DEF))}{100.0} \]  \hspace{1cm} (PD 14)

když \( OSA > 0.9775 \) pak \( OSA = 0.9775 \)  \hspace{1cm} (PD 15)

\[ OSA = OSA - VPF \times 0.6 \]  \hspace{1cm} (PD 16)

Celkovou koncentraci kyslíku v arteriální krvi (OVA) obdržíme vynásobením saturace hemoglobinu kyslíkem (OSA), hematokritem (HM) a konstantou (charakterizující kyslíkovou kapacitu krve):

\[ OVA = OSA \times HM \times 5.0 \]  \hspace{1cm} (PD 17)
### PULMONARY O2 DELIVERY

**INPUTS:**
- **VPF** - volume of free fluid in the pulmonary interstitium [l/min]
- **DOB** - non-muscle oxygen usage [ml/min]
- **RMO** - muscle oxygen usage [ml/min]
- **PO2DEF** - normalised diffusion coefficient for oxygen through the pulmonary membrane [at norma PO2DEF=1]
- **PO2AMB** - ambient oxygen pressure [torr]
- **HM** - hematocrit

**OUTPUTS:**
- **OSA** - arterial oxygen saturation
- **OVA** - arterial blood oxygen content [ml O2/l blood]

---

**PULMONARY OXYGEN DELIVERY**

---

82
**DYNAMIKA ELEKTROLYTŮ A VODY V EXTRACELULÁRNÍ A INTRACELULÁRNÍ TEKUTINĚ**

**Voda, sodík a draslík v ICT a ECT**

Objem extracelulární tekutiny (VEC), vyjádřený v litrech je součtem objemů (systémové) intersticiální tekutiny VTS, plnic intersticiální tekutiny (VPF) a plazmy (VP):

\[
VEC = VTS + VPF + VP \tag{EL 01}
\]

Rychlost přijmu sodíkových iontů \(\text{Na}^+\) do organismu (NAINT), vyjádřená v mmol/min se rovná normální rychlosti přijmu sodíku (NID) (Guyton uvádí hodnotu 0.1 mmol/min) vynásobeným faktorem chuti na slané (STH), který je normálně rovný 1:

\[
NAINT = NID \times STH \tag{EL 02}
\]

Změna množství zásob sodíkových iontů v extracelulární tekutině (NED), vyjádřená v mmol/min se rovná rozdílu mezi rychlostí přijmu sodíkových iontů do organismu (NAINT) a rychlosti ztrát sodíkových iontů moči (NOD):

\[
NED = NAIN T - NOD \tag{EL 03}
\]

Integrací rychlostí této změny (NED) odbržíme zásoby sodíkových iontů (v mmol) v extracelulární tekutině (NAE):

\[
NAE = \int NED \, dt \tag{EL 04}
\]

Koncentrace sodíkových iontů v extracelulární tekutině (a v plazmě – předpokládáme, že koncentrace \(\text{Na}^+\) v systémovém a plnicím intersticiu a v plazmě jsou vyrovnány) - (CNA) obdržíme po vydělení zásob sodíkových iontů (NAE) v extracelulární tekutině objemem extracelulární tekutiny (VEC):

\[
CNA = \frac{NAE}{VEC} \tag{EL 05}
\]

Rychlost změn zásob draselných iontů v extracelulární tekutině (KED), vyjádřená v mmol/min se rovná rychlosti přijmu draselných iontů (KID) - Guyton uvádí normální hodnoty 0.6 mmol/min, minus rychlost přestupu draselných iontů z extracelulární do intracelulární tekutiny (KCD), minus rychlost exkrece draselných iontů do moči (KOD):

\[
KED = KID - KCD - KOD \tag{EL 06}
\]

Integrací této změn obdržíme celkové zásoby draselných iontů \(\text{K}^+\) v extracelulární tekutině (KE), vyjádřené v mmol:

\[
KE = \int KED \, dt \tag{EL 07}
\]

Koncentraci draselných iontů (v mmol/l) v extracelulární tekutině (a v plazmě – předpokládáme, že koncentrace \(\text{K}^+\) v systémovém a plnicím intersticiu a v plazmě jsou vyrovnány) - (CKE) obdržíme po vydělení zásob draselní iontů (KE) v extracelulární tekutině objemem extracelulární tekutiny (VEC):

\[
CKE = \frac{KE}{VEC} \tag{EL 08}
\]

Distribuce draslíku mezi buňkou a extracelulární tekutinou závisí mimo jiné na hladině aldosteronu. Zvýšení hladiny aldosteronu zvyšuje aktivitu \(\text{Na}^+/\text{K}^+\) pumpy a vede k částečnému přesunu draselných iontů.
iónů z extracelulární do intracelulární tekutiny. Aldosteron ovlivňuje pozice část draselných íonů v buňce, větší část závisí na metabolických procesech uvnitř buňky (a je vázana např. na některé intracelulární polymery, jako je např. glykogen). V modelu se při výpočtu cílové hladiny intracelulárních zásob draslíku (KIR) proto rozlišují dvě části zásob, jedna je v modelu brána jako konstantní (KE2=2850 mmol), ovlivňovaná metabolickými procesy, a druhá (KE1) je proměnná, která závisí na koncentraci extracelulárního draslíku a na koncentraci aldosteronu.

Vypočítáme nejprve hodnotu části nitrobuněčných zásob draslíku závislých na koncentraci extracelulárního draslíku a hladině aldosteronu (KE1). Normální hodnota této části nitrobuněčných zásob draslíku (KE1N) je za normálních okolností 140 krát větší, než je koncentrace extracelulárního draslíku.

\[ KE1N = RKIE \cdot CKE \] (EL 00)

Multiplikační koeficient RKIE (za normálních okolností RKIE=140) vyjadřuje přes náležitý objem intracelulární tekutiny (VICnorm=25), tím pak dostaneme výraz umožňující počítat s přenosem draslíku i u jedinců s různou konstituci a váhou (a různou náležitou hodnotou objemu intracelulární tekutiny):

\[ RKIE = 5.6 \cdot VICnorm \] (EL 10)

Zvýší-li se hladina aldosteronu, zvýší se obsah draslíku v buňkách. Hodnotu části nitrobuněčných zásob draslíku, závislou na koncentraci extracelulárního draslíku a hladině aldosteronu (KE1) vypočítáváme z náležitě hodnoty těchto zásob odpovídající momentální extracelulární hladině draslíku (KE1N) a multiplikačním koeficientem vyjadřujícím vliv aldosteronu (AMK), citlivostního koeficientu vyjadřující citlivost vlivu aldosteronu na změnu nitrobuněčných zásob (ACLK):

\[ KE1 = (AMK-1) \cdot ACLK + 1.0 \cdot KE1N \] (EL 11)

Druhá, z hlediska množství podstatnější část zásob draslíku, závislá na metabolických procesech (KE2) je v modelu konstanta. Její hodnotu odvozujeme od náležité hodnoty celkových zásob draslíku v buňkách (KInorm), který počítáme z náležité hodnoty objemu intracelulární tekutiny (VICnorm) vynásobenou normální hodnotou intracelulární koncentrace draslíku (142 mmol/l):

\[ KInorm = 142 \cdot VICnorm \] (EL 12)

Hodnotu části zásob draslíku závislou na metabolických procesech (KE2) vypočítáme tak, že od hodnoty náležité hodnoty zásob draslíku (KInorm) odečteme normální hodnotu druhé části zásob draslíku ovlivňovanou aldosteronem a hladinou extracelulárního draslíku – proto multiplikační koeficient (RKIE) vynásobíme normální hodnotou koncentrace extracelulárního draslíku, tj. 5. Za normálních okolností, při bude hodnota KE2=2850 mmol:

\[ KE2 = KInorm - RKIE \cdot 5 \] (EL 13)

Součet hodnot zásob draslíku závislých na aldosteronu a na hladině draslíku v extracelulární tekutině (KE1) a zásob draslíku závislých jen na metabolických procesech b buňkách (KE2) je cílová hodnota množství zásob draslíku v buňkách:

\[ KIR = KE1 + KE2 \] (EL 14)

Rychlost toku draslíku z extracelulární tekutiny do buněk (KCD) je úmerná rozdílu (KIE) mezi cílovou hodnotou zásob draslíku v buňkách (KIR) a momentální hodnotou zásob draslíku v buňkách (KI). Pro prevenci oscilací nepočítáme rychlost toku draslíku KCD z rozdílu mezi momentální a cílovou hodnotou zásob draslíku (KIE) přímo, ale využíváme zde tlumící člen:

\[ KIE = KIR - KI \] (EL 15)

\[ KCZ = KIE \cdot 0.013 \] (EL 16)
\[ DKCZ = \frac{(KCZ - KCD)}{5.0} \quad (EL\ 17) \]

\[ KCD = \int DKCZ\ dt \quad (EL\ 18) \]

Integrací rychlosti toku draslíku do buněk (KCD) pak získáme hodnotu zásob draslíku v buňkách:

\[ KI = \int KCD\ dt \quad (EL\ 19) \]

Konzentrace intracelulárního draslíku (CKI) počítáme z objemu intracelulární tekutiny (VIC) a celkových zásob draslíku v buňkách (KI):

\[ CKI = \frac{KI}{VIC} \quad (EL\ 20) \]

Rozdíl koncentrací elektrolytů mezi buňkou a extracelulární tekutinou (CCD) je počítán jako rozdíl mezi koncentrací draslíných iontů v buňkách (CKI) a koncentrací sodíkových iontů v extracelulární tekutině (CNA). Předpokládá se že koncentrace ostatních elektrolytů na obou stranách membrány je přibližně stejná a že osmotické popyby vody závisí na rozdílu koncentrací intracelulárního draslíku a extracelulárního sodíku:

\[ CCD = CKI - CNA \quad (EL\ 21) \]

Rychlost osmotických přesunů vody mezi extracelulární tekutinou a buňkou (VID) je úměrná rozdílu koncentrací elektrolytů v buňce a v extracelulární tekutině (CCD). Pro prevenci oscilací při výpočtu VID zavádíme tůmivý člen:

\[ VIZ = 0.01 \times CCD \quad (EL\ 22) \]

\[ DVID = (VIZ - VID) / 5.0 \quad (EL\ 23) \]

\[ VID = \int DVID\ dt \quad (EL\ 24) \]

Integrací rychlosti přesunu vody z extracelulární tekutiny do buněk (VID) získáme celkový objem intracelulární tekutiny (VIC):

\[ VIC = \int VID\ dt \quad (EL\ 25) \]

Celkový objem tělesné vody (VTW) je součet intracelulární (VIC) a extracelulární tekutiny (VEC):

\[ VTW = VIC + VEC \quad (EL\ 26) \]
ECF AND ICF ELECTROLYTES AND VOLUMES

INPUTS:
- VP - plasma volume [l]
- VTS - systemic interstitial fluid volume [l]
- VPF - pulmonary interstitial fluid volume [l]
- STH - salt appetite multiplier factor (ratio to normal)
- NDD - rate of excretion of sodium in the urine [mmol/min]
- KDD - rate of excretion of potassium in the urine [mmol/min]
- AMK - aldosterone multiplier factor for the effect of aldosterone on the transport of potassium
- ACLK - sensitivity control of the effect of aldosterone on cellular membrane transport of potassium (ratio to normal effect)
- VICNorm - normal value of the intracellular fluid volume [l]

OUTPUTS:
- VEC - volume of extracellular fluid [l]
- VIC - volume of intracellular fluid [l]
- VTW - total body water [l]
- VID - rate of fluid transfer between interstitial fluid and cells [l/min]
- NAE - total extracellular sodium content [mmol]
- CNA - extracellular sodium concentration [mmol/l]
- KCE - extracellular potassium concentration [mmol/l]
- KI - total intracellular potassium content [mmol]
- CKI - intracellular potassium concentration [mmol/l]
- KCD - rate of transfer of potassium from the interstitial fluid into the intracellular fluid [mmol/min]

Equations and variables are not shown here but are typically defined in the context of the model and relevant physiological principles.
Celkové množství močoviny v organismu (PLUR) je určováno rozdílem mezi rychlostí metabolické tvorby močoviny (URFORM - za normálních okolností URFORM=0.24 mmol/min) a rychlostí její eliminace moči (UROD):

\[
DPLUR = (URFORM - UROD) \quad (UR \ 01)
\]

\[
PLUR = \int DPLUR \, dt \quad (UR \ 02)
\]

Močovina snadno prochází buněčnými membránami – její koncentrace je prakticky stejná v buňkách i v extracelulární tekutině. Koncentrace močoviny (PLUR) je počítána z celkového množství močoviny (PLUR) a objemu celkové tělesné vody (VTW):

\[
PLURC = \frac{PLUR}{VTW} \quad (UR \ 03)
\]
FUNKCE LEDVIN


Perfúze ledvin

Perfúzní tlak (PAR) – tj. tlak v renální artérii před jejím vstupem do ledvin se počítá z arteriálního systémového tlaku od kterého se odečte tlakový gradient (GLB), způsobený kontrakcí renální artérie (normálně není renální artérie komprimována a tento gradient je nulový). Tento blok umožňuje modelovat renální hypertenzii způsobenou vaskonstrikcí přívodní artérie (Goldbladtovu hypertenzii)⁶:

\[ PAR = PA - GBL \]  
(KD 01)

Průtok krve (dvěma) normálními ledvinami (RFN) se rovná gradientu mezi perfúzním tlakem (PAR) a tlakem v systémových žilech (PVS) dělenému celkovou intrarenální rezistenci (RR):

\[ RFN = \frac{(PAR-PVS)}{IRR} \]  
(KD 02)

Aktuální průtok krve ledvinami (RBF) se rovná renálnímu průtoku zdravými ledvinami (RFN) násobenému faktorem (REK) reprezentujícím podíl funkční tkánek ledvin k normě (u zdravých ledvin REK=1):

\[ RBF = REK \times RFN \]  
(KD 03)

Celková intrarenální rezistence (IRR) se v tomto modelu počítá jako součet rezistence v aferentní (AAR) a eferentní (EAR) aretriole. Zanedbává se zde odpor zbývající části aferentní (vasa recta a renální žíly) – jejich

⁶ Ve fortranském výpisu Guytonova modelu z roku 1986 jsou uvažovány ještě další bloky, které umožňují modelovat perfúzní tlak (PAR) jako na systémovém arteriálním tlaku (PA) nezávislou veličinu a umožnit tak simulaci případů, kdy změněný perfúzní tlak se v ledvinách v dlouhodobém časovém úseku stává nezávislým na arteriálním tlaku a asymptoticky se přiblížuje k normální hodnotě 100 torrů. Proto do modelu zavedl vstupní parametr RASP, který, pokud má nenulovou hodnotu, reprezentuje hodnotu perfúzního tlaku:

\[ \text{když } RASP > 0 \text{ pak } PAR = RASP \]

Další blok umožnil simulovat asymptotické přiblížování změněných hodnoty perfúzního tlaku k normální hodnotě 100 torrů. Blok se zapínal dalším přepínačem:

\[ \text{když } RASP1 > 0 \text{ pak } PAR = PARI \]

Hodnota perfúzního tlaku (PARI) se z hodnoty okolního arteriálního tlaku postupně přiblížovala k hodnotě závislé na koefficinu RCFDPC (když RCFDPC=0, pak se výsledná hodnota přiblížila 100 torrům) rychlosti závislé na hodnotě koefficinu RCFDFP (Guyton zde používal hodnotu RCFDFP=2000):

\[ DPAR = ((100.0 + (PA - 100.0) \times RCFDFP) - PARI) / RCFDFP \]

\[ PARI = \int DPAR \, dt \]

V implementaci modelu v Simulinku jsme tyto bloky vynechali, protože experimenty se vstupními hodnotami je možné provádět mnohem pohodlnějšími prostředky.
RENAL PERFUSION

**Inputs:**
- PA - systemic arterial pressure [torr]
- GLB - pressure drop in renal artery caused by renal arterial constriction [torr]
- PVS - average systemic venous pressure [torr]
- AAR - resistance in afferent arteriole [torr min/ll]
- EAR - resistance in efferent arteriole [torr min/ll]
- REK - percent of normal renal function [ratio to normal]

**Outputs:**
- PAR - renal perfusion pressure [torr]
- RFN - renal blood flow if kidney is not damaged [l/min]
- RBF - renal blood flow [l/min]
- RR - renal resistance [torr min/l]

\[ IRR = AAR + EAR \]  \hspace{1cm} (KD 04)

Celkovou renální rezistenci (zahrnující i případnou obstrukci v renální artérii i omezení ledvinné tkáně, charakterizované vstupním parametrem REK) spočítáme z celkového průtoku krve ledvinami (RBF) a gradientu systémového arteriálního tlaku a systémového venózního tlaku (PVS):

\[ RR = (PAS - PVS) / RBF \]  \hspace{1cm} (KD 05)

http://physiology.umc.edu/themodelingworkshop/Model%20Library/Kidney%20Model/Kidney%20Model_HTML
Aferentní arteriola

Výstupem z bloku je rezistence aferentní arterioly, na níž v podstatné míře závisí glomerulární filtrace. Rezistence je řízena jak nervovými vlivy, hormonálními vlivy (angiotenzinem), tak i signály z macula densa, která zprostředkovává tubuloglomerulární zpětnou vazbu - zvýšení glomerulární filtrace vede ke zvýšené filtraci sodíkových a chloridových iónů, a k jejich následné zvýšené koncentrace ve vtoku do distálního tubulu, kde je tato zvýšená koncentrace detekována buňkami macula densy, důsledkem je signál, který vede ke zpětnovazebnému zvýšení rezistence v aferentní arteriole a následnému snížení glomerulární filtrace.

Rezistence aferentní arterioly (AAR) se počítá z její normální hodnoty (AARK=40 torr) vynásobením multiplikátorů vyjadřujících vliv sympatické nervové inervace (AUMK), vliv angiotenzinu (ANMAR), vliv tubuloglomerulární zpětné vazby prostřednictvím macula densa (RNAUG1), vliv myogenní odpovědi cév na tlak (MYOGRSAA) a multiplikátor vyjadřující relativní viskozitu krve (VIM). Výsledná hodnota je zdola ohraničena hodnotou minimální rezistence AARL=18 torr min/l, odpovídající maximálně roztažení aferentní arterioly:

\[ AAR = AUMK \times RNAUG1 \times ANMAR \times MYOGRSAA \times VIM \times AARK \]  

(kD 06)

když \( AAR < AARLL \) pak \( AAR = AARLL \)  

(kD 07)

Vliv angiotenzinu na rezistenci aferentní arterioly vyjadřuje faktor (ANMAR), který je počítán z vlivu angiotenzinu na cévní rezistenci (ANM), vyjádřeném poměrem účinku momentální hladiny angiotenzinu k normě a citlivostního koeficientu vlivu angiotenzinu na rezistenci aferentní arterioly (ANMAM=0.5). Výsledná hodnota ANMAR je ohraničena zdola nejnižší povolenou hodnotou (ANMARL=0.86):

\[ ANMAR = (ANM-1.0) \times ANMAM + 1.0 \]  

(kD 08)

když \( ANMAR < ANMARL \) pak \( ANMAR = ANMARL \)  

(kD 09)

Vliv sympatické nervové inervace na rezistenci aferentní arterioly je vyjadřen multiplikačním faktorem (AUMK). Jeho hodnota se počítá z multiplikačního faktoru (AUM), vyjadřující obecný vliv sympatiku na cévní rezistenci. Výkyvy sympatického tonu jsou zesíleny pomocí zesilovacího koeficientu (ARF=1.5)

\[ AUMK = (AUM-1.0) \times ARF + 1.0 \]  

(kD 10)

když \( AUMK < 0.8 \) pak \( AUMK = 0.8 \)  

(kD 11)
Afferent Arteriole

**Inputs:**
- AARK - normal resistance in afferent arteriole [torr min]
- RNAUG1 - Macula densa feedback signal [ratio to normal]
- AUM - Sympathetic vasoconstrictor effect on arteries [ratio to normal]
- ANM - Angiotensin multiplier effect on vascular resistance [ratio to normal]
- VIM - blood viscosity [ratio to normal blood]

**Outputs:**
- AAR - afferent arteriolar resistance [torr l/min]
Eferentní arteriola

Rezistence eferentní arterioly má, spolu s rezistencí aferentní arterioly, význam pro regulaci tlaku v gomerulární kapiláře a tím i regulaci glomerulární filtrace. Důležitější význam má zde však aferentní arteriola. Na rozdíl od aferentní arteriole, při výpočtu rezistence v eferentní arteriole není v modelu uvažováno jak vliv sympatické inervace tak i myogenní regulace (v Guytonově modelu z roku 1986 je myogenní regulace formálně uvažována, avšak s multiplikátorem 1, v pozdějším Colemanově modelu dostupném na webu⁸ je tato regulace vypuštěna zcela).

Rezistence eferentní arterioly (EAR) se počítá z její normální hodnoty (EAR%=43.333 torr) vynásobením multiplikátorů vyjadřujících vliv angiotenzinu (ANMER), vliv tubulogomerulární zpětné vazby prostřednictvím macula densa (MDEF) a vliv viskozity krve (VIM), vyjádřené jako relativní hodnota vzhledem k normě. Výsledná hodnota rezistence eferentní arteriole je zdola ohraničena hodnotou minimální rezistence EARL=24 torr min/l, odpovídající maximálně roztažení eferentní arteriole:

\[
EAR=43.333 \times \text{ANMER} \times \text{MDEF} \times \text{VIM} \quad (\text{KD } 12)
\]

když \( \text{EAR}<\text{EARL} \) pak \( \text{EAR}=\text{EARL} \) \( (\text{KD } 13) \)

Angiotenzin ovlivňuje rezistenci aferentní a eferentní arteriole. Eferentní arteriolu ovlivňuje silněji. Multiplikační faktor, vyjadřující vliv angiotenzinu na rezistenci eferentní arteriole (ANMER) je počítán z vlivu angiotenzinu na cévní rezistenci (ANM), vyjádřené poměrem účinku momentální hladiny angiotenzinu k normě a citlivostního koeficientu vlivu angiotenzinu na rezistenci eferentní arteriole (ANMEM=1.5):

\[
\text{ANMER}= (\text{ANM}-1.0) \times \text{ANMEM} + 1.0 \quad (\text{KD } 14)
\]

Multiplikátor, vyjadřující vliv tubulogomerulární vazby na rezistenci eferentní arteriole (MDEF) se počítá ze signálu generovaném maculou densou, působícím jaka na aferentní, tak na eferentní arterioli (RNAUG1). Na eferentní arterioli se však tubulogomerulární vazba projevuje spíše prostřednictvím angiotenzinu, než přímo – proto je odchylka koeficientu MDEF od jedničky menší než odchylka RNAUG1 (kterou macula densa působí na aferentní arterioli). Poměr bezprostředního vlivu macula densy na eferentní a aferentní arterioli určuje citlivostní koeficient EFAFR (měl by být obecně menší než 1, kdyby byla jeho hodnota 1, pak by MDEF=RGNAUG1, tedy jako v aferentní arteriole). Guyton v modelu z roku 1986 uvažuje hodnotu tohoto koeficientu nulovou – pak MDEF je konstanta 1.0 a macula densa pak nemá žádný bezprostřední vliv na eferentní arterioli:

\[
\text{MDEF}= (\text{RNAUG1}-1.0) \times \text{EFAFR} + 1.0 \quad (\text{KD } 15)
\]

⁸http://physiology.ume.edu/themodelingworkshop/Model%20Library/Kidney%20Model/Kidney%20Model _HTML.
EFFECTIVE ARTERIOLE

INPUTS:
- AARK - normal resistance in efferent arteriole (torr min/l)
- RNAUG1 - Macula densa feedback signal to efferent arteriole (ratio to normal)
- EFARF - Sensitivity controller of macula densa feedback signal to efferent arteriole
- ANM - Angiotensin multiplier effect on vascular resistance (ratio to normal)
- VIM - blood viscosity (ratio to normal blood)

OUTPUT:
- AAR - afferent arteriolar resistance (torr l/min)

Efferent Arteriole
Myogenní stimulace aferentní arterioly

Myogenní regulace aferentní arterioly spočívá v tom, že náhle zvýšení perfúzního tlaku, které by jinak vedlo ke zvýšení průtoku, vyvolá vázokonstrikci a následně zvýšení rezistence (snížení vodivosti) arterioly zvýšený průtok omezí. Opačně, při poklesu tlaku (a následném snížení průtoku) se céva rozšíří, poklesne rezistence a zvýší se průtok. Trvání zvýšení či snížení tlaku delší dobu, dbě se na zvýšený, či snížený tlak po čase adaptuje a bude se bránit změnám tohoto nově nastaveného tlaku. Vstupem je perfúzní tlak v ledvinách (PAR), výstupem je multiplikovátor MYOGRSAA, který v rovnici KD 05 ovlivňuje rezistenci aferentní arterioly. Nastavený tlak se "pamatuje" v integračním členu (PADAPT). Rychlost myogenní odpovědi ovlivňuje časová konstanta MYOGTAU (normálně je 240 min). Rozsah vlastní odpovědi je aproximována splinovou funkcí prokázující experimentální data změny vodivosti (obrácená hodnota rezistence) navržená Colemanem. Multiplikační faktor TENSGN určuje zesílení myogenní odpovědi (normálně TENSGN=1) 9.

\[ PDIFF = PAR - PADAPT \]  
\[ DPADAPT = PDIFF / MYOGTAU \]  
\[ PADAPT = \int (DPADAPT) \, dt \]  
\[ MYOGRS1 = \text{function myogenResp}(PDIFF) \]  
\[ MYOGRSAA = \text{TENSGN} / \text{MYOGRS1} \]

9 Struktura je stejná jako i myogenní regulace pro ostatní neledvinové tkáně s tím rozdílem, že místo perfúzního reálného tlaku je uvažován součet arteriálního a kapilárního tlaku. Krom toho je struktura bloku myogenní adaptace upravena tak, aby se zabránilo algebraické smyčce přes tento blok.
Glomerulus

Tlakový spád na aferentní arteriole (APD) je počítán z celkového renálního průtoku zdravých ledvin (RFN). Předpokládá se tedy, že u ledvin, jejichž funkčnost byla snižena (ztrátou ledvinné tkáně) a průtok omezen (když konstanta REK<1) se ve stejném poměru také sniží rezistence aferentní arterioly:

\[ APD = AAR \times RFN \]  

Glomerulární tlak (GLP), tj. tlak v glomerulární kapiláře se rovná renálnímu perfúznímu tlaku (PAR) bez tlakového spádu na aferentní arteriole:

\[ GLP = PAR - APD \]  

Přítok plazmy do dvou ledvin (se 100% fungující ledvinné tkáně) – (RPNIN) vypočítáme z průtoku krve nepoškozenými ledvinními arteriími (RFN), objemu plazmy (VP) a objemu krve (VB), nebo hematokritu (HM):

\[ RPNIN = RFN \times VP / VB = RFN \times (1 - HM) \]  

Odtok plazmy (RPNOUT) bude nižší než přítok plazmy (RPNIN) o od filtrovaný glomerulární filtrát (GFN):

\[ RPNOUT = RPNIN - GFN \]  

Plazmatické proteiny neprocházejí glomerulární membránou (v nenarušených ledvinách) a proto se v odtékající plazmě se zakoncentrují – poměr koncentrace proteinů v odtékající plazmě ku koncentraci proteinů v přítékající plazmě (EFAFPR) se rovná poměru průtokového objemu přítékající plazmy do glomerulu (PRNIN) k průtokovému objemu plazmy odtékající z glomerulu (PRNOUT):

\[ EFAFPR = RPNIN / RPNOUT \]  

Pro prevenci oscilací zde zavádíme "hlídací" okrajeovou podmínku, vyjadřující, že objem odtékající plazmy z glomerulu nikdy nemůže být větší, než objem plazmy z glomerulu odtékající:

když (EFAFPR<1.0) pak EFAFPR=1.0  

Koloidně osmotický tlak v krvi přítékající do glomerulu (PPC) a poměr koncentraci proteinů v odtékající a přítékající krvi do z glomerulu (EFAFPR) jsou vstupními hodnotami, z nichž se počítá průměrný koloidně osmotický tlak v glomerulární kapiláře. Koloidně-osmotický tlak je nelineární (kvadratický) závislý na koncentraci plazmatických proteinů (viz rovnice CP 14). Průměrný koloidně osmotický tlak (GLPC0) pro dané hodnotu PPC a EFAFPR Guyton počítá podle empirického vzťahu:

\[ GLPC0 = 0.98 \times EFAFPR^{1.35} \times PPC \]  

Pro prevenci oscilací (a také i pro odstranění algebraické smyčky: glomerulární filtrace je podkladem k výpočtu onkotického tlaku v glomerulární kapiláře a ten zase ovlivňuje glomerulární filtraci) je zde při počítání průměrného koloidně osmotického tlaku (GLPC) zavedenu túmivý člen s časovou konstantou (GPPD=50):

\[ GLPC = GLPC0 / GPPD \times \frac{dGLPC}{dt} \]  

Tlakový gradient přes kapilářní glomerulární membránu (PFL) se rovná glomerulárnímu tlaku (GLP) minus nasávací průměrný koloidně-osmotický tlak v glomerulární kapiláře (PFL) minus protitlak v proximálním tubulu (PXTP=8), který je v tomto mopdelu uvažován jako konstanta:
Glomerulární filtrace v nepoškozených ledvinách (GFN0) je úměrná filtračnímu tlakovému gradientu přes glomerulární kapilární membránu (PFL) vynásobenému glomerulárním filtračním koeficientem. (GFLC=0.0208 l/min/torr):

\[
GFN0 = PFL \times GFLC
\]  

Pro prevenci oscilací (a také i pro odstranění možné algebraické smyčky) je při výpočtu glomerulární filtrace (GFN) vložen tlumivý člen s časovou konstantou (GFNDMP=3):

\[
DGFN = (GFN0 - GFN) / GFNDMP
\]

\[
GFN = \int DGFN \, dt
\]

Nakonec je ještě vložena okrajová omezující podmínka, která nedovolí poklesnout glomerulární filtraci (GFN) pod hodnotu spodního limitu (GFNL=0.001 l/min)

\[
když \, (GFN < GFNLL) \, pak \, GFN = GFNLL
\]

Dosud jsme počítali gomerulární filtraci v ledvinách, které nebyly poškozeny (GFN). Vynásobíme-li tuto hodnotu glomerulární filtrace koeficientem vyjadřujícím podíl funkční tkáně ledvin k normě (REK) – u zdravých ledvin REK=1, dostaneme skutečnou hodnotu glomerulární filtrace (GFR)

\[
GFR = GFN \times REK
\]
Glomerulus

**INPUTS:**
- PAR - renal perfusion pressure [torr]
- RFN - renal blood flow if kidney is not damaged [l/min]
- AAR - resistance in afferent arteriole [torr min/l]
- HM - hematocrit
- PPC - plasma colloid osmotic pressure [torr]
- PXTP - back pressure in proximal tubule [torr]
- GLFC - glomerular filtration coefficient [l/min/torr]
- REK - percent of normal renal function [ratio to normal]

**OUTPUTS:**
- GFR - glomerular filtration rate [l/min]
- GFN - glomerular filtration rate if kidney is not damaged [l/min]
- GLP - glomerular pressure [torr]
- GLPC - average glomerular plasma colloid osmotic pressure [torr]
- PFL - net pressure gradient in glomerulus [torr]

*Glomerulus diagram*
**Macula densa**

Pro tubuloglomerulární vazbu je důležitá velikost toku sodíku ze začátku distálního tubulu do buněk macula densa. Tato velikost toku závisí na profiltrovaném množství sodíku do glomerulu a také i na resorpci sodíku v proximálním tubulu a v Henleho klíčce.

Zde se ale zjednodušeně předpokládá, že je vtok sodíku do buněk macula densa je závislý pouze na profiltrovaném množství sodíkových iontů (FNA), které se počítá z koncentrace sodíku v plazmě (CNA) a z glomerulární filtrace v nepoškozených ledvinách (GFN):

\[ FNA = GFN \times CNA \]  
(KD 36)

Vstupní signál pro macula densu (NAPT), je pak úměrný rychlosti čerpání sodíkových iontů do buněk macula densa, a je lineárně závislý na filtraci sodíku (FNA) s koeficientem úměrnosti (KNAPT=0.057). Magické číslo 0.057 vychází z toho že signál vtoku sodíku do buněk macula densa je za normálních okolností jednoznačný. Při normální glomerulární filtraci 0.125 l/min a koncentraci sodíku v krvi 140 mmol/l bude filtrace sodíku: FNA=17.5 mmol/min. Aby vyšla jedničková hodnota signálu musí být hodnota konstanty 1/17.5=0.0571:

\[ NAPT = FNA \times 0.057 \]  
(KD 37)

Tato hodnota je zdole ohraničená konstantní hodnotou (NAPTLL=0.1):

\[ když \ (NAPT < NAPTLL), pak \ NAPT = NAPTLL \]  
(KD 38)

Hodnota NAPT je ohraničena i shora (NAPTUL=3):

\[ když \ (NAPT > NAPTUL), pak \ NAPT = NAPTUL \]  
(KD 39)

Pro utlumení oscilací je zavedena zpětná vazba s tlamivou konstantou (GF=0.1). Výsledkem je normalizovaná hodnota dodávky sodíku pro buňky macula densy (NAPT1) - její hodnota se za normálních okolností se rovná jedné:

\[ (NAPT - NAPT1) \times GF^2 \]  
(KD 40)

\[ NAPT1 = \int DNAPT1 \, dt \]  
(KD 41)

Pro výpočet vlastního signálu pro aferentní arteriolu je odchylka signálu NAPT1 od jedničky je zesílena koeficientem RNAUGN=5.0, který představuje normální velikost signálu z macula densy pro aferentní arteriolu. Výsledná hodnota (RNAUG1) je ohraničena zdola (RNAUG1UL=10.0) a zdola (RNAUG1LL=0.3). Výsledkem je multiplikacní koeficient RNAUG1 – jako signál z macula densy pro přílehlou aferentní arteriolu, který ovlivňuje její rezistenci. Signál je určen případně i eferentní arteriolu kde je jeho hodnota modifikována pronásobením koeficientem udávajícím citlivost k tomuto signálu (v modelu Guytona z roku 1986 je ale citlivost na tento signál na eferentní arteriole nastavena na nulovou hodnotu, v Colemanově modelu10, dostupném na Internetu je s vlivem macula densy na eferentní arteriole počítáno, ale model má poněkud jinou strukturu):

\[ RNAUG1 = (NAPT1 - 1) \times RNAUGN \]  
(KD 42)

\[ když \ (RNAUG1 < RNAUG1LL), pak \ RNAUG1 = RNAUG1LL \]  
(KD 43)

\[ když \ (RNAUG1 > RNAUG1UL), pak \ RNAUG1 = RNAUG1UL \]  
(KD 44)

10[http://physiology.umd.edu/themodelingworkshop/Model%20Library/Kidney%20Model/Kidney%20Model.HTML](http://physiology.umd.edu/themodelingworkshop/Model%20Library/Kidney%20Model/Kidney%20Model.HTML)
Macula densa

INPUTS:
- CNA - Extracellular sodium concentration [mmol/l]
- GFN - Glomerular filtration rate if kidney is not damaged [l/min]

OUTPUTS:
- RNAUG1 - Macula densa feedback signal [ratio to normal effect]
- NAPT1 - Delivery of sodium to the macula densa area [ratio to normal value]
**Exkrece sodíku**

Podle učebnicových pramenů se za normálních okolností (při normálním objemu extracelulární tekutiny) se největší část profiltrovaného sodíku vstřebá v proximálním tubulu (viz tabulka). Resorpce v proximálním tubulu je regulována a může klesat či stoupat v závislosti na neurohumorálních stimulech. Sběrné kanálky resorbují normálně ¼ dodaného množství sodíku (teče do nich ze 4% filtrovaného množství sodíku a vstřebá se 3% filtrovaného množství). Při zvýšení resorpce sodíku v proximálním tubulu v důsledku reakce na snížení objemu ECT do sběrných kanálků přitéká menší množství sodíku (místo 4% pouze 2% celkového filtrovaného množství), relativní resorpce sodíku v nich se však zvýší (díky působení aldosteronu se může vstřebat skoro všechno množství sodíku příkřejší do sběrných kanálků). Naopak, při poklesu resorbce v proximálním tubulu v důsledku reakce na zvýšení objemu ECT se zvýší přítok sodíku do sběrných kanálků ale resorbuje se z nich relativně menší množství sodíku, než za normálních okolností (díky snížení hladiny aldosteronu) – místo ¼ třeba jen ¼ dodaného množství.

<table>
<thead>
<tr>
<th>Vstřebávání sodíku v jednotlivých částech nefronu (% profiltrovaného množství)</th>
<th>Norma</th>
<th>Reakce na zvýšení objemu ECT</th>
<th>Reakce na snížení objemu ECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z glomerulárního filtrátu</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Proximální tubulus</td>
<td>-67%</td>
<td>-50%</td>
<td>-80%</td>
</tr>
<tr>
<td>Vtok do Henleho klíčky</td>
<td>33%</td>
<td>50%</td>
<td>20%</td>
</tr>
<tr>
<td>Henleho klíčka</td>
<td>-25%</td>
<td>-30%</td>
<td>-15%</td>
</tr>
<tr>
<td>Vtok do distálního tubulu</td>
<td>8%</td>
<td>20%</td>
<td>5%</td>
</tr>
<tr>
<td>Distální tubulus</td>
<td>-4%</td>
<td>-12%</td>
<td>-3%</td>
</tr>
<tr>
<td>Vtok do sběrných kanálků</td>
<td>4%</td>
<td>8%</td>
<td>2%</td>
</tr>
<tr>
<td>Sběrný kanálek</td>
<td>-3%</td>
<td>-2%</td>
<td>-2%</td>
</tr>
<tr>
<td>Moč</td>
<td>1%</td>
<td>6%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Vyjádříme-li procenta pro filtrované množství sodíku číselně, pak při filtraci 17.5 mmol/min, na vtoku do Henleho klíčky za normálních okolností příště 4.02 mmol/min při hypervolémii 8.75 mmol/min, při hypovolémií 3.5 mmol/min, na vtoku do distálního tubulku při euvolémii teče 1.4 mmol/l, při zvýšení objemu je vzestup na 3.5 mmol/min, při snížení objemu pokles na 0.875 mmol/min; vtok do sběrných kanálků je normálně 0.7 mmol/min, při zvýšení objemu ECT je vzestup až na dvojnásobek – 1.4 mmol/min, při poklesu ECT je naopak pokles na polovičku 0.35 mmol/min, odpad do moči je normálně 0.175 mmol/min, při vzestupu ECT je až šestinásobný vzestup 1.05 mmol/min, při snížení objemu ECT sodík v moči klesá až na téměř nulové hodnoty.

V tomto modelu se nejprve počítá normalizovaná hodnota dodávky sodíku (NADT) do distálního tubulárního systému ledvin (distální tubulus a sběrné kanálky v modelu berou jako jeden celek). Tato hodnota je vyjádřena jako poměr k normální hodnotě. Počítá se na základě hodnoty normalizované dodávky sodíku do buněk macula densy (NAPT1). Vztah je lineární pokud je hodnota NAPT1 větší než norma (tj. větší než 1). Při výpočtu vycházíme z normalizovaného vztahu mezi dodávkou sodíku do distálních tubulů (NADT) a čerpáním sodíku z tohoto toku do macula densy (NAPT1). Při zvýšení toku sodíku do distálních tubulů nad normu lineárně stoupá i čerpání sodíku do buněk macula densy. Při snížení dodávky sodíku do distálních tubulů se čerpání sodíku do macula densy snižuje nelineárně (viz obrázek). V modelu ale nepočítáme dodávku sodíku do macula densy (NAPT1) z dodávky sodíku do distálního nefronu (NADT) ale obráceně, z NAPT1 se počítá NADT. Vztah je určován citlivostním koeficientem (NARSB1=16), vyjadřujícím vliv renální tubulární a cévní dynamiky:

\[
NADT = \frac{(NAPT1-1.0)}{NARSB1} + 1.0 \quad (\text{KD 45})
\]

\[
\text{když } NADT < 1.0, \text{ pak } NADT = NAPT1^{NARSB1} \quad (\text{KD 46})
\]

Vztah mezi NAPT1 a NADT ukazuje následující graf:

Závislost hodnot NAPT1 na NATD počítaná podle rovnic KD 44-45
Skutečná hodnota dodávky sodíkových iontů do distální části nefronu (DTNAI) se pak počítá z normalizované hodnoty (NADT) a příslušné konstanty\(^{11}\):

\[
DTNAI = \frac{NADT}{2.0}
\]

(Verylčování sodíkových iontů do moči v nepoškozených ledvinách (NODN) je počítáno jako rozdíl mezi dodávkou sodíkových iontů do distálního nefronu (NADT) a rychlosti jeho vstřebávání v distálním nefronu (DTNARA):

\[
NODN = DTNAI - DTNARA
\]

Aby nenastala situace, kdy je vylučování sodíku záporné, je v modelu pro jistotu nastavena dolní mez pro vylučování sodíku do moči:

\[
když (NODN < 0.00000001) pak NODN = 0.00000001
\]

Výslednou rychlost vylučování sodíku v ledvinách (NOD) dostaneme z vylučování sodíku v nepoškozených ledvinách (NODN) po vynásobení koeficientem (REK), který vyjadřuje podíl skutečně fungujících ledvin k normě (pro simulaci poškození ledvin ztrátou ledvinné tkáně – normálně je REK = 1):

\[
NOD = REK \times NODN
\]

Z rychlosti vylučování sodíku do moči v nepoškozených ledvinách (NODN) a z diurézy v nepoškozených ledvinách (VUDN) (v případě, že pracuje jen část ledvin – koeficient REK < 1, pak se stejným dílem snižuje hodnota NODN i VUDN, výsledná koncentrace sodíku v moči se nezmění) je počítána koncentrace sodíku v moči (CNU). Protože také koncentrace je v modelu brána jako jeden z faktorů, který ovlivňuje zpětné vstřebávání v distálním nefronu, pak pro prevenci oscilací (a pro eliminaci algebraické smyčky) je do výpočtu koncentrace vělen integrační blok (iniciální hodnota DTNAC = 100 mmol/l, časová konstanta NADMP = 40):

\[
CNU = \frac{NODN}{VUDN}
\]

\[
DDTNAC = \frac{(DTNAC - CNU)}{NADMP}
\]

\[
DTNAC = \int DDTNAC \, dt
\]

Tlumivý integrační člen je využit i při výpočtu vstřebávání sodíku v distální části nefronu (DTNARA), kdy se vypočtená hodnota zpětného vstřebávání (DTNAR1) tlumí integračním členem s časovou konstantou (GFR3 = 0.1). Hodnota integrálu je omezena zdola minimální hodnotou zpětného vstřebávání (DTNARL = 0.999*10^-6):

\[
DDTNARA = \frac{(DTNARA - DTNAR1)}{GFR3}
\]

\[
DTNARA = \int_{\text{dt}}^{\text{DTNARL}} (DDTNARA) \, dt
\]

\(^{11}\) Za normálních hodnot je podle tohoto vztahu dodávka sodíkových iontů 0.5 mmol/min – což je ale méně než udávají jiné modely a některé fyziologické prameny, podle nich se dodávka z Henleho kličky do distálního tubulu pohybuje kolem 1.5-3.5 mmol/min. Dodávka sodíkových iontů do distální části nefronu (DTNAI) spíše představuje dodávku sodíku do sběrných kanálků. Tomu ostatně v modelu odpovídá i vypočtená hodnota rychlosti vstřebávání sodíku v distálním nefronu za normálních okolností (DTNARA = 0.4 mmol/l). Ve sběrných kanálkách se normálně vstřebává asi 2-3% profiltrovaného množství tj. zhruba 3.5-5 mmol/min.
Hodnota vstřebávání sodíku v distální části nefronu (DTNAR1) je počítána jako součin normální hodnoty (DTNARN=0.4) a multiplikačního koeficientu (MDTNAR) který kumulativně vyjadřuje příslušné regulační vlivy (koncentraci sodíkových iontů v tubulech, vliv aldosteronu, zprostředkováný vliv reabsorbce vody účinkem antidiuretického hormonu, vliv renální hemodynamiky na reabsorci sodíku a případný vliv diuretik).

$$DTNAR1=DTNARN*MDTNAR$$ (KD 56)

Pro výpočet multiplikačního koeficientu MDTNAR je uvažována závislost rychlosti zpětného vstřebávání sodíku v distální části nefronu na koncentraci protékajících sodíkových iontů v tubulech – pro multiplikátor je proto v modelu brána výsledná koncentrace sodíku na konci sběrných kanálků – tedy v koncentrace sodíku v moči (DTNAC), dělená její normální hodnotou (NDTAC=100 mmol/l). Dalším uvažovaným vlivem je hladina aldosteronu, resp. jeho vliv na zpětné vstřebávání sodíku, vyjádřený jako poměr účinků aldosteronu vzhledem k normě (AMNA). Dalším vlivem je zprostředkováný vliv antidiuretického hormonu (AHM), vyjádřený jako poměr k normě, který podněcuje vstřebávání vody ze sběracích kanálků do dřeně ledvin, a se vstřebávanou vodou jsou částečně strhávány rozpuštěné ionty sodíku – citlivost je zde určována koeficientem AHMNAR=0.3. Vstřebávání ovlivňuje i prokrvení dřeně – normalizovaný vliv hemodynamického faktoru na resorbci sodíku je zde vyjádřen koeficientem RFAB. Konečně, posledním uvažovaným faktorem je vliv diuretik (charakterizovaný koeficientem DIURET, který je normálně jedničkový, protože diuretika snižují reabsorci je tento koeficient ve jmenovateli):

$$MDTNAR=DTNAC/DTACN*AMNA*((AHM-1)*AHMNAR+1)*RFAB/DIURET$$ (KD 57)
SODIUM EXCRETION

INPUTS:
- NAPT1 - delivery of sodium to the macula densa area [ratio to normal]
- RFAB - the multiplier factor for the effect of renal hemodynamics on reabsorption of sodium and potassium in the distal tubule collecting duct [ratio to normal]
- VUDN - rate of urinary output if kidney is not damaged [l/min]
- AMNA - aldosterone for control of sodium reabsorption [ratio to normal effect]
- AHM - antidiuretic hormone [ratio to normal effect]
- REK - percent of normal renal function [ratio to normal]
- DIURET - effect of diuretic on the distal tubule collecting duct [ratio to normal - without diuretics]

OUTPUTS:
- NADT - the normalized delivery of sodium to the distal tubular system [ratio to normal]
- DTNAI - rate of sodium entry into the distal tubular system [mmol/min]
- NODN - sodium excretion rate if kidney is not damaged [mmol/min]
- NOD - sodium excretion rate [mmol/min]
- CNU - concentration of sodium in urine [mmol/l]

SODIUM EXCRETION

DAMPING

DTNARA = 0.9999e-6
Treshold = 1
if (NADT1 > Treshold) {NADT = NADT1}
else {NADT = NADT2}

lower limit DTNARA=0.9999e-6

lower limit 0.00000001

104
**Exkrece draslíku**

Zdravé ledviny jsou schopny vyloučit přijaté množství draslíku v potravě a snížit ztráty draslíku močí při jeho nedostatku v organismu. V proximálním tubulu a v Henleho klíčce není za fyziologických okolností význačná regulace exkrece draslíku. Naproti tomu distální tubulus a sběrné kanály jsou schopné vstřebávat nebo vylučovat draslík (viz tabulka).

<table>
<thead>
<tr>
<th></th>
<th>Norma</th>
<th>Reakce na omezení průjmu draslíku</th>
<th>Reakce na zvýšení průjmu draslíku</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z glomerulárního filtrátu</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Proximální tubulus</td>
<td>-67%</td>
<td>-67%</td>
<td>-67%</td>
</tr>
<tr>
<td>Vtok do Henleho klíčky</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
<tr>
<td>Henleho klíčka</td>
<td>-20%</td>
<td>-20%</td>
<td>-20%</td>
</tr>
<tr>
<td>Vtok do distálního tubulu</td>
<td>13%</td>
<td>13%</td>
<td>13%</td>
</tr>
<tr>
<td>Distální tubulus</td>
<td>-3%</td>
<td>-3%</td>
<td>-3% až +37%</td>
</tr>
<tr>
<td>Vtok do sběrných kanálků</td>
<td>10%</td>
<td>10%</td>
<td>10-50%</td>
</tr>
<tr>
<td>Sběrný kanálek</td>
<td>+5%</td>
<td>-9%</td>
<td>+5 až +30%</td>
</tr>
<tr>
<td>Moč</td>
<td>15%</td>
<td>1%</td>
<td>15-80 %</td>
</tr>
</tbody>
</table>

V modelu se předpokládá, že draslík se v proximálním tubulu a v Henleho klíčce vstřebává proporceně stejně jako sodík. Proto se výpočet rychlosti toku draslíku do distálního nefronu (DTKI) počítá z rychlosti vtoku draslíku do distálního tubulárního systému (DTNAI) a podílu koncentrací draslíku (CNE) a sodíku (CNA) v extracelulární tekutině.

\[ \text{DTKI} = \text{DTNAI} \times \text{CNE} / \text{CNA} \]  

(KD 58)

V modelu se nejprve počítá s faktory, které mají vliv na sekreci draslíku do distálního nefronu. Je to vliv toku sodíku v tubulech, vliv angiotenzinu, vliv koncentrace draslíku v extracelulární tekutině a vliv aldosteronu.

Vliv toku sodíku v distálních tubulech na sekreci draslíku vyjadřuje multiplikační faktor NADTK, který se počítá z normalizované hodnoty toku sodíku do distálního tubulárního systému (NADT) a citlivostní konstanty (NADTKM=0.5). Výsledek je zdola ohraničen hodnotou 0.1:

\[ \text{NADTK} = (\text{NADT} - 1.0) \times \text{NADTKM} + 1.0 \]  

(když \( \text{NADTK} < 0.1 \) pak \( \text{NADTK} = 0.1 \))  

(KD 59)

Sekreci draslíku ovlivňuje hladina angiotenzinu. Výpočet multiplikačního faktoru vyjadřujícího vliv angiotenzinu na exkreci draslíku (ANMKE) je založen na hladině angiotenzinu (AMN), vyjádřeného jako poměr k normě. Bere se odchylka hodnoty AMN od jedničky, násobí se citlivostní faktorem (ANMKEM=2) a výsledek se přičte k jedničce. Výsledek je zdola ohraničen hodnotou (ANMKEL=0.3):

\[ \text{ANMKE} = (\text{AMN} - 1.0) \times \text{ANMKEM} + 1.0 \]  

(když \( \text{ANMKE} < \text{ANMKEL} \) pak \( \text{ANMKE} = \text{ANMKEL} \))  

(KD 61)

Multiplikátor, vyjadřující vliv extracelulární koncentrace draslíku na sekreci draslíku v distálním nefronu (MCKE) se počítá jako čtvrtá mocnina poměru extracelulární hladiny draslíku (CKE) k normální hodnotě (5 mmol/l) - exponent (CKEEX=4):

\[ \text{MCKE} = (\text{CKE} / 5)^{\text{CKEEX}} \]  

(KD 63)

Rychlost sekrece draslíku do distálního tubulárního systému nefronu (DTKSC) je počítána vynásobením bazální hodnoty (0.06) multiplikátorů vyjadřujícími vliv hladiny draslíku v ECT (MCKE), vliv aldosteronu
(AMK), vliv přísnějším sodíku do distálních tubulů (NADTK) a vliv angiotenzinu (ANMKE) – ten snižuje sekreci, proto je ve jmenovateli:

\[ DTKSC = 0.06 \ast AMK \ast MCKE \ast NADTK / ANMKE \]  

(KD 64)

Resorbce draslíku v distálním nefronu (RFABK) vlivem hemodynamického faktoru závisí na multiplikátoru vyjadřující vliv hemodynamiky (RFAB) a citlivostním koeficientu (RFABKM=0.03).

\[ RFABK = (RFAB - 1.0) \ast RFABKM \]  

(KD 65)

Resorbce draslíku v distální části nefronu (DTKA) lineárně závisí na koncentraci draslíku na konci sběrných kanálků – tedy na koncentraci draslíku v moči (CKU) – do výpočtu je vložen integrační člen pro utlumení oscilací s časovou konstantou (KDMP=3):

\[ DTKA = CKU \ast 0.000293 \]  

(KD 66)

\[ DDTKA = (DTKA1 - DTKA) / KDMP + 0.00000001 \]  

(KD 67)

\[ DTKA = \int DDTKA \, dt \]  

(KD 68)

Vylučování draslíku v obou nepoškozených ledvinách (KODN) závisí na toku draslíku do distální části nefronu (DTKI), sekreci draslíku do distálního nefronu (DTKSC), od součtu těchto toků se odečítá resorbce draslíku v distálním tubulu a případně i dodatečná resorbce vlivem hemodynamického faktoru (normálně jsou nulová) RFABK. Výsledek je zdola ohraničen nulou:

\[ KODN = DTKI + DTKSC - DTKA - RFABK \]  

(když (KODN<0) pak KODN=0.0)  

(KD 69)

Výslednou rychlost vylučování draslíku v ledvinách (KOD) dostaneme z vylučování sodíku v nepoškozených ledvinách (KODN) po vynásobení koeficientem (REK), který vyjadřuje podíl skutečně fungujících ledvin k normě (pro simulaci poškození ledvin ztrátou ledvinné tkáně – normálně je REK=1):

\[ KOD = KODN \ast REK \]  

(KD 70)

Z rychlosti vylučování draslíku do močí v nepoškozených ledvinách (KODN) a z diurézy v nepoškozených ledvinách (VUDN) (v případě, že pracuje jen část ledvin – koeficient REK<1, pak se stejným dílem snižuje hodnota NODN i VUDN, výsledná koncentrace sodíku v moči se nezmění) je počítána koncentrace draslíku v moči (CKU).

\[ CKU = KODN / VUDN \]  

(KD 72)
POTASSIUM EXCRETION

INPUTS:
- CKE - extracellular potassium concentration [mmol/l]
- CNA - extracellular sodium concentration [mmol/l]
- NADT - the normalized delivery of sodium to the distal tubular system [ratio to normal]
- DTNAI - rate of sodium entry into the distal tubular system [mmol/min]
- AMI - angiotensin multilifier effect to vascular resistance [ratio to normal]
- AMK - effect of aldosterone on potassium secretion [ratio to normal]
- RFAB - the multiplier factor for the effect of renal hemodynamics in the distal tubule collecting duct [ratio to normal]
- REK - percent of normal renal function [ratio to normal]
- VUDN - rate of urinary output if kidney is not damaged [l/min]

OUTPUTS:
- KODN - potassium excretion rate if kidney is not damaged [mmol/min]
- KOD - potassium excretion rate [mmol/min]
- CKU - concentration of potassium in urine [mmol/l]

Potassium Excretion
Exkrece močoviny a vody

Ledviny vylučují zhruba 40% filtrovaného množství močoviny (toto procento se mění v závislosti na glomerulární filtraci) – v modelu je při výpočtu rychlosti vylučování močoviny zdravými ledvinnami (DTURI) využita empirická závislost vylučování močoviny na glomerulární filtraci zdravých ledvin (GFN) a koncentrace močoviny (PLURC) v ECT:

\[ DTURI = PLURC \times 3.84 \times GFN^2 \]  

(KD 73)

Skutečnou rychlost vylučování močoviny ledvinnami počítáme z rychlosti vylučování zdravými ledvinnami (DTURI) pronásobením koefficientu (REK), vyjadřujícím podíl funkční tkáně ledvin k normě (normálně je \( REK = 1 \)):

\[ UROD = DTURI \times REK \]  

(KD 74)

Rychlost vylučování mimiosmolských močovin a elektrolytů (OSMPN) nepoškozených ledvin se počítá z vylučování zdravými ledvinnami (DTURI) a z vylučování sodíku (KODN) a draslíku (NODN):

\[ OSMOPN = DTURI + 2.0 \times (NODN + KODN) \]  

(KD 75)

Počítáme, že tok do 0.56 mmosm/min je pod vlivem ADH, co je nad tuto hranici – na tu už ADH nemá vliv. Proto se počítáme rychlost vylučování nadbýtku miliosmolů nad 0.56 mmosm/min (OSMOP1) z celkové rychlosti vylučování miliosmolů (OSMOPN). Pokud hodnota OSMOPN je menší než 0.56 mosm/min, pak je pochopitelně i hodnota toku OSMOP1 nulová:

\[ OSMOP1 = \begin{cases} OSMOPN - 0.56 & \text{když } OSMOP1 < 0.0 \\ 0.0 & \text{inak} \end{cases} \]  

(KD 76)

(když \( OSMOP1 < 0.0 \) pak \( OSMOP1 = 0.0 \))  

(KD 77)

Pro určení osmotického toku, který je ovlivňován hladinou ADH (OSMO0) ořizneme rychlost vylučování sumární hodnoty elektrolytů a močoviny (OSMOPN horní hranice 0,56):

\[ OSMOPN_0 = OSMOPN \]  

(když \( OSMOPN > 0.56 \) pak \( OSMOPN_0 = 0.56 \))  

(KD 78)

(když \( OSMOPN > 0.56 \) pak \( OSMOPN_0 = 0.56 \))  

(KD 79)

Nyní máme vylučování osmoticky aktivních látek rozděleno na dva toky: tok ovlivňitelný ADH (OSMOPN0) a dodatečný tok neovlivnitelný ADH. Pro výpočet diurézy se v modelu předpokládá, že při normální hladině ADH je v moči koncentrace osmoticky aktivních látek 560 mmol/l. Toto množství se může měnit v závislosti na hladině ADH. Pak můžeme spočítat, jaká diuréza (VUDN0) odpovídá této koncentraci:

\[ VUDN_0 = OSMOPN_0 / 560.0 / AHM \]  

(KD 80)

(když \( OSMOPN_0 > 0.56 \) pak \( OSMOPN_0 = 0.56 \))  

Obdobně si spočítáme objem vylučovanej moči, odpovídající vylučení druhé části osmotických toků (OSMOP1), za předpokladu, že cílová koncentrace osmoticky aktivních látek v moči je 360 mmol/l:

\[ VUDN_1 = OSMOP1 / 360.0 \]  

(KD 81)

Celková diuréza v nepoškozených ledvínách (VUDN) pak bude součtem těchto toků (VUDN0) a (VUDN1):

\[ VUDN = VUDN_0 + VUDN_1 \]  

(KD 82)

Skutečnou diurézu (VUD) počítáme z diurézy zdravých ledvín (VUDN) pronásobením koefficientu (REK), vyjadřujícím podíl funkční tkáně ledvin k normě (normálně je \( REK = 1 \)):

\[ VUD = VUDN \times REK \]  

(KD 83)
UREA AND WATER EXCRETION

INPUTS:
- PLURC - concentration of urea in body fluids [mmol/l]
- GFN - glomerular filtration rate if kidney is not damaged [l/min]
- NODN - the normalized delivery of sodium to the distal tubular system [ratio to normal]
- KODN - sodium excretion rate if kidney is not damaged [mmol/min]
- DTNAI - rate of sodium entry into the distal tubular system [mmol/min]
- AHM - antidiuretic hormone [ratio to normal]
- REK - percent of normal renal function [ratio to normal]

OUTPUTS:
- UROD - urea excretion rate [mmol/min]
- VUDN - rate of urinary output if kidney is not damaged [l/min]
- VUDN - rate of urinary output [l/min]

WATER AND UREA EXCRETION
Peritubulární kapiláry

Do peritubulárních kapilár se odvádí rezorbované látky z tubulů. Protože do kapilár se teče krev, kde se, díky filtraci v glomerulech, zvýšil koloidně-osmotický tlak, zvýšilo se tím i nasávání rezorbované vody a solutů z intersticia ledvin. Hemodynamické poměry tak mohou ovlivňovat resorpční procesy v ledvinových tubulech. Tento modul počítá faktor RFAB, kterým hemodynamika ovlivňuje vstřebávání sodíku a draslíku. Základem je opět Starlingova rovnováha (tentokrát na peritubulární) kapiláře.

Nejprva vypočítáváme renální kapilární tlak v peritubulárních kapilárách (RCPRS) z rozdílu mezi normálním průtokem ledvin (1,2 l/min) a skutečným průtokem, rozdíl se pronásobí konstantou (RFABX=0.8) a opět přičteme k hodnotě 1.2 – obdržený průtok kapilárami se vynásobí odporem venózního renálního řečiště (RVRS=19.1669):

\[
RCPRS=((RFN-1.2)*RFABX+1.2)*RVRS
\]  
(KD 84)

Výpočet koloidně-osmotického tlaku v ledvinové tkáni (RTSPPC) se počítá z průměrného koloidného osmotického tlaku v glomerulárních kapilárách (GLPC) pronásobeným faktorem (RTPPR=0.8999), který snižuje tento tlak díky resorpci tekutiny do peritubulárních kapilár minus faktor reprezentující rozdíl v koncentraci proteinů v plazmě a peritubulárních tkání (RTPPRS=15.19999). Výpočet je ohraničen zdola:

\[
RTSPPC=GLPC*RTPPR-RTPPRS
\]  
(když (RTSPPC<1.0) pak RTSPPC=1.0)  
(KD 85)

Výpočet gradientu způsobujícího resorpci tekutiny do peritubulárních kspilár (RABSPR) se počítá z nasávacího onkotického tlaku v peritubulárních kapilárách, který se bere stejný jako průměrný koloidně osmotický tlak v glomerulárních kapilárách (GLPC) plus tkáňový protitlak ledvinové tkáni (RTSPRS=6) minus kapilární tlak v peritubulárních kapilárách (RTSPRS), minus nasávací koloidně osmotický tlak ledvinné tkáni (RTSPSC):

\[
RABSPR=GLPC+RTSPRS-RCPRS-RTSPPC
\]  
(KD 87)

Nyní se počítá reabsorbční faktor (RFAB1), charakterizující reabsorci do peritubulárních kapilár na základě tlakového gradientu mezi tkání ledvina a peritubulární kapilárou a peritubulárního kapilárního absorbního koeficientu (RABSC=0.5):

\[
RFAB1=RABSPR*RABSC
\]  
(KD 88)

Pro prevenci oscilací zde zařazujeme integrační tlučivý člen s časovou konstantou RFABD=1:

\[
DRFAB2=(RFAB1-RFAB2)/RFABD
\]  
(KD 89)

\[
RFAB2=\int DRFAB2 \, dt
\]  
(KD 90)

Nakonec vypočítáme normalizovaný multiplikátor RFAB, vyjadřující vliv renální hemodynamiky na reabsorci sodíku a draslíku v tubulech. Nejprve odečteme výslednou hodnotu RFAB2 od normy (kterou je 1.0) a vynásobíme váhovým koeficientem (RFABM=0.3), charakterizující váhu hladiny plazmatických proteinů proteinů a změn tlaku v peritubulárních kapilárách na změny resorbce sodíku a draslíku. Druhou částí normalizovaného multiplikátoru je vyjádření vlivu změn krevního průtoku ledvin na změny resorbce sodíku a draslíku (váhový koeficient RFNM=0 je zde nastaven na nulu, ale tuto vstupní hodnotu je možno měnit). Výsledný faktor RFAB nakonec ohraničíme zdola hodnotou 0.0001.

\[
RFAB=(RFAB2-1.0)*RFABM+(RFN-1.2)*RFNM+1.0
\]  
(když (RFAB<0.0001) pak RFAB=0.0001)  
(KD 90)

110
INPUTS:
- RFN - renal blood flow if kidney is not damaged [l/min]
- GLPC - average glomerular plasma colloid osmotic pressure [torr]

OUTPUTS:
- RFAB - the multiplier factor for the effect of renal hemodynamics on reabsorption of sodium and potassium in the distal tubule collecting duct [ratio to normal]
- RCPRS - renal capillary pressure around the tubular system [torr]

Peritubular Capillaries

lower limit = 1.0  RFAB2
ŽÍZEŇ, PITÍ ČI CHUŤ NA SLANÉ

Chuti na slané je vyjádřen koeeficientem STH – na něm pak závisí změna rychlosti přijmu sodíku (viz modul elektrolytů, rovnice EL 02). Normální hodnota tohoto koeeficientu je 1. Vzestup chuti na slané může nastat při cirkulačním šoku, když se sníží adekvátní dodávka kyslíku ke tkáním. V modelu se s tím počítáno přes obvlnění koeeficientu STH prostřednictvím tenze kyslíku v buňkách nesvalových tkání (POT).

Počítáme vliv poklesu PO2 v buňkách nesvalových tkání (POT) na koeeficient chuti na slané (STH), aktivace nastává při poklesu POT od hodnoty (Z10=8.25), přes koeeficient zesílení (Z11=4). Výsledná hodnota je ohraničena zdola (1) a zhora (8):

\[
\begin{align*}
STH &= (Z10 - POT) \times Z11 \quad \text{(TS 01)} \\
&\text{když (STH < 1.0) pak STH = 1.0} \quad \text{(TS 02)} \\
&\text{když (STH > 8.0) pak STH = 8.0} \quad \text{(TS 03)}
\end{align*}
\]

Dále počítáme ovlivnění žízně, která je závislá na koeeficientu "chuti na slané" (STH) a vlivem antidiuretického hormonu (AHC1) – výsledkem je kumulativní koeeficient AHTH, vyjadřující rychlost pití, výsledek je ohraničen zdola:

\[
\begin{align*}
AHTH &= ((AHC1 - 1.0) \times AHTHM + 1.0) \times STH \times 0.001 \quad \text{(TS 04)} \\
&\text{když (AHTH < 0) pak AHTH = 0.0} \quad \text{(TS 05)}
\end{align*}
\]

Dalším faktorem, který ovlivňuje (a potenciálně) pocit žízně je hladina angiotenzinu (AMN). Z hodnoty vyjádřené jako poměr hladiny angiotenzinu k normě, vypočítáme další příspěvek ANMTH (vyjádřený jako rychlost pití):

\[
\begin{align*}
ANMTH &= (ANM - 1.0) \times ANMTM \times 0.001 \quad \text{(TS 06)}
\end{align*}
\]

Nakonec spočítáme součet obou příspěvků a dostaneme hodnotu požadované rychlosti příjmu vody (TVZ), výsledek je ohraničen zdola nulou:

\[
\begin{align*}
TVZ &= ANMTH + AHTH \quad \text{(TS 07)} \\
&\text{když (TVZ < 0) pak TVZ = 0.0} \quad \text{(TS 08)}
\end{align*}
\]

Nakonec mezi požadovanou rychlost příjmu vody (TVZ) a skutečným příjmem vodík vložíme integrační tlumivý člen s časovou konstantou TVSDL=30. Krom vypočtené hodnoty příjmu tekutiny dané žízně je dalším vstupem přímo vstupní hodnota rychlosti příjmu tekutiny DR – pomocí této hodnoty je možno např. simulovat infúze apod.

\[
\begin{align*}
DTVD &= \left(\frac{TVZ + DR - TVD}{TVDDL}\right) \quad \text{(TS 09)}
\end{align*}
\]

\[
\begin{align*}
TVD &= \int DTVD \, dt \quad \text{(TS 10)}
\end{align*}
\]

Nakonec jsou v modulu uvažovány dvě řídící proměnné, jsou-li kladné výpočet probíhá podle výše uvedených rovnic, jsou-li nulové či záporné, pak hodnota STH (na níž závisí rychlost příjmu sodíku) je jedničková a rychlost příjmu sodíku bude určována hodnotou NID (viz rovnici EL 02). Pokud bude nastavena záporná či nulová hodnota vstupní řídící proměnné TVDENABLED, pak rychlost příjmu tekutin bude určována hodnotou proměnné DR:

\[
\begin{align*}
&\text{když (STHENABLED <= 0) pak STH = 1} \quad \text{(TS 11)} \\
&\text{když (TVDENABLED <= 0) pak TVD = DR} \quad \text{(TS 12)}
\end{align*}
\]
THIRST, DRINKING AND SALT APPETITE

INPUTS:
- POT: non-muscle cell PO2 [torr]
- AHC1: antidiuretic hormone concentration factor in the circulating body fluids [ratio to normal]
- ANM: angiotensin multiplier effect to vascular resistance [ratio to normal]
- AMK: effect of aldosterone on potassium secretion [ratio to normal]
- DR: forced input of fluid over and above the natural drinking desire (it may be used for intravenous infusion as well) [l/min]

STHENABLED: switching variable:
- If STHENABLED=0, then STH is not calculated and STH=1

TVDENABLED: switching variable:
- If TVDENABLED=0, then TVD is not calculated and TVD=DR

OUTPUTS:
- STH: salt appetite multiplier factor [ratio to normal]
- TVD: actual rate of fluid intake [l/min]

Thirst, Drinking and Salt Appetite
REGULAČNÍ FUNKCE ANGIOTENZINU

Podnětem pro sekreci angiotenzinu je čerpání sodíku z oblasti macula densa – vyjádřené normalizovanou vstupní hodnotou NAPT1 (viz modul Macula Densa). První tři rovnice vypočítávají závislost bazální sekrece angiotenzinu na hodnotě NAPT1. Konstanty (ANRUL=200) a (SLOPE=0.4) ovlivňují sklon a maximum výsledné křivky závislosti rychlosti bazální sekrece (ANR) na čerpání sodíku z tubulu v oblasti macula densa (viz graf)

\[ NAPTR = \frac{1.0}{NAPT1} \]  (AN 01)
\[ \text{když } (NAPTR>100.0) \text{ pak } NAPTR = 100.0 \]  (AN 02)
\[ ANR = ANRUL - (ANRUL-1)/(1+SLOPE)(NAPTR-1.0) \]  (AN 03)

Další rovnice počítají efekt hypertrofie juxtaglomerulárního aparátu jako odpověď na chronické dráždění buněk macula densy. Stupeň hypertrofie je řízen řídící proměnnou ANXM (při ANXM=0 není žádná hypertrofie). Výstupem je proměnná ANX1, která charakterizuje dodatečnou rychlost sekrece v důsledku hypertrofie:

\[ ANX = (ANR-1)*ANXM \]  (AN 03)
\[ DANX1 = (ANX-ANX1)/ANV \]  (AN 04)
\[ ANX1 = \int \text{DANX1 }dt \]  (AN 05)
Další rovnice počítají celkovou sekreci angiotenzinu z normálních a hypertrofovaných buněk juxtaglomerulárního aparátu (ANP). Sumární rychlost sekrece (ANP) je modifikována multiplikátorom REK který reprezentuje podíl funkční tkáně ledvin k normě (normálně REK=1) a dovoluje simulovat patologické stavy při snížení ledvinného parenchymu.

\[
ANP = (ANR + ANX1) \times REK
\]

když \( ANP < 0.00001 \) pak \( ANP = .00001 \)  

\[(AN 06)\]

Dále se na základě rychlosti sekrece počítá postupné hromadění angiotenzinu v organismu a příslušné změny jeho hladiny. Počítá se hladina angiotenzinu (ANCN) – jako poměr k normě s časovou konstantou \( ANT=50 \):

\[
DANCN = (ANP - ANCN) / ANT
\]

\[(AN 08)\]

Koncentrace angiotenzinu (ANC) je počítána z vlastní produkce (ANCN) a z infúze angiotenzinu (ANG):

\[
ANC = ANCN + ANG
\]

\[(AN 10)\]

Při výpočtu multiplikačních faktorů odpovídajících vlivu angiotenzinu na různé funkce (např. na vaskulární rezistenci) musíme počítat s tím, že účinek není přímě úměrný koncentraci. Proto při výpočtu účinku angiotenzinu na cívní rezistenci (AMN) tento multiplikátor počítáme podle nelineárního vztahu z hladiny angiotenzinu (ANC). ANCE=0.699, ANNM=0.15 a ANMM1=0.85 jsou v této rovnici empiricky zjištěné parametry:

\[
ANM = ANC \times ANCE \times ANMM + ANMM1
\]

\[(AN 11)\]

Multiplikátor odpovídající účinku angiotenzinu na systémovou arteriální rezistenci (ANU) se počítá z obecného multiplikátoru vyjadřujícího vliv angiotenzinu na cívní rezistenci (ANM), ANUM=3 je empiricky zjištěný koeficient:

\[
ANU = (ANM-1) \times ANUM + 1
\]

když \( ANU < 0.8 \) pak \( ANU = 0.8 \)  

\[(AN 12)\]

Další rovnice vypočítávají multiplikátor vyjadřující vliv na arteriolární systémovou rezistenci (ANUVN) - zesilovací koeficient ANUVM je vstup (při ANUVM=0 není uvažován žádný vliv).

\[
ANUVN = (ANU - 1) \times ANUVM + 1.0
\]

\[(AN 14)\]

Výpočet koeficientu vyjadřující vliv angiotenzinu na senzitivitu baroreceptorů (ANUBR). Zesilovací koeficient ANUBRM je vstup modelu:

\[
ANUBR = (ANU - 1) \times ANUBRM
\]

\[(AN 15)\]
THE CONTROL FUNCTIONS OF ANGIOTENSIN

INPUTS:
NAPT1 - delivery of sodium to the macula densa area [ratio to normal value]
REK - percent of normal renal function [ratio to normal]
ANXM - controls of degree of hypertrophy of the juxtaglomerulal apparatus [0 = no hypertrophy]
ANG - excess of angiotensin concentration caused by infusion [ratio to normal level of angiotensin]
ANUBRM - sensitivity controller for the effect of angiotensin of the baroreceptor system
ANUVM - sensitivity controller for the multiplier factor of the systemic veins

OUTPUTS:
ANM - angiotensin multiplier factor on vascular resistance [ratio to normal]
ANU - angiotensin multiplier factor on peripheral arteriolar resistance [ratio to normal]
ANUBR - angiotensin multiplier factor for the effect in controlling the sensitivity of the baroreceptor system [ratio to normal]
ANUVN - angiotensin multiplier factor for the constriction of systemic veins [ratio to normal]
ANC - angiotensin concentration in blood [ratio to normal]
REGULAČNÍ FUNKCE ALDOSTERONU

Řídícími signály pro sekreci aldosteronu je angiotenzin a hladina extracelulárního draslíku. Jako vstup se zde bere obecný multiplikační člen angiotenzinu (ANM) a koncentrace draslíku v extracelulární tekućině (CKE). Nejprve se spočte vliv angiotenzinu – senzitivita je řízena hodnotou v řídící proměnné (ANMALM=3), výsledná hodnota má ohraničení zespoda (-0.2). K výsledku je přičtena hladina draslíku, součet je podělen normální hodnotou hladiny draslíku (5 mmol/l). Rozdíl od jedničky je pak pronásoben koeficientem senzitivity (AMNUL=12) a výsledek je podkladem pro výpočet rychlosti tvorby aldosteronu (AMR), vyjádřené jako poměr k normě:

\[
ANMAL = (ANM-1.0) \times ANMALM \\
\text{když } (ANMAL < -0.2) \text{ pak } ANMAL = -0.2
\]

\[
AMR = (((CKE+ANMAL)/5.0-1.0) \times AMKUL + 1.0) \\
\text{když } (AMR < 0.0) \text{ pak } AMR = 0.0
\]

Z rychlosti sekrece aldosteronu (AMR) a případně i z rychlosti podávané infúze aldosteronu (ALD) se pak počítá koncentrace aldosteronu AMC, časová konstanta: AMT=60. Výsledkem je koncentrace aldosteronu (AMC) vyjádřená jako poměr k normální hodnotě:

\[
AMC = \frac{AMR + ALD - AMC}{AMT}
\]

Protože účinek aldosteronu není lineárně závislý na jeho hladině, nejprve se z hladiny aldosterony (AMC), vyjádřené relativně jako poměr k normě, počívá obecný multiplikátor vyjadřující účinek aldosteronu (AM). Koeficienty empirické funkce jsou AMREX=0.3 a ALDMM=6.0:

\[
AM1 = AMC \times AMREX
\]

\[
AM = (AM1-1.0) \times ALDMM + 1.0
\]

Z obecného multiplikátoru vyjadřujícího účinek aldosteronu je počítán multiplikátor, vyjadřující účinek aldosteronu na trasport draslíku přes buněčnou membránu (AMK). Citlivostní koeficient je zde AMKM=0.5:

\[
AMK = (AM-1.0) \times AMKM + 1.0 \\
\text{když } (AMK < 0.2) \text{ pak } AMK = 0.2
\]

Druhým multiplikátorem je multiplikátor vyjadřující vliv aldosteronu na trasport sodíku přes buněčnou membránu (AMNA). Citlivostní koeficient AMNAM=1.2 ukazuje, že na trasport sodíku má aldosteron obecně větší vliv než na trasport draslíku:

\[
AMNA = (AM-1.0) \times AMNAM + 1.0 \\
\text{když } (AMNA < 0.4) \text{ pak } AMNA = 0.4 \\
\text{když } (AMNA > 12) \text{ pak } AMNA = 12
\]
THE CONTROL FUNCTIONS OF ALDOSTERONE

INPUTS:
ANM - angiotensin multiplier effect on vascular resistance [ratio to normal value]
CKE - extracellular fluid potassium concentration [mmol/l]
ALD - rate of infusion of aldosterone [relative to normal rate of aldosterone secretion]

OUTPUTS:
AMK - effect of aldosterone on potassium secretion [ratio to normal]
AMNA - aldosterone for control of sodium reabsorption
AMC - aldosterone concentration [ratio to normal]

The control functions of aldosterone.
REGULAČNÍ FUNKCE ANTIDIURETICKÉHO HORMONU

Nejprve jsou kumulovány jednotlivé dílčí vlivy působící na sekreci antidiuretického hormonu. Dílčí vliv autonomní stimulace (AUP) na sekreci antidiuretického hormonu (AH8) se počítá snadno – od vstupní realitní hodnoty se odečte jedněčka:

\[ AH8 = AUP - 1.0 \]  
(AD 01)

Dílčí vliv extracelulární koncentrace sodíku (CNA) na výdej antidiuretického hormonu (CNAH) se počítá tak, že se od hodnoty CNA odečte konstanta (CNR=139) a rozdíl se vynásobí citlivostním koeficientem (CNZ=1):

\[ CNAH = CNZ * CNB \]  
(AD 02)

\[ CNB = CNA - CNR \]  
(AD 03)

Dílčí vliv angiotenzinu (ANM) spolu s vlivem arteriálního tlaku (PA) na výdej ADH (AH12) závisí na citlivostních koeficientech (ANAPDM=12 a ANADHS=0.15):

\[ AH12 = (ANM - 1.0) * ANADPM + 100.0 - PA \]  
(AD 04)

\[ AH12 = (ANM - 1.0) * ANADPM + 100.0 - PA \]  
(AD 05)

\[ AH12 = (ANM - 1.0) * ANADPM + 100.0 - PA \]  
(AD 06)

\[ AH12 = (ANM - 1.0) * ANADPM + 100.0 - PA \]  
(AD 07)

Všechny tyto dílčí vlivy, spolu s vlivem příspěvkům stimulace atrialních receptorů na sekreci ADH při volumorecepti (vyjádřený proměnnou AH7), se sečtou a pronásobí se koeficientem aby se výsledné číslo za normálních vstupních hodnot (CNA=142, PA=100, AUP=1, ANM=1, AH7=0) normalizovalo na 1.

\[ AH = CNZ * CNB + AH8 - AH7 + AH12 \]  
(AD 05)

\[ AH = CNZ * CNB + AH8 - AH7 + AH12 \]  
(AD 06)

\[ AH = CNZ * CNB + AH8 - AH7 + AH12 \]  
(AD 07)

\[ AH = CNZ * CNB + AH8 - AH7 + AH12 \]  
(AD 08)

\[ AH = CNZ * CNB + AH8 - AH7 + AH12 \]  
(AD 09)

Aktuální koncentrace ADH (AHC) se počítá z AHC1 (exponent AHMM se ale rovná 1) a sečítá se zde ještě se vstupní infúzi ADH:

\[ AHC = AHC1 ^ {AHMM} + ADH \]  
(AD 10)

\[ AHC = AHC1 ^ {AHMM} + ADH \]  
(AD 11)

\[ AHC = AHC1 ^ {AHMM} + ADH \]  
(AD 12)

\[ AHC = AHC1 ^ {AHMM} + ADH \]  
(AD 13)

Výsledek je ohraničen zdola a shora:

\[ když (AHC > 1.0) \]  
\[ pak AHM = (3.15 - 4.0 * AHC) / (0.15 - AHC) \]  
\[ jelikož AHM = 0.15 + 0.85 * AHC \]  
(AD 11)

\[ když (AHC < 0.12) pak AHM = 0.12 \]  
\[ když (AHC > 2.5) pak AHM = 2.5 \]  
(AD 12)

\[ když (AHC < 1.5) pak AHM = 2.5 \]  
(AD 13)

\[ když (AHC > 2.5) pak AHM = 2.5 \]  
(AD 14)

Nakonec se z (AHM) počítá multiplikátor vlivu ADH na arteriální resistenci (AHMR), AHMRM je citlivostní koeficient (Guyton ale uvádí hodnotu tohoto koeficientu nulovou!)
THE CONTROL FUNCTIONS OF ANTIDIURETIC HORMONE

INPUTS:
- CNA - Concentration of sodium in extracellular fluid [mmol/l]
- AUP - autonomic multiplier effect on ADH hormone excretion etc.
  [ratio to normal]
- ANM - angiotensin multiplier effect [ratio to normal]
- PA - systemic arterial pressure [torr]
- H7 - effect of right atrial volume receptor reflex on ADH secretion
  [relative additive factor, normal value = 0]
- AHMRM - sensitivity coefficient for the effect of ADH
  on systemic arterial resistance.

OUTPUTS:
- AHC1 - antidiuretic hormone concentration in the circulating body fluids
  [ratio to normal]
- AHM - antidiuretic hormone multiplier [ratio to normal effect]
- AHMR - effect of antidiuretic on systemic arterial resistance
  [ratio to normal effect]
- ANUBRM - sensitivity controller for the effect of angiotensin
  of the baroreceptor system
- ANUVM - sensitivity controller for the multiplier factor
  of the systemic veins

The control Functions of Antidiuretic Hormone
KIDNEY DYNAMICS AND EXCRETION

INPUTS:
- PA - aortic pressure (torr)
- PPC - plasma colloid osmotic pressure (torr)
- VIM - blood viscosity (ratio to normal blood [torr min/ml])
- REK - percent of normal renal function (ratio to normal)
- CNE - sodium concentration abnormality causing third factor effect (mmol/l)
- AUM - sympathetic vasoconstrictor effect on arteries (ratio of normal effect)
- AHM - antidiuretic hormone (ratio of normal effect)
- AM - aldosterone multiplier (ratio of normal effect)

OUTPUTS:
- VUD - rate of urinary output (l/min)
- RBF - renal blood flow (l/min)
- RFN - renal blood flow if kidney is not damaged (l/min)
- NOD - rate of renal excretion of sodium (mmol/min)
- AAR - afferent arteriolar resistance (torr min/l)
- RR - renal resistance (torr min/l)
- GLP - glomerular pressure (torr)
- GFN - glomerular filtration rate of undamaged kidney (l/min)
- GFR - glomerular filtration rate (l/min)

Kidney - Guyton, Coleman & Granger 1972
Evaluation of cardiorespiratory functions during heart catheterisation through simulation model identification

Kofránek, Jiří; Munclinger Miroslav; Šerf, Boris; Fusek Martin; Kautzner, Josef; Ducháč, Václav; Pokorný, Zdeněk; Brelidze, Zurab and Gondžilašvili, Jason

str. 431-436
EVALUATION OF CARDIORESPIRATORY FUNCTIONS DURING HEART CATHETERISATION

THROUGH SIMULATION MODEL IDENTIFICATION

J. Kofráněk, M. Munclinger, B. Šerf, M. Pusek, J. Kautzner, V. Ducháč, Z. Pokorný, Z. Brelidze and J. Gondzhalasvili

Department of Pathological Physiology and 2nd Department of Medicine, Charles University, Prague, Czechoslovakia; and Central Science Laboratory, Ministry of Health, Tbilisi, USSR

BASIC RELATIONSHIP

The development of personal computers has made it possible for complex computations to be performed during clinical examination in catheterisation laboratories. For this purpose we have developed a mathematical model of blood gas transport, which takes as its input the data gained in the heart catheterisation presented in Table 1. The mathematical realisation of the physiological laws in the simulation model is presented in Appendix 1, with

Table 1. Input Data

<table>
<thead>
<tr>
<th>BASIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>- body weight,</td>
</tr>
<tr>
<td>- body height,</td>
</tr>
<tr>
<td>- rectal temperature.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VENTILATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>- barometric pressure,</td>
</tr>
<tr>
<td>- room temperature,</td>
</tr>
<tr>
<td>- expired volume (ATPS),</td>
</tr>
<tr>
<td>- time for measurement of expired volume,</td>
</tr>
<tr>
<td>- fractional concentration of oxygen in inspired and expired gas,</td>
</tr>
<tr>
<td>- fractional concentration of carbon dioxide in inspired and expired gas (optional).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>- haemoglobin concentration,</td>
</tr>
<tr>
<td>- pH, PCO₂, PO₂ in arterial and mixed venous blood (at 37°C),</td>
</tr>
<tr>
<td>- oxygen saturation in arterial and mixed venous blood (optional).</td>
</tr>
</tbody>
</table>
Table 2. Output Data

<table>
<thead>
<tr>
<th>ARTERIAL AND MIXED VENOUS BLOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>- pH, PO₂, PCO₂ corrected for rectal temperature,</td>
</tr>
<tr>
<td>- base excess,</td>
</tr>
<tr>
<td>- actual and standard bicarbonate concentrations,</td>
</tr>
<tr>
<td>- oxygen saturation,</td>
</tr>
<tr>
<td>- blood O₂ and CO₂ contents.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>END PULMONARY CAPILLARY BLOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>- O₂ and CO₂ contents.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALVEOLAR GAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>- PO₂ and PCO₂.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESPIRATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>- minute ventilation,</td>
</tr>
<tr>
<td>- O₂ consumption (BTPS),</td>
</tr>
<tr>
<td>- CO₂ production (BTPS),</td>
</tr>
<tr>
<td>- alveolar ventilation (BTPS),</td>
</tr>
<tr>
<td>- respiratory quotient.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CIRCULATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>- systemic blood flow (cardiac output and index)</td>
</tr>
<tr>
<td>- pulmonary blood flow,</td>
</tr>
<tr>
<td>- right-to-left shunt blood flow.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VENTILATION-TO-PERFUSION RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>(alveolar ventilation/cardiac output)</td>
</tr>
</tbody>
</table>

The symbols used listed in Appendix II. Incorrect data are rejected and, if the measured data are consistent with physiological relationships, the clinically relevant data presented in Table 2 are calculated.

In the first step of model identification, the total blood oxygen and carbon dioxide contents in arterial and mixed venous blood are calculated from pH, PCO₂ and PO₂ (measured at a temperature of 37°C) using (4), (5) and (6) (see Appendix I). Then the values of pH, PCO₂ and PO₂ are corrected for the patient's temperature using (8) - (10), and arterial and mixed venous blood acid-base parameters (base excess, base excess in virtually fully oxygenated blood - BEox, actual and standard bicarbonate concentrations) are calculated using (2), (3), (7), (11) and (12).

Furthermore, the respiratory quotient (RQ) is determined by (14) from the O₂ and CO₂ contents in arterial and mixed venous blood. If the RQ value obtained is beyond the physiologically feasible range, the inconsistent nature of the measured input data is indicated.

In the second step, the minute ventilation is calculated using (17) and (18). For the correct evaluation of the oxygen consumption we have to know the volume fractions of oxygen and carbon dioxide in dry expired gas (FeCO₂) - see (19). If the value of FeCO₂ is not measured, the oxygen consumption can be evaluated indirectly from alveolar ventilation, using (22) - (24).
If $\text{FeCO}_2$ is measured, the comparison between the direct and indirect calculations of oxygen consumption can be used for testing the consistency of the measured data. Moreover, it is possible to calculate the carbon dioxide consumption directly from (20), and the respiratory quotient, which can be compared with the indirectly obtained value, from (14), for checking the consistency of the measured data.

For indirect estimation of alveolar ventilation it is necessary to have knowledge of the value of alveolar oxygen tension. This can be obtained from the Bohr equation (24). However, for this calculation we need the value of alveolar $\text{CO}_2$ tension, which can be obtained using an iterative procedure. At the start of this iteration, the equivalence of the arterial and alveolar carbon dioxide blood tensions is assumed. Then, oxygen alveolar tension is determined from carbon dioxide tension using the Bohr equation (24). If the value of alveolar $\text{PO}_2$ obtained is lower than arterial $\text{PO}_2$, the inconsistency of the measured data is indicated. Now, supposing that the blood gaseous tensions in alveoli and in the pulmonary end-capillary blood are in equilibrium, the oxygen and carbon dioxide contents in the pulmonary end-capillary blood are calculated by a computer procedure (using (1), (3) - (6) and (8) - (11)) from the values of $\text{PO}_2$, $\text{PCO}_2$, $\text{BE} \text{ox}$ (this value of base excess in virtually fully oxygenated blood is the same in arterial and pulmonary end-capillary blood), haemoglobin concentration and blood temperature.

Subsequently, using (16) the new value of carbon dioxide content in the end-pulmonary capillary blood is determined. Afterwards, the new value of $\text{PCO}_2$ in pulmonary-end capillary blood is calculated by another computer procedure from the values of $\text{BE} \text{ox}$, blood gas contents, haemoglobin concentration and blood temperature, using (1), (3) - (6) and (8) - (11). Assuming equivalence of carbon dioxide tensions in alveoli and pulmonary end-capillary blood, we obtain the new value of alveolar $\text{CO}_2$ tension for the next iteration.

The values of alveolar oxygen and carbon dioxide tensions (which are essential for the calculation of alveolar ventilation and, subsequently, exact determination of oxygen consumption) are obtained as the result of this iterative procedure. A further result of this iterative sub-routine is the value of oxygen content in pulmonary end-capillary blood, which is necessary for determination of right-to-left shunt perfusion (by (15)).

In contrast to the classical procedure for calculating the oxygen content in the pulmonary end-capillary blood from arterial and mixed venous blood gases (Kelman, 1966b; Ruiz et al., 1975; Newell et al., 1980), no assumptions are needed regarding the identity of $\text{PCO}_2$ and pH in arterial and pulmonary end-capillary blood. Finally, cardiac output, pulmonary perfusion and the ventilation to perfusion quotient can be determined.

Through model identification, not only were relevant clinical data obtained, but also the consistency of the input measured data was checked. The program informs us whether the instrumentally measured data are consistent with physiological relationships. Incorrect data are rejected in the following program steps:

(i) if the respiratory quotient value calculated by (14) from $\text{O}_2$ and $\text{CO}_2$ contents in arterial and mixed venous blood deviated from the predicted physiological limit (0.15 - 1.7: incorrect data);

(ii) if the alveolar $\text{O}_2$ pressure computed by Bohr's equation did not exceed the measured arterial pressure (discrepancy of blood gases).

In the case of measured $\text{FeCO}_2$, the oxygen consumption and carbon dioxide production can be calculated directly using (19) and (20).
operation leads to an additional control on the validity of the measured data:

(iii) if the oxygen consumption calculated directly by (19) deviated more than 10% from the value calculated indirectly by (22);

(iv) if the carbon dioxide production estimated by (20) diverged more than 10% from the value obtained by (21) from directly calculated oxygen consumption and indirectly calculated respiratory quotient;

(v) if the respiratory quotient obtained by directly calculated O₂ consumption and CO₂ production diverged more than 10% from the value of respiratory quotient calculated indirectly from O₂ and CO₂ arterial and mixed venous blood contents.

RESULTS

The simulation model implemented on a PDP-85 microcomputer was evaluated by identification using catheterisation data from 131 patients. Cardiac output was computed by the model using the Fick principle from paired samples of arterial and mixed venous blood obtained in the 5th minute of breathing into a Douglas bag. The fractional concentration of oxygen in inspired and expired gas was measured with an OA101 MkII oxygen analyser (Servomex, Great Britain), pH and blood gases pressure values were determined by means of a 1302 I.T.S. analyser (USA). The fractional concentration of carbon dioxide in expired gas was not measured. Cardiac output was subsequently assessed in 37 patients by the dye-dilution method (Cariogene Hynson, Westcott and Dunning, USA; cardiodensitometer Beckman, USA).

The input data were rejected as incorrect (respiratory quotient exceeded the physiologically feasible limit) in 9 patients (7%) and further data in another 30 patients (23%) were taken out due to the discrepancy of the blood gases with Bohr's equation of alveolar gases. Blood gas values were accepted as correct data in accord with physiological principles in 92 patients (70%).

The reliability of the program was verified by a good correlation of cardiac index values calculated by the model (using the Fick principle) with those determined by dye dilution \( y = 0.612 \times x + 1.144, r = 0.734, p < 0.001 \). The difference between cardiac index values assessed by both methods did not exceed 1.01 min\(^{-1} \) m\(^{-2} \) in 78% of the patients. The reproducibility was confirmed by close correlation between the maximal and the minimal cardiac index values computed by the simulation model from different combinations of blood samples after these values had been randomly divided into two groups \( y = 0.884 \times x + 0.447, r = 0.891, p < 0.001 \). The difference between maximal and minimal cardiac index values did not exceed 0.51 min\(^{-1} \) m\(^{-2} \).

DISCUSSION

The application of the simulation model identification procedure in the evaluation of the cardiorespiratory functions provides a test of the validity of laboratory data obtained by instrumental measurements against basic physiological principles. The rapid delivery of results by the program enables invalid data to be detected in the course of cardiac catheterisation, to be eliminated immediately, and to be substituted with correct data gained from the re-examination of blood samples. These arrangements undoubtedly add significantly to the quality of parameters derived from the haemodynamic investigation.
The limits for the respiratory quotient values in the program were set rather wide so as to reject only the very extreme values arising either from incorrect sampling of blood specimens, from their inappropriate handling during transportation, or caused by faulty instrument processing. Smaller deviations in the respiratory quotient, caused for instance by irregular breathing of the patient, were accepted by the program.

The control functions of our model, as well as the accuracy of its output parameters, could be further enhanced through the direct measurement of the fractional concentration of CO₂ in inspired and expired gas and by direct estimation of arterial and mixed venous oxygen saturation. Clinical exploitation of some output data computed by simulation model identification (Table 2) have not so far been used in routine practice in haemodynamic laboratories. This is something which still remains to be explored.

Copies of the computer program used (Basic and Turbo-Pascal Versions) are available on request from the author by sending a blank IBM-compatible diskette and self-addressed envelope to Dr. Jiří Kofránek (see Address List of Contributors).

APPENDIX I. LIST OF EQUATIONS USED IN THE SIMULATION MODEL

\[
\text{pH}_{37} = A1 + A2*Y + (A3 + A4*Y) \times \log \text{PCO}_{37}/(A5 + A6*Y) \tag{1}
\]

where:

\[
Y = (A7 + \sqrt{A8 + A9*(Bmax + 0.3*Hb*(1 - SO₂_{37}))})/A10
\]

\[
A1 = 9.963500*10^{-2} - 1.03500*10^{-1} \times Hb
\]

\[
A2 = 3.516875*10^{-1} + 2.58750*10^{-1} \times Hb
\]

\[
A3 = -8.241000*10^{-1} + 2.01000 \times Hb
\]

\[
A4 = -5.276250 - 5.02500*10^{-2} \times Hb
\]

\[
A5 = 1.210000*10^{-2} - Hb
\]

\[
A6 = 2.625000 + 2.50000*10^{-2} \times Hb
\]

\[
A7 = -2.556000 - 9.44000*10^{-2} \times Hb
\]

\[
A8 = 1.387634*10^{-1} + 1.86653*10^{-1} \times Hb + 5.34936*10^{-3} \times Hb^2
\]

\[
A9 = 5.480000*10^{-1} + 1.37000*10^{-2} \times Hb
\]

\[
A10 = 2.740000*10^{-1} + 1.37000*10^{-2} \times Hb
\]

[reference: Kofránek, 1980]

\[
BE = (A10*X - A7)^2 - A8/A9 \tag{2}
\]

where:

\[
X = (A1 + A3 \times \log \text{PCO}_{37} - A5 \times \text{pH}_{37})/(A6 \times \text{pH}_{37} - A2 - A4 \times \log \text{PCO}_{37})
\]

Coefficients A1 - A10 (see (1))

[reference: Kofránek, 1980]

\[
Bmax = BE - 0.3 \times Hb \times (1 - SO₂_{37}) \tag{3}
\]

[reference: Siggaard-Andersen, 1974]

\[
SO₂_{37} = 0.9995 - 1.0000/(1 + ((P + 7)/33.7)^{3.3})
\]

\[
- 0.0050/(1 + ((P + 130)/35)^2)
\]

\[
+ 0.0045/(1 + ((P - 68)/12)^6)
\]

\[
- 0.0050/(1 + ((P - 35)/3)^4)
\]

\[
- 0.0050/(1 + ((P - 15)/4)^4)
\]

\[
+ 0.0035/(1 + ((P - 26)/3)^6)
\]

\[
+ 0.0020/(1 + ((P - 53)/8)^4)
\]

\[
- 0.0040/(1 + ((P - 40)/0.9)^4)
\]

\[
- 0.0020/(1 + ((P - 200)/95)^8)
\]

\[
+ 0.0040/(1 + ((P - 9)/3)^6)
\]

315
where:

\[ P = P_{O_37} \times 10^{0.40} \times (pH37 - 7.4) + 0.06 \log (40/P_{CO_237}) \]

[references: Kelman, 1966b; Ruiz et al., 1975]

\[ \text{O}_{2}\text{tot} = 3.0473 \times 10^{-5} \times P_{O_37} + 1.39 \times 10^{-2} \times Hb \times SO_237 \]  
(5)

[reference: Siggaard-Andersen, 1974]

\[ \text{CO}_2\text{tot} = 0.02226 \times P_{CO_237} \times (C1 + (C2 \times aH + C3)/aH^2 + C4 \times D/aH \]
\[ + C5 \times (1 - SO_237)/(aH/D)^2 + C6 \times aH/D + C7 \times P_{CO_237} \]
\[ + C8 \times SO_237/((aH/D)^2 + C9 \times aH/D + C10 \times P_{CO_237}) \]

(6)

where:

\[ aH = 10^{(9-pH37)} \]
\[ D = X0 \times SO_237X1 \times pH37X2 \times eX3 \times SO_237 + X4 \times pH37 \]
\[ X0 = 7.0388002 \times 10^{-5} \]
\[ X1 = 3.6447450 \times 10^{-4} \]
\[ X2 = 7.9099077 \]
\[ X3 = -2.0113444 \times 10^{-1} \]
\[ X4 = -1.4795026 \]
\[ C1 = 3.0700000 \times 10^{-2} - 2.2580645 \times 10^{-4} \times Hb \]
\[ C2 = 2.3038631 \times 10^{-1} - 6.7561967 \times 10^{-1} \times Hb \]
\[ C3 = 4.7648549 \times 10^{-1} - 1.3973181 \times Hb \]
\[ C4 = 5.5042129 \times 10^{-1} \times Hb \]
\[ C5 = 1.3735969 \times 10^{1} \times Hb \]
\[ C6 = 3.9800000 \times 10^{1} \]
\[ C7 = 2.2152680 \times 10^{1} \]
\[ C8 = 3.2420343 \times Hb \]
\[ C9 = 1.7900000 \times 10^{1} \times Hb \]
\[ C10 = 5.2285900 \times 10^{1} \]

[reference: Kofranek, 1980]

\[ HCO_3{}_{st} = 1.2 \times 10^{pOHst} - 6.1008 \]  
(7)

where:

\[ pH37 = (A1 \div A2 \div Y + 1.60206 \times (A3 + A4 \times Y))/(A5 + A6 \times Y) \]
\[ Y = (A7 \div \text{SQRT} (A8 \div A9 \times BE)/A10 \]

Coefficients A1 - A10 (listed in (1))

[reference: Kofranek, 1980]

\[ pH37 = pH37 - (0.0146 - 0.0065 \times (7.4 - pH37) \]
\[ - 0.00003 \times BEox \times (t - 37) \]  
(8)

[reference: Severinghaus, 1966]

\[ P_{CO_2}t = P_{CO_237} \times 10^{0.0185 \times (t - 37)} \]  
(9)

[reference: Kelman and Nunn, 1966]

\[ P_{O_2}t = P_{O_237} \times 10^{(t - 37) \times (0.0049 + 0.0261 \times (1 - e^X))} \]  
(10)

where:

\[ X = 52 \times (SO_237 - 1) \]

[reference: Severinghaus, 1979]
\[
\text{SO}_2t = \frac{(O_2\text{tot} - aO_2t \times \text{PO}_{2t})}{(1.39 \times 10^{-2} \times \text{Hb})}
\]
(11)
where:
\[
aO_2t = 5.9519 \times 10^{-5} - 1.266 \times 10^{-6} \times t + 1.3 \times 10^{-8} \times t^2
\]
[reference: Kelman, 1966a]
\[
\text{HCO}_3t = 10^{\text{pHt} - \text{pKt}} + \log (\text{aCO}_3t \times \text{PCO}_3t)
\]
(12)
\[
\text{pKt} = a_1 + a_2 \times t + a_3 \times t^2 + a_4 \times t^3
\]
\[
a\text{CO}_3t = b_1 + b_2 \times t + b_3 \times t^2 + b_4 \times t^3
\]
\[
a_1 = 6.3852 \quad b_1 = 0.0907
\]
\[
a_2 = -1.3288 \times 10^{-2} \quad b_2 = -3.3730 \times 10^{-3}
\]
\[
a_3 = 1.7364 \times 10^{-4} \quad b_3 = 6.7490 \times 10^{-5}
\]
\[
a_4 = -6.0084 \times 10^{-7} \quad b_4 = -5.4076 \times 10^{-7}
\]
[reference: Reeves, 1976]
\[
\text{Qt} = \frac{\text{VO}_2}{(O_2\text{tot}_\text{a} - O_2\text{tot}_\text{v})} = \frac{\text{VCO}_2}{(\text{CO}_2\text{tot}_\text{v} - \text{CO}_2\text{tot}_\text{a})}
\]
(13)
\[
\text{RQ} = \frac{\text{VCO}_2/\text{VO}_2 = (\text{CO}_2\text{tot}_\text{v} - \text{CO}_2\text{tot}_\text{a})/(O_2\text{tot}_\text{a} - O_2\text{tot}_\text{v})}
\]
(14)
\[
\text{Qsh/Qt} = \frac{(O_2\text{tot}_\text{C} - O_2\text{tot}_\text{a})/(O_2\text{tot}_\text{C} - O_2\text{tot}_\text{v})}
\]
(15)
\[
\text{CO}_2\text{tot}_\text{C} = \text{CO}_2\text{tot}_\text{a} - \text{RQ} \times (O_2\text{tot}_\text{C} - O_2\text{tot}_\text{a})
\]
(16)
\[
\text{VE} = \frac{\text{Vexp/time} \times (\text{Patm} - \text{PH}_2\text{O}_{\text{lab}}) \times (273.15 + t)}{((\text{Patm} - \text{PH}_2\text{O}_\text{t}) \times (273.15 + t))}
\]
(17)
where:
\[
\text{PH}_2\text{O}_\text{t} \text{ and } \text{PH}_2\text{O}_{\text{lab}} \text{ were calculated by (18).}
\]
\[
\text{PH}_2\text{O}_\text{t} = 2.4225 + 0.67734 \times t - 0.0082 \times t^2 + 0.00061 \times t^3
\]
(18)
[reference: Kofranek, 1980]
\[
\text{VO}_2 = (\text{VI} \times \text{FiO}_2 - \text{VE} \times \text{FeO}_2) \times \text{kBTPS}_\text{STPD}
\]
(19)
where:
\[
\text{VI} = \text{VE} \times \text{FeN}_2/\text{FiN}_2
\]
\[
\text{FeN}_2 = 1 - \text{FeO}_2 - \text{FeCO}_2
\]
\[
\text{FiN}_2 = 1 - \text{FiO}_2 - \text{FICO}_2 \quad (\text{FICO}_2 = 0)
\]
\[
\text{kBTPS}_\text{STPD} = (\text{Patm} - \text{PH}_2\text{O}_\text{t}) \times 273.15/(760 \times (273.15 + t))
\]
(20)
\[
\text{VCO}_2 = (\text{VE} \times \text{FeCO}_2 - \text{VI} \times \text{FICO}_2) \times \text{kBTPS}_\text{STPD}
\]
(21)
\[
\text{VO}_2 = \text{VO}_2 \times \text{RQ}
\]
(22)
\[
\text{VO}_2 = \text{VA} \times ((\text{FAN}_2/\text{FiN}_2) \times \text{FiO}_2 - \text{FAO}_2) \times \text{kBTPS}_\text{STPD}
\]
where:
\[
\text{FAN}_2/\text{FiN}_2 = (\text{PACO}_2 \times \text{RQ} \times \text{PAO}_2)/(\text{FICO}_2 \times \text{(Patm} - \text{PH}_2\text{O}_\text{t}) + \text{RQ} \times \text{FiO}_2 \times \text{(Patm} - \text{PH}_2\text{O}_\text{t}))
\]
\[
\text{kBTPS}_\text{STPD} \text{ (see (19))}
\]
\[ VA = VE - VD = VE \cdot (1 - (PAO_2 - FEO_2)/(PAO_2 - FIO_2)) \]  
(23)

where:

\[ PAO_2 = PAO_2/(Patm - PH_2Ot) \]
\[ PAO_2 = (FiO_2 \cdot (Patm - PH_2Ot) + PACO_2 \cdot (FiO_2 \cdot (1 - RQ) - 1)/RQ \]  
(24)

APPENDIX II. LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aCO_2t</td>
<td>mmol/torr</td>
<td>Carbon dioxide solubility coefficient in plasma</td>
</tr>
<tr>
<td>aO_2t</td>
<td>l-STPD/torr</td>
<td>Oxygen solubility coefficient in plasma</td>
</tr>
<tr>
<td>BE</td>
<td>mmol/l</td>
<td>Blood base excess concentration</td>
</tr>
<tr>
<td>BBox</td>
<td>mmol/l</td>
<td>Base excess concentration in virtually oxygenated blood</td>
</tr>
<tr>
<td>CO_2tot</td>
<td>l-STPD/l blood</td>
<td>Total blood carbon dioxide content</td>
</tr>
<tr>
<td>HCO_3st</td>
<td>mmol/l</td>
<td>Standard bicarbonate concentration</td>
</tr>
<tr>
<td>HCO_3t</td>
<td>mmol/l</td>
<td>Actual bicarbonate concentration at given temperature</td>
</tr>
<tr>
<td>FACO_2</td>
<td></td>
<td>Volume fraction of carbon dioxide in dry alveolar gas</td>
</tr>
<tr>
<td>FAN_2</td>
<td></td>
<td>Volume fraction of nitrogen in dry alveolar gas</td>
</tr>
<tr>
<td>FAO_2</td>
<td></td>
<td>Volume fraction of oxygen in dry alveolar gas</td>
</tr>
<tr>
<td>FeCO_2</td>
<td></td>
<td>Volume fraction of carbon dioxide in dry mixed expired gas</td>
</tr>
<tr>
<td>FeN_2</td>
<td></td>
<td>Volume fraction of nitrogen in dry mixed expired gas</td>
</tr>
<tr>
<td>FeO_2</td>
<td></td>
<td>Volume fraction of oxygen in dry mixed expired gas</td>
</tr>
<tr>
<td>FiCO_2</td>
<td></td>
<td>Volume fraction of carbon dioxide in dry inspired gas</td>
</tr>
<tr>
<td>FiN_2</td>
<td></td>
<td>Volume fraction of nitrogen in dry inspired gas</td>
</tr>
<tr>
<td>FiO_2</td>
<td></td>
<td>Volume fraction of oxygen in dry inspired gas</td>
</tr>
<tr>
<td>O_2tot</td>
<td>l-STPD/l blood</td>
<td>Total blood oxygen content</td>
</tr>
<tr>
<td>PACO_2</td>
<td>torr</td>
<td>Carbon dioxide tension in alveoli</td>
</tr>
<tr>
<td>PAO_2</td>
<td>torr</td>
<td>Oxygen tension in alveoli</td>
</tr>
<tr>
<td>Patm</td>
<td>torr</td>
<td>Atmospheric pressure</td>
</tr>
<tr>
<td>PCO_2t</td>
<td>torr</td>
<td>Blood carbon dioxide tension at given temperature</td>
</tr>
<tr>
<td>PCO_237</td>
<td>torr</td>
<td>Blood carbon dioxide tension at 37°C</td>
</tr>
<tr>
<td>PH_2Ot</td>
<td>torr</td>
<td>Vapour pressure at given temperature</td>
</tr>
<tr>
<td>PH_2Otlab</td>
<td>torr</td>
<td>Vapour pressure at room temperature</td>
</tr>
<tr>
<td>PH_2O37</td>
<td>torr</td>
<td>Vapour pressure at 37°C</td>
</tr>
<tr>
<td>pH</td>
<td></td>
<td>Plasma pH at given temperature</td>
</tr>
<tr>
<td>pH37</td>
<td></td>
<td>Plasma pH at 37°C</td>
</tr>
<tr>
<td>PO_2t</td>
<td>torr</td>
<td>Blood oxygen tension at given temperature</td>
</tr>
<tr>
<td>PO_237</td>
<td>torr</td>
<td>Blood oxygen tension at 37°C</td>
</tr>
<tr>
<td>Qah</td>
<td>l/min</td>
<td>Right-to-left shunt perfusion</td>
</tr>
<tr>
<td>Qp</td>
<td>l/min</td>
<td>Lung perfusion</td>
</tr>
<tr>
<td>Qt</td>
<td>l/min</td>
<td>Cardiac output</td>
</tr>
<tr>
<td>RQ</td>
<td></td>
<td>Respiratory quotient</td>
</tr>
</tbody>
</table>

318
\[ \text{SO}_2t \] Oxygen haemoglobin saturation at given temperature
\[ \text{SO}_237 \] Oxygen haemoglobin saturation at 37°C
\[ t \] [°C] Given temperature
\[ t_{\text{lab}} \] [°C] Room temperature
\[ \text{time} \] [min] Time for measurement of expired volume
\[ \text{VA} \] [l-STPD/min] Alveolar ventilation
\[ \text{VD} \] [l-STPD/min] Death volume ventilation
\[ \text{VE} \] [l-STPD/min] Minute ventilation
\[ \text{VI} \] [l-STPD/min] Inspired minute ventilation
\[ \text{Vexp} \] [l-STPD] Expired volume
\[ \text{WCO}_2 \] [l-STPD/min] Carbon dioxide production
\[ \text{VO}_2 \] [l-STPD/min] Oxygen consumption

Indices:
\[ a \] Arterial blood
\[ c \] Pulmonary end-capillary blood
\[ v \] Mixed venous blood

REFERENCES

Reeves, R. B., 1976, Temperature-induced changes in blood acid-base status: \[ \text{pH} \] and \[ pCO_2 \] in a binary buffer, J. Appl. Physiol., 40: 752.
Komplexní model acidobazické rovnováhy

Complex model of blood acid-base balance

Kofránek, Jiří

str. 23-60


17 str.
COMPLEX MODEL OF BLOOD ACID-BASE BALANCE
Jiří Kofránek

Laboratory of Biocybernetics, Department of Pathological Physiology, First Faculty of Medicine, Charles University of Prague.
e-mail: kofranek@email.com

Annotation

Originally, the classic Siggaard-Andersen nomogram, widely used in clinical practice for the assessment of acid-base balance, experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation of the baseline experimental data to 37°C and set a new nomogram for 37°C. Compared with the original nomogram, there are no significant deviations, if BE does not deviate by more than 10 mmol/l; the results are, however, different with the deviations exceeding 15 mmol/l. We suggested an algorithm and a program, which enables calculation of BE from pH and pCO₂ according to the original as well as adjusted normograms. However, the data, having been a base of the normogram, count with normal plasma protein and phosphate concentrations. At these conditions, BE corresponds to SID changes in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations. However, a nomogram is used in clinical practice to calculate from the data measured in blood samples tempered at 37°C. We made a simulation recalculation experimentally obtained at 38°C with the precondition of normal plasma protein concentrations.

Key words

Acid-base balance, formalised description, simulation model, blood gases, educational simulators

1. Introduction

Acid-base balance in the organism is controlled by two balances – carbon dioxide flow balance (respiration control) and strong acid production/excretion balance (regulation of acidification processes in the kidney). Both flows are connected via buffer systems. The balance disturbances result in pH changes in blood liquids. Drifts in the chemical balances of buffer systems, transport of substances between the buffer systems, H⁺/Na⁺ H⁺/K⁺ exchange between the cell and the interstitial liquid (and, in a long-term scale, washing out NaHCO₃, KHC₀₃ and, later, CaCO₃ and CaHPO₄ from the bone mineral mass in chronic acidemia) are only suppressive mechanisms in acid-base disturbances. The basic regulation organs, able to control acid-base balance (by their effect on CO₂ and H⁺/HCO₃ flows) include the respiratory system and kidney.

From the clinical point of view, the arterial blood buffer system is an important indicator of the status of acid-base balance. CO₂ retention or depletion during the change of carbon dioxide balance as well as H⁺/HCO₃ retention or depletion during the changes of strong acid production/excretion balance develop into the drift of the chemical balance in bicarbonate and non-bicarbonate buffer systems.

Labelling the total concentration of non-bicarbonate bases [Buf] – which, in fact, are the buffer bases of plasma proteins and phosphates (and haemoglobin concentrations in the whole blood) – then the total concentration of non-bicarbonate buffer bases forms the Buffer Base (BB) value:

BB=[HCO₃⁻]+[Buf]

2. Classical approach of the “Danish School” for the assessment of acid-base disturbances

The variations in pCO₂ result in pH changes; if the titration curve of pCO₂ and pH changes is plotted in the semi-logarithmic scale, these titration curves verge on lines in the range of life-compatible pH values. This precondition was a base of blood acid-base balance tests introduced in the first half of 1950s by Paul Astrup. At that time, there were no electrodes which enable direct measurement of plasma pCO₂. There were, however, relatively accurate methods of pH measurement. Astrup’s method of pCO₂ analysis (1956) was based on the following procedure: first, blood pH was measured, then, the sample was automatically equilibrated by O₂/CO₂ mixture with accurately set pCO₂. The blood sample was equilibrated with a high pCO₂ gas mixture and the equilibration was followed by measuring pH. Then, the blood was equilibrated with a mixture with low carbon dioxide partial pressure and the equilibration was followed by another pH measurement. The points obtained were plotted into a semi-logarithmic graph to create a line, used to read out pCO₂ according to baseline pH (see Fig. 1).

The Buffer Base concept made by Singer and Hastings (1948) was further improved by Siggaard-Andersen in (1960,1962), who introduced the difference of Buffer Base and its normal value - Normal Buffer Base (NBB) - as a clinically relevant factor:

BE=BB-NBB

At normal circumstances, BE values (for blood samples with any haemoglobin concentrations) are zero. They are changed during a buffer reaction with strong acid or base added.
Fig. 1 The titration curve of pH/PCO₂, variations after blood equilibration with carbon dioxide is practically a line. This therefore enabled pCO₂ determination in the tested blood sample according to the titration curve plotted after blood equilibration with low and high partial CO₂ pressure.

Siggaard Andersen used the equilibrium titration curves to determine BB and BE. He added defined amounts of strong acids or bases to blood samples with various haematocrit concentrations, changing their BE. Then, the samples were titrated and the results were plotted in log PCO₂/pH coordinates. The titration curves (being lines in the semi-logarithmic coordinates) of the blood samples with various haematocrit and the same BE always crossed in the same points (see Fig. 2). Similarly, the titration curves of the samples with various haematocrit concentrations (and various BE), but with the same BB crossed in the same points, too.

Thus, a nomogram with BE and BB curves with semi-logarithmic coordinates was obtained; the curves enabled the determination of BE and BB in the samples having been tested.

Siggaard-Andersen used this procedure to find experimentally the dependence of hydrogen ion [H⁺] concentrations or pH on pCO₂ and haemoglobin (Hb) concentrations; the results obtained were used to create clinically applicable nomograms expressing the following dependence:

\[ \text{[H⁺]} = \text{function (pCO₂, BE, Hb)}\]

In the assessment of acid-base disturbances by BE and pCO₂, it should be taken into consideration that the increase or the fall in CO₂ affects neither the total concentration of the buffer bases (BB) nor BE. The increase results in the increase in carbonic acid concentration, dissociating into bicarbonate and hydrogen ions, which are, however, completely bound to non-carbonate buffer bases [Buf]; the increase in bicarbonate concentrations therefore corresponds with the same fall in non-bicarbonate buffers with the total [HCO₃⁻]+[Buf] concentrations and, thus, BB as well as BE remaining practically unchanged. BB and BE are therefore considered pCO₂ independent. This applies for plasma exactly but not exactly for the whole blood – pCO₂ affects haemoglobin oxygenation. However, as deoxygenated haemoglobin has higher affinity to protons than oxygenated haemoglobin (the oxygenated blood therefore contains virtually higher non-bicarbonate buffer concentrations), the total concentration of buffer bases BB also depends on haemoglobin oxygen saturation (susceptible by pCO₂).

Hence, to make acid-base balance models, it is beneficial to define standardised Buffer Base oxy-value (BBox) as BB, potentially found in the blood sample with full oxygen saturation of oxyhaemoglobin (i.e. full 100% oxygen saturation of haemoglobin). Similarly, the standardised Base Excess oxy-value (BEox) is defined as BE measured in the blood sample with full oxygen saturation of oxyhaemoglobin (Kofránek, 1980). Thus, BEox is really pCO₂ independent.

It is necessary to say that the independence of pCO₂ and BEox does not apply for “in vivo” whole blood completely, as the increase in pCO₂ is connected with higher increase of bicarbonates in plasma compared with that in the interstitium; thus, part of the bicarbonates is transported into the interstitial liquid during the increase in pCO₂ (with a mild fall in BEox in acute pCO₂ increase).

BB and BE (or BBox and BEox) change after addition of a strong acid (or strong base) or bicarbonates to the blood sample. Addition of one millimol of a strong acid to one litre of blood results in BE fall by one millimol; addition of one millimol of bicarbonates (or withdrawal of one millimol of hydrogen ions by a reaction with a strong base) results in BB and BE (BBox and BEox) increase by one millimol.

The variations in dissolved CO₂ plasma concentrations (expressed as pCO₂) and BE therefore characterise carbon dioxide flow balance and the variation in strong acid production/excretion balance, respectively. Thus, pCO₂ and BE characterise the respiratory and metabolic parts of acid-base balance, respectively.
To use pH, pCO₂, and BE in clinical practice for the diagnosis of acid-base balance, so called compensation diagrams were created, expressing the effect of adaptation responses of the respiratory and renal systems to acid-base disturbances (Dell and Winters, 1970, Goldberg et al., 1973, Siggaard-Andersen, 1974, Grogono et al., 1976).

Siggaard-Andersen nomogram (expressed in the form of approximate equations) became a base for algorithm assessment in a number of laboratory automat for the measurement of acid-base balance. A certain problem was that the experimental measurements for the construction of Siggaard-Andersen nomogram were carried out at 38°C. Modern devices for the measurement of acid-base balance (allowing direct measurement of pCO₂, pH and pO₂) usually give data for samples adjusted to 37°C.

However, a more serious problem was that the titration done to create an experimental nomogram was carried out with blood with normal plasma protein concentrations (72 g/l). If the plasma protein concentrations are lower (which is not rare in critically ill patients), the points on the nomogram are shifted and all the clinical counts derived from this nomogram are incorrect.

Later, Siggaard-Andersen published certain corrections, considering various plasma protein concentrations (Siggaard-Andersen, 1977, Siggaard-Andersen et al. 1985, Siggaard-Andersen, Fogh-Andersen, 1995); however, they were not included into clinical practice properly.

3. Stewart’s “modern” approach

The abovementioned inaccuracies of the classical approach to the assessment of acid-base balance resulted in the attempt to find new methods of the description and assessment of blood acid-base balance in 1980s. The most used method was Stewart’s one (1983), improved later for clinical practice by Fencl et al. (1989, 1993, 2000).

Unlike Siggaard-Andersen’s method, Stewart’s description is limited to plasma only; however, it enables accurate description of hypo- and hyperalbuminaemia, dilute acidosis as well as concentration alkalosis. Stewart’s calculations are based on the combination of physical-chemical equations. The original Stewart’s calculations are based on simple preconditions:

1. The equation for water must apply:

\[ [H^+] [OH^-] = K'w \]

2. The constancy of the sum of weak acid concentrations (Buf.), and their dissociated buffer bases (Buf.)

\[ [Buf.] + [HBuf] = [Buf_{TOT}] \]

3. Dissociation balance of non-bicarbonate buffer system:

\[ [Buf] [H^+] = K_{buf} \times [HBuf] \]

4. Dissociation balance of bicarbonate buffer:

\[ [H^+] [HCO_3^-] = M \times pCO_2 \]

5. Dissociation balance between bicarbonate and carbonate:

\[ [H^+] [CO_3^{2-}] = N \times [HCO_3^-] \]

6. Electroneutrality:

\[ \text{SID} + [H^+] - [HCO_3^-] - [Buf] - [CO_3^{2-}] - [OH^-] = 0 \]

with SID meaning the value of “strong ion difference” (residual anion) – defined as the difference between the concentrations of fully dissociated anions and cation (expressed in mEq/l). Practically, the value can be found out by the following equation:

\[ \text{SID} = [Na^+] + [K^+] + [Mg^{2+}] + [Ca^{2+}] - [Cl^-] \]

Combining these two equations, the result is the fourth degree algebraic equation, enabling calculation of hydrogen ion concentrations in dependence on SID, the total concentration of weak acids and their buffer bases [Buf_{TOT}] and pCO₂ (the dependent variable is underlined in the equation, independent variations and constants are in bold and italic, respectively):

\[ [H^+]4 + ([\text{SID} + K_{buf}] \times [H^+]3 + (K_{buf} \times [\text{SID} - [Buf_{TOT}])] \times K'w - M \times pCO_2) \times [H^+]2 - ([K_{buf} \times (K'w)^2 + M \times pCO_2] \times N \times M \times pCO_2 \times [H^+] - K'w \times N \times M \times pCO_2 = 0 \]

Solving of the equation gives hydrogen ion concentration, depending on the respiratory part of acid-base balance – i.e. pCO₂ and, moreover, on the respiratory part of SID independent metabolic parameters as well as on the total concentration of non-bicarbonate bases and acids [Buf_{TOT}]:

\[ \text{pH} = \text{function} (\text{pCO}_2, \text{SID}, [\text{Buf}_{TOT}]) \]

The total concentration of non-bicarbonate bases [Buf_{TOT}] is related to the total plasma protein (albumin) concentration. More detailed studies consider the total phosphate concentrations, too. The results of these studies are relationships enabling (by means of a computer programme) calculation of pH (and other variables such as bicarbonate concentrations etc.) from pCO₂, SID, and total phosphat [Pi] and plasma albumin [Alb_{TOT}] concentrations (see, for example, Watson, 1999):

\[ \text{pH} = \text{function} (\text{pCO}_2, \text{SID}, [\text{Alb}_{TOT}], [\text{Pi}]) \]

One of the most detailed quantitative analyses of plasma acid-base balance (Figge, 2009) improving Figge-Fenc’s model (Figge et al. 1992) even corrects the effect of externally added citrate [Cit] in the plasma sample used for the laboratory test.

\[ \text{pH} = \text{function} (\text{pCO}_2, \text{SID}, [\text{Alb}_{TOT}], [\text{Pi}], [\text{Cit}]) \]

4. Benefits and drawbacks of Stewart’s approach

Mathematical relationships between the variables derived from the quantitative physical-chemical analysis enable calculation of dependent variables – pH, being a base for other dependent variables, i.e. bicarbonate concentrations – from independent variables (i.e. pCO₂, SID, albumin and phosphate concentrations or, as the case may be, concentrations of the citrate added to the plasma sample).

Stewart’s approach enables more detailed description of some of the pathophysiological conditions (the effect of hypo- and hyperalbuminaemia on acid-base balance, dilution acidosis or concentration alkalosis) and, at first site, gives the clinicians the feeling of better insight into the ethiology of acid base disturbances. To determine “independent” variables, used for the calculation of other acid-base parameters, it is
necessary to do explicit measurements of phosphate, Na⁺, Cl⁻, HCO₃⁻ and other ion concentrations, which clinicians work in their diagnostic forethought with.

On the contrary, the drawbacks of Stewart’s theory include the fact that he works with plasma only. Moreover, some Stewart’s followers, fascinated by the possibility to calculate acid-base parameters - pH (and proper concentrations of bicarbonates, carbonates and non-bicarbonate acids) – from independent variables (pCO₂, SID, [Alb], [Pi]), often make objectively incorrect conclusions in their interpretation. In the calculation, the independence of baseline variables, particularly SID, is meant not in a causal but in a strictly mathematical meaning. This is, however, often forgotten in clinical-physiological practice, which often results in incorrect interpretation of the causality relationship between the causes of acid-base disturbances.

5. “Mathematical wizardry” of Stewart’s followers

A number of Stewart’s followers considered his mathematical relationships as “oracle” – incorrect causal relationships are deducted from substantially correct mathematical relationships. The causality of mathematical calculations (where independent variables are calculated from dependent ones) is confused with the causality of pathophysiological relationships.

For example, some authors deduct that one of the elementary causal relationships of acid-base disturbances are changes in SID concentrations. Sirker et al. (2001) even states that “the transport of hydrogen ions through membranes (via hydrogen channels) does not affect their actual concentration. Direct removal of H⁺ from one compartment can alter neither the value of any independent variable nor [H⁺] concentration... the equilibrium dissociation of water balances any fluctuations in [H⁺] concentrations and serves as an inexhaustible source or sink for H⁺ ions”.

There is no rational explanation for the opinion that SID (as a mathematical construct, not a physical-chemical characteristic) affects [H⁺] concentrations in a certain mechanistic way to keep electroneutrality – any buffer reaction is a shifted chemical balance only; thus, there is no way how they could affect the electroneutrality themselves (without membrane transport).

6. Are both approaches significantly different?

Excited debates lead by supporters of both theories in international journals (e.g. Dubin et al. 2007, Dubin 2007, Kaplan 2007, Kurz et al., 2008, Kelum 2009) might suggest that both theories are completely different and their applicability will be proved during the time. In fact, both theories are complementary. If similar conditions of their applicability are observed (i.e. they are used for plasma with normal albumin and phosphate concentrations only), the results are, in fact, identical. It is obvious that if one of the theories is used out of the area which it was proposed for, it fails and the other theory seems to be more accurate. For example, reduced protein concentrations do not correspond to the conditions determined experimentally for Siggaard-Andersen nomogram; if this nomogram is used for BE assessment in patients with hyperalbuminaemia, incorrect values are obtained. In this case, the use of Stewart’s method prevents incorrect diagnosis. On the other hand, Stewart does not calculate with the effect of such an important blood buffer - haemoglobin in erythrocytes. Stewart’s approach is applicable neither for the calculation of the amount of infusion solutions for the correction of the acid-base disturbance nor for the assessment of the grade of respiratory and renal compensation of the acid-base disturbance. During the bedside diagnostics it is advisable to consider both theories and to realise their benefits and limits (Kelum, 2005).

The accordance and differences of both approaches are as follows.

Both Stewart and Siggaard-Andersen use pCO₂ as a parameter describing the respiratory part of acid-base balance. According to the “Danish School”, the metabolic part is represented by BB or its deviation from the norm – BE. According to Stewart, the metabolic part is represented by SID as the difference of fully dissociated positively and negatively charged anions and cations – in the respect of keeping the principle of electroneutrality, it might seem at first sight that, numerically, SID is identical with plasma BB (Fig. 3).

SID = [HCO₃⁻] + [Buf] = BB

But is it true really? Siggaard-Andersen (2006) states so. However, focused on the importance of non-bicarbonate bases, certain differences can be seen.

Plasma non-bicarbonate bases include phosphates and plasma proteins – particularly albumin (the effect of globulins on acid-base balance is insignificant). The albumin hydrogen ion can be bound to the following negatively charged amino acids (Figge, 2009): cysteine, glutamic and aspartic acid, tyrosine and carboxyl end of protein polymer. Labelling these binding sites as Alb⁻, the binding of hydrogen ions can neutralise the electric charge (as presumed in the classical Stewart’s theory):

Alb⁻ + H⁺ = HAlb

Hydrogen ions van, however, be bound to imidazol cores of histidine as well as to arginine, lysine and NH₂-end of an albumin molecule. Labelling these binding sites as Alb⁻, then the binding of hydrogen ions results in the creation of positive charge:

**Fig. 3 SID and BB are nearly identical. The variations in SID and BB are completely identical: dSID=dBB.**
Alb + H⁺ = HAlb⁺

Labelling the total concentrations of non-bicarbonate bases by Stewart and Siggaard-Andersen as [Buf⁺], respectively, a small difference can be observed (the concentrations are considered in miliequivalents):

\[ [\text{Buf}^+] = [\text{PO}_4^{2-}] + [\text{HPO}_4^{2-}] + [\text{H}_2\text{PO}_4^-] + [\text{Alb}] - [\text{HAlb}^+] \]

\[ [\text{Buf}^-] = [\text{PO}_4^{3-}] + [\text{HPO}_4^{2-}] + [\text{H}_2\text{PO}_4^-] + [\text{Alb}] + [\text{Alb}] \]

The concentration of non-bicarbonate bases is a bit higher by Siggaard-Andersen, as the relationship [Alb]>[HAlb] applies at physiological conditions. This obviously suggests the difference between normal SID (around 38 mmol/l) and normal plasma BB (stated as 41.7 mmol/l).

However, as it applies that the variation in [Alb] concentrations is related to the variation in [HAlb⁺] concentrations:

\[ d[\text{Alb}] = d[\text{HAlb}⁺] \]

the variation in the concentrations of non-bicarbonate bases by Siggaard-Andersen will be identical with that of non-bicarbonate bases by Stewart:

\[ d[\text{Buf}^+] = d[\text{Buf}^-] \]

The variation in BB or BE is therefore the same as that of SID:

\[ \text{dBB} = \text{dSID} \]

Thus, it would be meaningful for clinical purposes to calculate normal SID for various plasma and phosphate concentrations: NSID= function ([Alb\text{\textsubscript{st}}], [Pi]), similarly as Siggaard-Andersen calculates NBB as a variable dependent on haemoglobin concentrations. It would not be complicated in any respect.

However, the problem is that what circulates in the blood vessels is not plasma only, but plasma and erythrocytes. A more accurate quantitative analysis requires considering the whole blood and it is also necessary to re-evaluate and connect both the approaches.

The outcome of the connection will be the sufficiently quantified Figge-Fečen’s model of plasma (Figge, 2009) and experimental data for the whole blood, included in Siggaard-Andersen nomogram.

7. Formalisation of Siggaard-Andersen nomogram

The first step necessary for the realisation of this connection is to formalise Siggaard-Andersen nomogram.

The literature describes a number of equations which formalise Siggaard-Andersen nomogram with higher or lower accuracy (e.g. Siggaard-Andersen et al. 1988). Lang and Zander (2002) compared the accuracy of BE calculation in 7 approximations of various authors. The most accurate approximation was that of Van Slyk equation by Zander (1995). Surprisingly, it was, however, shown that the formalisation of Siggaard-Andersen nomogram from 1980, used in a lot of our models in the past, approximated Siggaard-Andersen nomogram with higher accuracy than the relationships having been published later (Fig. 4)

It is possible to try further specification of our approximation.
models have been identified for 37°C. Thus, it was necessary to correct Siggaard-Andersen nomogram from 38°C to the standard temperature of 37°C.

In clinical practice, the temperature corrections of pH and pCO₂ from t° to the standard temperature of 37°C are based on simple relationships, e.g. (Ashwood et al. 1983):

\[ \text{pH}_{37} = \text{pH}_t - 0.0147(37-t°) \]

\[ \log_{10}(\text{pCO}_2_{37}) = 0.019 \log_{10}(\text{pCO}_2_{37°}) + (0.02273 - 0.00126 \times (7.4 - \text{pH}_{37°C})) \]

For proper temperature corrections of Siggaard-Andersen nomogram it is advisable to use the more accurate relationship by Ashwood et al. (1983):

\[ \log_{10}(\text{pCO}_2_{37}) = \log_{10}(\text{pCO}_2_t) + (0.02273 - 0.00126 \times (7.4 - \text{pH}_{37°C})) \]

\[ (37 - t°) - 0.0000396(37^2- t^2) \]

However, to correct Siggaard-Andersen nomogram from 38°C to 37°C, it is insufficient to transfer simply log₁₀ pCO₂ and pH, representing the coordinates of BE and BB curves in Siggaard-Andersen nomogram, from one temperature to another.

The trouble is that, according to the definition, BE is calculated as a titratable base in blood titration to the standard values (pCO₂=40 torr and pH=7.4). BE is zero at these standard values. Thus, the zero point of the BE curve, where all titration curves of blood with various haematocrit cut each other, lies in the coordinates of pH=7.4, and pCO₂=40 torr. Using a simple re-calculation of the values from 38°C to 37°C, the zero point of the BE curve is transferred to pCO₂=38.2195 torr and pH=7.421 then (Fig. 8). Our aim is, however, to achieve that pCO₂ and pH corresponding to zero BE are 40 torr and 7.4 on the curve for 37°C.

Thus, standard pH and pCO₂ are re-calculated from 37°C to 38°C as follows:

\[ \text{pH}_{38} = 7.4 \rightarrow \text{pH}_{38} = 7.3878 \]

In the points pH_{37°C} = 7.4 and pCO₂_{37°C} = 40 torr, there is an intersection of plasma and erythrocyte titration curves with various haematocrit and BE=0 mmol/l. After the temperature increase by one degree centigrade, all lines are shifted with the intersection in the same point (pH_{38°C} = 7.3878 and pCO₂_{38°C} = 41.862 torr); BEs are, however, non-zero and differ for each blood sample.
The dependence of BE on haemoglobin concentrations at 37°C – pH and pCO₂, from particular curves in the previous figure were re-calculated from 38°C to 37°C. The curves cut each other in the zero point of BE curve for 37°C, which lies on the coordinates pCO₂=40 torr and pH=7.4.

To obtain a set of the values characterising the BE curve for Sigaard-Andersen nomogram corrected to 37°C, it is advisable to carry out simulation experiments with carbon dioxide blood titration in blood samples with various haemoglobin concentrations for each BE, in the condition of full oxygen saturation (see the calculation algorithm scheme). Correction factor dBE_{37°C} (depending on haemoglobin concentration and corresponding to BE zero value at 37°C) was always added to each BE_{37°C}. This correction shift was a base for BE_{37°C}.

\[
\text{BE}_{37°C} = \text{BE}_{38°C} + \text{dBE}_{37°C}
\]

A set of pH_{37°C} was calculated from a set of BE_{37°C} and pCO₂ from Sigaard-Andersen nomogram (by means of BEINV algorithm – see Fig. 7). pCO₂_{37°C} and pH_{37°C} were then re-calculated to the values corresponding to 37°C.

This procedure enabled obtaining the titration curves for 37°C. The intersections of the curves with the standard values of pH=7.4 and pCO₂=40 torr (see Fig. 14).
For new coordinates of BE curves, see Fig. 15 and 16.

The calculation of new coordinates of BB curves (i.e. the coordinates where the curves – or lines in the semi-logarithmic scale – of blood samples with the same BB cut each other is simpler. In anaerobic heating (or cooling) must apply that:

\[ d[HCO_3^-] = -d[Buf^-]+d[H^+] , \]

as \( d[H^+]<c d[HCO_3^-] \),

thus, it applies that \( d[HCO_3^-]=d[Buf^-] \), i.e. BB do not vary; thus:

\[ BB_{37°C} = BB_{38°C} \]

The pH and pCO\(_2\) at 38°C are re-calculated from 38°C to 37°C by Ashwood et al. (1983) – however, the transformation curve will correspond to the same BB (but to a different BE value).

It therefore suggests that the coordinates of the points of the BB curve of Siggaard-Andersen nomogram for 37°C can be obtained by the transformation of the coordinates of the points on the BB curve of the original Siggaard-Andersen nomogram (representing the coordinates of the intersections of the transformation curves with the same BB value at 38°C) into new values by Ashwood et al. (1983).

BBs depend on BE normal BB (NBB). Although BB\(_{38°C}\) and BB\(_{37°C}\) are the same, it is possible to show that their normal values (NBB\(_{37°C}\) and NBB\(_{38°C}\)) are different for 37°C and 38°C:

\[ NBB_{37°C} = BB_{37°C} - BE_{37°C} = BB_{38°C} - BE_{37°C} \]

As (see above):

\[ BE_{37°C} = BE_{38°C} - dBE_{38°C} \]

then:

\[ NBB_{37°C} = BB_{38°C} - BE_{38°C} + dBE_{38°C} = NBB_{38°C} + dBE_{38°C} \]

The value of dBE\(_{38°C}\) shift is calculated by the algorithm stated in Fig. 13 and depends on haemoglobin concentration. The
consequent dependence can be linearised by the following relationship (Fig. 17):

$$dBE = 0.3 - 0.018 \text{cHb}$$

where cHb is haemoglobin concentration in g/100ml.

NBB$^{38\degree C}$ is calculated by the known, in clinical practice used, relationship (Siggaard-Andersen, 1960):

$$NBB^{38\degree C} = 41.7 + 0.42 \text{cHb}$$

The substitution of NBB$^{37\degree C}$ results in a slightly different relationship:

$$NBB^{37\degree C} = 42.0 + 0.402 \text{cHb}$$

BB$^{37\degree C}$ value will be calculated from BE$^{37\degree C}$ and haemoglobin concentration:

$$BB^{37\degree C} = 42.0 + 0.402 \text{cHb} + \text{BE}^{37\degree C}$$

For the comparison of the curve Siggaard-Andersen nomograms for 37°C and 38°C, see Fig. 18 and Table 1.

In clinical laboratory practice, data (pH and pCO$_2$) are measured at the standard temperature of 37°C; however,

Fig. 17 Linearization of the dependence of BE shift on haemoglobin concentration (cHb) expressed in g/100 ml during temperature change from 37°C to 38°C.

Fig. 18 Correction of the values on BE and BB curves of Siggaard-Andersen nomogram (created originally for 38°C) to the standard temperature 37°C.
they are assessed (BE calculation) by means of Siggaard-Andersen nomogram, created originally for 38°C. Thus, the comparison of the course of the titration curves according to the original and corrected Siggaard-Andersen nomogram (Fig. 19) is interesting in the view of clinical outcomes. It is obvious that noticeable deviations occur as late as with BE under 10 mmol/l and more significant ones at BE exceeding 15 mmol/l.

Table 1 Coordinates of BE and BB curves for original (37°C) and corrected (37°C) Siggaard-Andersen nomogram.

Table 1

<table>
<thead>
<tr>
<th>BB [mmol/l]</th>
<th>37°C</th>
<th>38°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>pH</td>
<td>pCO₂ [torr]</td>
<td>pH</td>
</tr>
<tr>
<td>14</td>
<td>6,903</td>
<td>4,0</td>
</tr>
<tr>
<td>15</td>
<td>6,904</td>
<td>9,6</td>
</tr>
<tr>
<td>16</td>
<td>6,904</td>
<td>14,9</td>
</tr>
<tr>
<td>17</td>
<td>6,905</td>
<td>20,1</td>
</tr>
<tr>
<td>18</td>
<td>6,906</td>
<td>25,1</td>
</tr>
<tr>
<td>19</td>
<td>6,908</td>
<td>29,8</td>
</tr>
<tr>
<td>20</td>
<td>6,911</td>
<td>34,3</td>
</tr>
<tr>
<td>21</td>
<td>6,916</td>
<td>38,6</td>
</tr>
<tr>
<td>22</td>
<td>6,923</td>
<td>42,7</td>
</tr>
<tr>
<td>23</td>
<td>6,932</td>
<td>46,6</td>
</tr>
<tr>
<td>24</td>
<td>6,940</td>
<td>50,3</td>
</tr>
<tr>
<td>25</td>
<td>6,949</td>
<td>53,9</td>
</tr>
<tr>
<td>26</td>
<td>6,958</td>
<td>57,2</td>
</tr>
<tr>
<td>27</td>
<td>6,967</td>
<td>60,4</td>
</tr>
<tr>
<td>28</td>
<td>6,976</td>
<td>63,3</td>
</tr>
<tr>
<td>29</td>
<td>6,985</td>
<td>66,1</td>
</tr>
<tr>
<td>30</td>
<td>6,994</td>
<td>68,7</td>
</tr>
<tr>
<td>31</td>
<td>7,004</td>
<td>71,1</td>
</tr>
<tr>
<td>32</td>
<td>7,013</td>
<td>73,4</td>
</tr>
<tr>
<td>33</td>
<td>7,022</td>
<td>75,6</td>
</tr>
<tr>
<td>34</td>
<td>7,032</td>
<td>77,6</td>
</tr>
<tr>
<td>35</td>
<td>7,042</td>
<td>79,4</td>
</tr>
<tr>
<td>36</td>
<td>7,051</td>
<td>81,0</td>
</tr>
<tr>
<td>37</td>
<td>7,060</td>
<td>82,5</td>
</tr>
<tr>
<td>38</td>
<td>7,070</td>
<td>83,9</td>
</tr>
<tr>
<td>39</td>
<td>7,080</td>
<td>85,1</td>
</tr>
<tr>
<td>40</td>
<td>7,090</td>
<td>86,3</td>
</tr>
<tr>
<td>41</td>
<td>7,100</td>
<td>87,3</td>
</tr>
<tr>
<td>42</td>
<td>7,110</td>
<td>88,3</td>
</tr>
<tr>
<td>43</td>
<td>7,120</td>
<td>89,0</td>
</tr>
<tr>
<td>44</td>
<td>7,131</td>
<td>89,7</td>
</tr>
<tr>
<td>45</td>
<td>7,141</td>
<td>90,3</td>
</tr>
<tr>
<td>46</td>
<td>7,151</td>
<td>90,7</td>
</tr>
<tr>
<td>47</td>
<td>7,162</td>
<td>91,1</td>
</tr>
<tr>
<td>48</td>
<td>7,173</td>
<td>91,4</td>
</tr>
<tr>
<td>49</td>
<td>7,183</td>
<td>91,6</td>
</tr>
<tr>
<td>50</td>
<td>7,194</td>
<td>91,7</td>
</tr>
<tr>
<td>51</td>
<td>7,205</td>
<td>91,8</td>
</tr>
<tr>
<td>52</td>
<td>7,215</td>
<td>91,8</td>
</tr>
<tr>
<td>53</td>
<td>7,226</td>
<td>91,7</td>
</tr>
<tr>
<td>54</td>
<td>7,237</td>
<td>91,5</td>
</tr>
<tr>
<td>55</td>
<td>7,248</td>
<td>91,2</td>
</tr>
<tr>
<td>56</td>
<td>7,260</td>
<td>90,8</td>
</tr>
<tr>
<td>57</td>
<td>7,271</td>
<td>90,4</td>
</tr>
<tr>
<td>58</td>
<td>7,282</td>
<td>89,9</td>
</tr>
</tbody>
</table>

9. Erythrocytes and plasma

Fig. 19 Comparison of the titration curves calculated according to original and corrected Siggaard-Andersen nomogram.
Now, Siggaard-Andersen nomogram is formalised for the same temperature, which detailed models of plasma acid-base balance, created by Stewart’s model, are identified for. These models (e.g. Figge 2009), anyhow considering the details of the effect of the dissociation constants of particular amino acids in an albumin molecule, entirely neglect the effect of such a substantial buffer as haemoglobin in erythrocytes. On the other hand, the drawback of the models based on experimental data derived from Siggaard-Andersen nomogram, is a precondition of normal plasma protein concentration.

The aim of this work is to connect both approaches into one model, potentially usable as a subsystem of the complex model of homeostasis in the organism with the possibility to simulate complex osmotic, ion, volume and acid-base disturbances.

First, using the experimental data from Siggaard-Andersen nomogram, the titration curves of plasma and erythrocytes should be separated – the result should be a model of the buffer behaviour of erythrocytes, connected with the detailed model of plasma acid-base balance, created by Stewart’s approach, regarding various plasma protein and phosphate concentrations.

Siggaard-Andersen verified experimentally that the curves of plasma and blood samples with various haematocrit and the same BE cut each other in one point on the BE curve (see Fig. 2). Similarly, the curves of blood samples with the same BB cut each other in one point on the BB curve. It raises a question, why the BB and BE titration curves cut each other in the same points on Siggaard-Andersen nomogram?

To reply this question, it is necessary to realise that blood titration with carbon dioxide results in the increase in bicarbonate concentrations in plasma and erythrocytes during the increase in pCO2.

Regarding the plasma itself by Stewart – then, during plasma titration with carbon dioxide, the sum of bicarbonates and all non-bicarbonate buffer bases, forming BB, and SID, are unchanged (Fig. 20) – SID and pCO2 are therefore mutually independent variables, which, together with another independent variable, plasma protein concentration, determines the value of the dependent variable – pH.

This basic Stewart’s canon does not apply in blood (see Fig. 21) – in the titration with carbon dioxide, plasma SID, corresponding (with the abovementioned objections) with BBp, varies. The increase in pCO2 causes the increase in BBp (and SID), whereas the decrease in pCO2 causes the decrease in BEp. As the erythrocyte has more non-bicarbonate bases (particularly due to haemoglobin) than plasma, and the dissociation reaction of carbonic acid is more shifted to the right, there is a higher increase in bicarbonate concentrations in erythrocytes than in plasma. Bicarbonates are transported into plasma by the concentration gradient (by exchange for chloride ions). Thus, the increase in CO2 concentrations is associated with the decrease or increase in BB concentrations in erythrocytes or plasma, respectively.

Blood titration with carbon dioxide helps achieve pCO2 at which BB concentrations in erythrocytes and plasma equilibrate (BB = BE). This value determines the place where the titration curves with the same total BB and various haematocrit (Hk) will cut each other on Siggaard-Andersen nomogram.

Fig. 20. Plasma titration with carbon dioxide – BE, BBp and SID do not vary. Thus, pCO2 and SID are mutually independent.

As:

\[
BB = BB_p (1 - Hk) + BB_e Hk = BB_p + Hk (BB_e - BB_p)
\]

The second member of the sum is zero with BBp = BBp and the whole blood BB does not depend on haematocrit. With this pCO2 (and proper plasma pH) when BBp = BBp, the blood exert any value of haematocrit; all titration curves of blood samples with various haematocrit therefore cut each other in this point. Thus, the BB curve on Siggaard-Andersen nomogram is a geometric site of the points where plasma and erythrocytes have the same buffer base concentrations, as at BBp = BBp, the whole blood BB does not depend on haematocrit (Hk):

A similar consideration applies for the BE curve, too. As:

\[
BE = BE_p (1 - Hk) + BE_e Hk = BE_p + Hk (BE_e - BE_p)
\]

the second member of the sum equals zero at BBp = BBp, then and the whole blood BE does not depend on haematocrit (Hk) or haemoglobin concentration. Thus, the BE curve on Siggaard-Andersen nomogram is a geometric site of the points with the same BE in the whole blood and plasma, as the whole blood BE does not depend on haematocrit at proper pCO2 and pH, when BEp = BEp.

Fig. 21 Blood titration with carbon dioxide – SID varies during pCO2 changes (thanks to the exchange of HCO3 for Cl). SID and pCO2 in the whole blood are not mutually independent.
The BE curve can also be interpreted in other way. Regarding the fact that BE is the difference between BB and normal proper NBB for the given haemoglobin concentration, then the precondition of the equality of BE in plasma and erythrocytes means:

$$BB_e - NBB_e = BB_p - NBB_p$$

This can be specified:

$$BB_e - BB_p = NBB_e - NBB_p = constant$$

This means that the BE curve can be interpreted as the geometric site of the points (i.e. pCO$_2$ and pH values) with a constant difference between BB in erythrocytes and plasma, which equals the difference between the proper values in erythrocytes and plasma (pCO$_2$=40 torr and plasma pH=7.4).

If the equation NBB$_{pcO2}$ = 41.7 + 0.42 chb applies (Siggaard-Andersen, 1962), then haemoglobin concentration in erythrocytes chb = 33.34 g/100ml is NBB$_e$-NBB$_p$=0.42×33.34 =14 mmol/l (according to our correction of Siggaard-Andersen nomogram, this value was 0.402×33.34 =13.4 mmol/l for 37°C).

Siggaard-Andersen used the mixture of O$_2$ - CO$_2$ for blood titration with fully oxygen-saturated blood – in fact, the BE curves are those for fully oxygenated blood – i.e. the abovementioned standardised oxyvalues of Base Excess – BEox (Kofránek, 1980). BE or BB exert a linear increase in haemoglobin oxygen desaturation:

$$BE = BEox + 0.2 \cdot cHB \cdot (1-SO_2)$$

where chb is haemoglobin concentration [g/100ml] and SO$_2$ is haemoglobin oxygen saturation (Siggaard-Andersen 1988).

10. Separation of plasma and erythrocyte titration curves on Siggaard-Andersen nomogram

It is recommended to test if it is possible to make a model of blood acid-base balance from the experimental data on Siggaard-Andersen nomogram as a combination of the models of plasma and erythrocyte titration curves (Fig 22). The titration curves (plotted as lines in the semi-logarithmic scale) can be read out direct from the nomogram. The titration curves of erythrocytes can be obtained from the nomogram as follows: chose the blood concentration of haemoglobin 33.34 g/100 ml, which is the value with haematocrit having the value of one. The titration curve of this “virtual blood” with carbon dioxide follows pH variations (measured on the outer side of the erythrocyte) during pCO$_2$ changes. The titration curve of the blood with a given haemoglobin and, thus, haematocrit concentrations chb (in g/100ml blood).

$$Hk = \frac{cHB}{33.34}$$

(supposing the normal haemoglobin concentration in erythrocytes 33.34 g/100ml) will lie between the titration curves of plasma and erythrocytes in the semi-logarithmic coordinates log$_{10}$ (pCO$_2$) – pH. It will cut the curves for plasma and erythrocytes in a point of the BE curve. As non-bicarbonate buffers (haemoglobin and phosphates) have a higher buffer capacity in erythrocytes than those in plasma (plasma proteins and phosphates), and non-bicarbonate bases in erythrocytes bind more hydrogen ions than those in plasma during blood titration with increasing concentrations of carbon dioxide, the concentration of bicarbonates increases more significantly in the erythrocyte than in plasma. The consequence is the transfer of bicarbonates between the erythrocyte and plasma (accompanied with a counter chloride transport). Labelling the amount of bicarbonates in 1 litre, transferred from erythrocytes into plasma during blood titration with carbon dioxide: mHCO$_3$ep [mmol/l], then the variations in plasma BE and BB is:

$$dBB_p = dBE_p = mHCO_3ep/(1-Hk)$$

The corresponding variation of BE in erythrocytes is:

$$dBB_e = dBE_e = mHCO_3ep/Hk$$

Choosing, for example, haemoglobin concentration 15 g/100 ml (and haematocrit concentration 15/33.34=0.4449) for the transfer of 1mmol of bicarbonate, there will be an increase and decrease in plasma and erythrocyte BE as well as BB concentrations by 1/(1-0.4449)=1.8015 mmol/l and by 1/0.4449=2.2477 mmol/l, respectively. There will be left and right shifts on plasma and erythrocyte titration curves (see Fig. 23), respectively – their intersection corresponds with the point on the titration curve with haemoglobin concentration 15 g/100 ml, in which 1 ml of bicarbonates were transferred from erythrocytes into plasma during the increase of pCO$_2$ from the baseline value of 40 tor. As seen in Figure 23, this intersection lies on the titration curve with haemoglobin concentration 15 g/100 ml, modelled according to the data in Siggaard-Andersen nomogram (by means of the abovementioned function BEINV). Similarly, this curve includes the intersections of the left and right s of plasma and erythrocyte curves after the transfer of 2 and 1 mmol of bicarbonates from erythrocytes into plasma (during pCO$_2$ increase) and from plasma into erythrocytes (during pCO$_2$ increase), respectively.

Figures 24 and 25 show the results of the modelling of the titration curves for blood titration with carbon dioxide at BE -10 mmol/l and 10 mmol/l. Fig. 26 shows the results of the modelling of blood titration with carbon dioxide in the range of BE -20 to 20 mmol/l.
It has been shown that the titration curves modelled by means of the intersections of the shifts of plasma and erythrocyte titration curves (due to the transfer of bicarbonates between the erythrocyte and plasma) copy the titration curves modelled direct by Siggaard-Andersen nomogram with a sufficient accuracy.

It therefore means that the modelling of blood titration with carbon dioxide can be based on the combination of plasma and erythrocyte titration curves. The modelling of blood titration with varied plasma protein concentration can be based on the combination of plasma titration curve with various plasma protein concentrations (for which, however, Siggaard-Andersen nomogram does not apply) – for example by Figge-Fenc’ s model (Figge, 2009), and erythrocyte titration curve (obtained from the experimental data of Siggaard-Andersen nomogram, corrected to 37°C).

11. Connection of the erythrocyte model by Siggaard-Andersen nomogram, corrected to 37°C and Figge-Fenc’s model of plasma

Fig. 27 shows erythrocyte titration curves with various BE by Siggaard-Andersen nomogram – the erythrocytes are modelled as blood with haemoglobin concentration 33.34 g/100ml (corresponding to the proper haematocrit value of 1). In the semi-logarithmic scale, these curves are lines with variable slopes (k) and offset (h), depending on BE concentrations in erythrocytes (BEer).

The erythrocyte titration curves will be approximated according to the following relationships:

\[
\log_{10}(pCO2) = k \cdot pH + h
\]

\[
k = f(BE_{er})
\]

\[
h = g(BE_{er})
\]

Functions “f” and “g” are approximated by polynomial regression according to the data from Siggaard-Andersen nomogram, corrected to 37°C (see Fig. 28 and 29).
pH is calculated according to the grade of desaturation (from sO2) and BEOx. This value is considered initial for plasma and erythrocytes (BE). pH is calculated from pCO2.

The erythrocyte model is connected with the plasma model. Figge-Fencl’s model (Figge, 2009), combined, in addition, with the effect of globulin concentrations (calculated by means of their “buffer value” by Siggaard-Andersen, 1995), was selected as a plasma model. BEINV function calculates blood pH in dependence on pCO₂, total phosphate (Pitot), albumin (Alb), globulin (Glob) and haemoglobin concentrations as well as on standardised oxyvalues BEOx, (i.e. BE found in fully oxygenated blood), pCO₂ and haemoglobin oxygen saturation:

\[ \text{pH}_{\text{ery}} = \text{BEINV}(\text{pCO}_2, \text{BE}_{\text{er}}) \]

For the principle of the calculation and for the algorithm itself, see Fig. 31 and 32, respectively.

First, BE is calculated according to the grade of desaturation (from sO2) and BEOx. This value is considered initial for plasma and erythrocytes (BE). pH is calculated from pCO₂.
**Fig. 31** Principle of the calculation of the whole blood titration curves. At given BE plasma and erythrocytes titration curves (plasmaBE and erythrocytesBE) have a different slopes, hence at given pCO2 a different pH can be calculated. Searched blood titration curve (bloodBE) lies between plasmaBE and erythrocytesBE curves. In the blood at given haematocrit (Hk) plasma and erythrocyte BE (BEp and BEer) is shifted because of HCO3-/Cl- erythrocyte-plasma exchanges. New titration curves of plasma and erythrocytes (plasmaBE and erythrocytesBE) can be calculated. Algorithm seeks the intersection of bloodBE and erythrocytesBE curves at given pCO2.

However, the plasma titration curve has a smaller slope than that for erythrocytes (see Fig. 31) and plasma pH (H(BEp)) is calculated according to plasma BE (BEp); pH on the outer side of erythrocytes (H(BEer)), calculated according to erythrocyte BE (BEer), is different. Then, the transfer of bicarbonates between plasma and erythrocytes is calculated by iteration – the transfer causes variations in plasma (BEp) and erythrocyte (BEer) BEs – the ratio of BE variations in erythrocytes and plasma depend on haematocrit. The iteration converges to the final value in plasma calculated according to both erythrocyte and plasma BEs (pH = pH(BEer) = pH(BEp)).

The algorithm also calculates the normal SID (NSID) – i.e. the SID, in which pH = 7.4 with the given haemoglobin, albumin, globulin and phosphate concentrations and pCO2 = 40 torr.

There is a wider definition of BE in this model compared with classical Siggaard-Andersen’s nomogram interpretation – its normal value depends not only on haemoglobin concentrations but also on albumin, globulin and phosphate concentrations - like Siggaard-Andersen’s van Slyke equation (Siggaard-Andersen, 1977, 2006). Unlike in classical plasma models by Stewart and his followers, this model enables to demonstrate that the relationship between SID and pCO2 does not apply in the whole blood. The model (and the related formalised relationships) can be used in a number of clinical-physiological calculations.

For the model, including its source text and the description of all used mathematical relationships and algorithms, see [www.physiome.cz/acidbase](http://www.physiome.cz/acidbase).

11. Conclusion

Siggaard-Andersen nomogram was recalculated from original 38°C to standard 37°C. The experimental data of Figel and Fencl’s model of plasma acid-base balance was combined with the data based on Siggaard-Andersen nomogram, corrected to 37°C. It was obtained a model of blood acid-base balance combining the plasma model with variable albumin, globulin and phosphate concentrations and connected with the erythrocyte model. The model is a core of an extent model of acid-base balance which enables the realisation of pathogenesis of acid-base disturbances in compliance with the balance approach to the interpretation of ABB disturbances, published earlier (Kofránek et al., 2007).

Acknowledgements

The work was supported by the project of National Programme of Research No. 2C06031, “e-Golem”, the development project of Ministry of Education, Youth and Sports C20/2008 and by Creative Connections s.r.o. company.

References


Border flux balance approach towards modelling acid-base chemistry and blood gases transport

Kofránek, Jiří; Matoušek, Stanislav and Andrlík, Michal

pp. TU-1-P7-4: 1-9
BORDER FLUX BALLANCE APPROACH TOWARDS MODELLING ACID-BASE CHEMISTRY AND BLOOD GASES TRANSPORT

Jiří Kofránek, Stanislav Matoušek, Michal Andrlík

Charles University in Prague, First Faculty of Medicine, Institute of Pathological Physiology, U Nemocnice 5, Praha, Czech Republic

kofranek@email.cz (Jiří Kofránek)

Abstract

Two widely discussed approaches to acid-base chemistry (classical vs. modern one) are not in contradiction with one another. Most of the misunderstanding originates when people don’t realize they are defined for different conditions – classical theory is valid for blood of various hemoglobin content but with normal plasma only while the modern theory describes plasma under general conditions, but not full blood. The two theories are complementary when both limiting conditions are met. Our border flux approach to acid-base and blood gases chemistry, together with the newly defined set of state variables form a logically consistent general theory. We are able to describe, identify and simulate blood gases transport as well as a wide array of acid-base disorders, e.g. respiratory alkaloses and acidoses, metabolic acidoses and alkaloses, hypo- and hyperalbuminemias (phosphatemias), hemodilution/ hyperhydration, hemoconcentration/dehydration, situations of abnormal body temperature etc. Complex simulation model based on our approach (Golem) is for several years successfully used in medical student education. Underlying mathematical relationships are as a part of a MATLAB/Simulink library “Physiolibrary” at your free disposition on site http://patf-biokyb.lf1.cuni.cz/wiki/projekty/physiolibrary. Lately, we have completed on-line version of simple plasma buffering system simulation model, which is available at http://www.physiome.cz/atlas/acidobaze/02/ABR_v_plazme1_2.swf.

Keywords: Acid-base chemistry of blood, Modeling, SID, BB, BE

Presenting Author’s biography

Stanislav Matoušek, M.D. Author is a Ph.D. student of Jiří Kofránek, main focus of his work being modeling of ventilation, blood gases transport and acid-base chemistry in humans. He studied Medicine at Charles University in Prague. He combined his studies with several years of mathematics and physics at Charles University and with some subjects regarding control theory at Czech Technical University. Stanislav Matoušek is a member of multidisciplinary team of laboratory of biocybernetics lead by Jiří Kofránek. He also teaches Pathological Physiology to medical students.
1 Preface

Efforts to formalize the description of the blood acid-base chemistry are not at all new, going back to the pioneering times of Henderson (1909) and Hasselbalch (1916). Currently, many clinical and physiological calculations are based on the existing formalization, as those carried out in the analytical devices for the acid-base status assessment, analyzers of respiratory system function, ventilators of scuba-divers, automatic ventilator control in critical-care medicine etc.

There are three mechanisms to acid-base chemistry regulation: 1) buffers, 2) ventilation, 3) renal regulation of \([H^+]\) and \([HCO_3^-]\). Fastest regulation (in order of milliseconds) that prevents swift changes of pH after disturbance relies on buffering reactions. Reflex changes of ventilation rate regulate level of pCO\(_2\) in blood, i.e. one side of the bicarbonate buffer. Kidneys provide the slowest response to acid-base disturbances, regulating final excretion of H\(^+\) and regeneration of HCO\(_3^-\). Thus, the last two mechanisms act by influencing the blood buffering systems.

2 Classical approach of the “Danish school”

Let us use the symbol [Buf\(^-\)] for the cumulative concentration of non-bicarbonate buffer base (basic forms of plasma proteins, hemoglobin and phosphate). Sum of concentration of bicarbonates and non-bicarbonate buffer bases gives us so called Buffer Base (BB) [6]:

\[
BB = [HCO_3^-] + [Buf^-] \tag{1}
\]

Value of BB is, of course, mainly determined by pH. However, part of BB value is independent of pH, hemoglobin concentration (Hb) being the principal variable upon which it also depends. To be able to compare the results of two patients with various hemoglobin concentrations, Base Excess (BE) was defined as a difference between actual concentration of BB at the given pH and its normal value (normal buffer base - NBB – dependent of Hb) at pH 7.4. BB and NBB are dependent on Hb concentration, but due to the definition, this dependence is eliminated with BE.

\[
BE = BB - NBB \tag{2}
\]

During metabolic disturbances of acid-base chemistry (i.e. during disturbances caused by imbalance between production and excretion of strong acids), the excess or lack of strong acids is dampened by buffer systems, causing changes of both BB and BE.

In order to clinically assess of acid-base disorders, Danish authors lead by Ole Sigggaard-Andersen [5] have created nomograms based on experimental data. Nomograms graphically describe the relationship between the pH (or hydrogen ion concentration) and the value of pCO\(_2\), BE and Hb.

\[
pH = \text{function}(pCO_2, \ BE, \ Hb) \tag{3}
\]

These nomograms are still widely used in clinical practice (although today mainly hidden in electronic computation algorithms inside the blood-gas/pH analyzers). However, they were constructed for the blood with normal concentration of plasma proteins. That can cause interpretation and/or validity problems in patients with plasma protein concentration disorders - rather common situation in acutely ill-patients.

3 “Modern” approach of Stewart and his followers

Limitation of Siggaard-Andersen nomogram to normal levels of plasma proteins (or more exactly normal values of plasmatic buffers generally) has lead Peter Stewart into formulation of a new approach to acid-base chemistry of plasma. Stewart’s calculations [8] are based on five notions that must hold and are expressed by following equations.

1. Water dissociation equation:

\[
[H^+] \ [OH^-] = K'_w \tag{4}
\]

2. Law of conservation of mass for the buffers in their two forms (basic and acidic):

\[
[\text{Buf}^+] + [\text{HBuf}] = [\text{Buf}_{\text{TOT}}] \tag{5}
\]

3. Dissociation equilibrium of the non-bicarbonate buffer system:

\[
[\text{Buf}^-] \ [H^+] = K_{\text{BUF}} \times [\text{HBuf}] \tag{6}
\]

4. Dissociation equilibrium of the bicarbonate buffer:

\[
[H^+] \ [HCO_3^-] = M \times pCO_2 \tag{7}
\]

5. Dissociation equilibrium between bicarbonate and carbonate:

\[
[H^+] \ [CO_3^{2-}] = N \times [HCO_3^-] \tag{8}
\]

6. Principle of electroneutrality:

\[
\text{SID} + [H^+] - [HCO_3^-] - [\text{Buf}^-] - [CO_3^{2-}] - [OH^-] = 0 \tag{9}
\]

SID in the last equation is an acronym for “strong ion difference”, defined as the difference between the concentration of the fully dissociated cations and fully dissociated anions (expressed in mEq/L). In practice, we usually find its value as:

\[
\text{SID} = [\text{Na}^+] + [\text{K}^+] + [\text{Mg}^{2+}] + [\text{Ca}^{2+}] - [\text{Cl}^-] \tag{10}
\]

By combining the six given equations, we can obtain a single algebraic equation of fourth order, which can be solved for the concentration of hydrogen ions (independent variables are in bold; constants in italics):
In other words, pH can be expressed from this equation as a function of three independent variables; pCO₂, SID and total concentration of non-bicarbonate buffers in both forms [BufTot]:

\[
\text{pH} = \text{function ( pCO}_2, \text{SID, [BufTot])} \quad (12)
\]

Clearly, pCO₂ can be viewed as a measure of respiratory disturbance, while changes in SID and [BufTot] reflect metabolic disturbances.

However, many of Stewart’s followers [1, 7] look at this formula as if it were an oracle – and incorrect causal relationships are deduced from factually correct mathematic formulas (fig.1). For instance, application of the above mentioned formula gives you tendency to alkalemia in cases of hypoalbuminemia (characterized by decrease in BufTot). Dilution of extracellular fluid (followed by a decrease of SID) gives tendency to alkalemia and concentration of extracellular fluid gives tendency to acidemia. In such a way one can explain the so called dilution acidosis in cases of saline solution infusions or so called contraction alkalosis observed in some cases of hypovolemia.

Some authors erroneously deduce from these facts that one of the original causes of acid-base disorders is the change in SID. For instance, Sirker et al. [7] states that:

“One implication which is stressed in Stewart’s original study is that movement of hydrogen ions between solutions (by ion channels or pumps) will not actually affect local hydrogen ion concentration. If one considers a membrane separating two body fluid compartments, for each compartment the value of \([H^+]\) depends solely on the value of the independent variables in that compartment. Directly adding and removing of \(H^+\) to or from one of the compartments will not alter the value of any of the independent variables present and hence \([H^+]\) will be maintained at the same value as previously by change in the dissociation of water to reverse any \([H^+]\) fluctuations. The water dissociation equilibrium is able to provide an essentially inexhaustible source or sink for \(H^+\) ions.”

However, regulation of the pH in the bodily fluids is achieved by adjusting the outflow of two entities to balance their production – kidneys regulate the excretion of strong acids while lungs regulate the outflow of carbon dioxide. The two regulated entities are connected by buffer systems. We can divide them into bicarbonate buffer system and non-bicarbonate buffer system (plasmatic proteins, hemoglobin and phosphates make part of this one).

![Fig. 1 Modeling acid-base chemistry according to Stewart’s theory](image1)

![Fig. 2 The value of plasmatic SID is basically identical with the BB, concentration of hydrogen ions in model is either calculated from the Stewart’s SID or from the BB and concentration of plasma albumins](image2)
4 Border flux balance approach to acid-base chemistry

In order to understand well the causal relationships in acid-base disorders, it is crucial to realize that values of SID and BB are basically identical (fig. 2).

Thus, pH can be calculated using the SID as well as the BB. Changes of SID and changes of BB (or BE) are identical, because the movements of ions across plasmatic membranes are electroneutral. Hydrogen ions or bicarbonates are always accompanied by other ions when they move from one compartment to another.

For instance, during metabolic production of strong acids, the amount of hydrogen ions that leaves intracellular fluid is accompanied by equivalent amount of the strong acid anions. Analogically, influx or loss of bicarbonates is always accompanied by equimolar transfers of other ions. Loss of bicarbonates by gastrointestinal tract is matched by retention of chloride ions in extracellular fluid, leading to hyperchloremic metabolic acidosis. Buffering reaction do not change electroneutrality, however, they can change the value of SID and BB. Transferred hydrogen ions or bicarbonates instantly react with buffering systems and cause an identical change in BB as was the change of SID caused by transport. Changes of the concentrations of ions comprising the SID and changes of the entities describing buffering capacity (BB or BE) are dual representations of one single phenomenon.

Our experience with the modeling of acid-base chemistry has helped us to formulate a theory that we have named border flux balance approach.

Our models are based on several principles:

Nor the buffering reactions nor the transfer of ions between various compartments of bodily fluids do change electroneutrality. Fluxes of hydrogen ions or bicarbonates are always accompanied by the fluxes of complementary ions. That is why the changes of the concentrations of ions that influence the SID and changes of the entities describing the buffering capacity (BB or BE) are dual representations of one single acid base status.

It is beneficial to work with the total content of the given substance rather than the measures of the various forms in which the substance is present in the sample (e.g. O2 in blood exists in the free form and in the form that is bound to hemoglobin, CO2 is in the free dissolved form, bound to hemoglobin and in the form of bicarbonate). The transitions between various
forms do not change the total amount of the given substance in the sample; the total content can change only by a flux of the substance across the imaginary borders of the sample. Commonly used variables are obtained by backward conversion.

Although its use in clinical practice can without doubt be of advantage, Stewart’s theory is not suitable for mathematical modeling. In order to use it, we would have to model homeostasis of all the ions contained in the definition of SID. That is why, where possible, we rather use the terms coined by Siggaard-Andersen. In the cases where the older definition is rather blurred, we precise it in such a way that ensures logical consistency. For instance, we precise the term BE of Siggaard-Andersen under the name of standard base excess (StBE) as follows: StBE is the amount of hydrogen ions that has to be added to 1 liter of the sample during the titration with a strong acid to reach pH of 7.4, all that with the given hemoglobin concentration and plasmatic buffer concentration (albumins and phosphates mainly) of the sample; during titration pCO2 is set to be 40 mmHg, temperature to be 37°C and hemoglobin saturation to be 100%. To summarize it, the difference between the BE and StBE is that BE is a function of pH, hemoglobin concentration and plasmatic buffer concentration. This definition is logically consistent.

The behaviour of blood under general conditions (abnormal temperature, low hemoglobin saturation, anemia, hypo- or hyperalbuminémie) can be formally described by a set of equations. In case of modeling blood gases and acid-base chemistry, seven degrees of freedom are enough to achieve a sufficient accuracy. We chose seven mutually independent state variables as follows:

- $cO_2_{Tot}$ – total concentration of O2 in all its forms;
- $cCO_2_{Tot}$ – total concentration of CO2 in all its forms;
- StBE – standard BE – generalized BE, as defined above;
- Buf$_{Tot}$ = cProt$_{Tot}$ + cPhosp – total concentration of plasmatic buffers, that is total concentration of plasmatic proteins (or to be more precise concentration of their buffering aminoacids; these are found mostly on albumin molecules) added with the concentration of plasma phosphates;
- $cHb$ – hemoglobin concentration;
- temperature – actual temperature (norm 37°C);
- $K^+$, $H^+$, $Cl^-$, $Na^+$, $HCO_3^-$, $H_2CO_3$ – ions of the comlimentary ions.
total volume of blood or alternatively dilution factor. It is not needed to describe uniquely a blood sample. However, if we want to simulate dilution acidosis and/or contraction alcalosis in conditions of whole organism, we need it.

Inflow/outflow of a blood gas leads to change in total concentrations of a blood gas, inflow/outflow of hydrogen ions (or alternatively hydroxyl ions) and bicarbonates lead to a change in StBE (adding 1 mmol of hydrogen ions to 1 liter of blood reduces the StBE by 1 mmol/L, adding 1 mmol of bicarbonates will lead to increase of StBE by 1 mmol/L.

This is why we are able to model the (respiratory) balance between the inflow/outflow of CO2 and the (metabolic) balance between the production and kidney excretion of strong acids. We are also able to calculate the changes of the state variables during hemoconcentration and hemodilution (thus, we obtain the pH at dilution acidosis and contraction alcalosis), as well as the changes of acid-base parameters at hypoalbuminemia or hypothermia.

The border flux balance approach to acid-base chemistry unifies the classical approach of the “Danish school” with the quantitative approach of Stewart and his followers. The causal chain in our approach goes from two regulated balances (that of influx/outflux of CO2 and that of influx/outflux of strong acids (fig. 3) to the buffer systems that connect them and onto the pH which is the result of these processes (fig. 4). Quantitative representation of our
5 Learning while playing – an acid-base chemistry simulation model as an e-learning application

The combination of multimedia interactive environment with simulation games is a very effective pedagogic tool, making it easier for a student to understand the complex dynamics of physiological regulation and pathophysiological disorders of it. It is possible to play with the system while the regulatory feedback loops are disconnected. Hence, students come to understand better the difference between the behaviour of the system itself and the influence of the feedback regulation.

The simulation model based on the border flux balance approach makes a base for our teaching multimedia application that includes buffering systems, regulatory influence of respiration and kidneys and influence of ionic and volume homeostasis on acid-base status. For instance, student can try to simulate the evolution (causal chain of events) of dilution acidosis in subsystem of plasma.
buffers (fig.5); he/she starts with normal plasma (A), first he/she dilutes the plasma (B), then finds out what happens after buffering equilibration (C) and finally applies the respiratory regulation (D).

A complex model of acid-base chemistry is embedded into our educational simulator named Golem [8]. The underlying mathematical model with all the equations is as a part of a MATLAB/Simulink library “Physiolibrary” freely available at internet site http://patf-biokyb.f1.cuni.cz/wiki/projekty/physiolibrary. Illustrations of graphical output of the educational simulator Golem are at figures…. Golem was made in the ControlWeb environment and is used as an effective means of education since 2001. We are currently working on a design of the new on-line version, where the simulation games would be integrated with a multimedia explication section. Thus, it would become part of the newly created internetAtlas of Physiology and Pathophysiology (referred about in another article of the Eurosim conference)

6 Conclusion

Understanding of pathophysiological causality of acid-base disorders gives insight especially into more complex disorders of internal environment, of which acid-base disorder may be part. It is a prerequisite of correct diagnosis and treatment. According to our experience, an educational simulation model can be an effective means of understanding true causal chain of acid-base disorders.
7 References


Acknowledgement
This research was supported by aid grant MŠMT č.2C06031 and BAJT servis s.r.o company.
Independence of Variables in Stewart's model of acid-base chemistry of the blood plasma

Matoušek, Stanislav; Kofránek, Jiří and Rees, Stephen E.

pp. 246-250
Independence of Variables in Stewart’s Model of the Acid-Base Chemistry of the Blood Plasma

Stanislav Matousek*, Jiri Kofranek**
Stephen E. Rees***

* First Faculty of Medicine, Charles University, Prague, 128 53 Czech Republic (Tel: +420-224965915; e-mail: stanislav.matousek@fjfi.cuni.cz).
** First Faculty of Medicine, Charles University, Prague, 128 53 Czech Republic (e-mail: kofranek@email.cz).
*** Department of Health Science and Technology, Aalborg University, Aalborg, Denmark, (e-mail: sr@hst.aau.dk)

Abstract: Several approaches have been taken to modelling of the acid-base chemistry of blood. The Stewart approach includes three independent variables (SID, PCO₂, A₀₂), which are postulated to completely describe causal mechanisms behind changes in acid-base status. This paper explores this postulate, simulating typical clinical examples using an online modeling tool. For changes in alveolar ventilation, production of strong acid, and selective removal of non-charged protein buffers, this postulate is true. However for non-selective protein buffer removal SID and A₀₂ cannot be seen as independent. The paper discusses the implication of this on diagnosis of acid-base disturbances in patients with abnormal protein concentration.

Keywords: acid base chemistry, computer simulation, computer experiment, equilibrium, model approximation, physiological models, physiology, state variables

1. INTRODUCTION

Several approaches have been taken to modelling of the acid-base chemistry of blood. Broadly, these can be classified into those of Siggaard-Andersen (1974, 1977) and colleagues and those seen as more ‘modern’, i.e. those of Stewart (1983) and other authors (Figge et al., 1991; Constable, 2000; Fencel et al. 2000). The Stewart approach has been seen by some authors as a revolution in ability to understand the true nature of an acid-base disturbance (Constable, 2003). This is largely due to the modelling formulation adopted by Stewart, where reaction equations are represented by mass balance and mass action equations, and 3 variables are selected as ‘independent variables’. These three independent variables are partial pressure of carbon dioxide pCO₂, total concentration of plasma weak acid Atot and strong ion difference SID. Their independence has meant two things: first, that values of these three variables can be used to solve all Stewart’s equations describing acid-base chemistry of plasma; and second, their independence has been ascribed to fact that they themselves do change as a result of different (independent) physiological mechanisms: Changes of alveolar ventilation affect the carbon dioxide level (pCO₂), disturbances of strong ion levels affect SID and changes of total plasma buffer level (mainly protein, especially albumin) affect A₀₂. These physiological mechanisms are seen as independent causes of acid-base disturbances.

This article explores the ability of Stewart’s three independent variables to describe some typical clinical disorders resulting in acid-base disturbances, i.e. changes in alveolar ventilation, the onset of anaerobic metabolism and the loss of plasma protein during for example acute nephrotic syndrome, burn trauma or sepsis. It illustrates situations where Stewart’s assumptions of independence are valid and more importantly, where they may be questioned.

2. STEWART’S MODEL OF ACID BASE CHEMISTRY OF PLASMA

Figure 1 illustrates Stewart’s model of the acid-base chemistry of plasma. It includes mass action equations describing reaction equilibria of the dissociation of water, the non-bicarbonate buffer, the bicarbonate buffer and the carbonate buffer, although it is usual to omit the equations representing water and carbonate (Constable, 1997). The model also includes mass balance equations keeping track of the total non-bicarbonate buffer (weak acid), and a single equation describing electrical neutrality in the plasma. The sum of strong cations minus the sum of strong anions – so called strong ion difference – has to equal the negative charge of the buffers. Since the buffers in Stewart’s model are only negatively charged in their base form (A⁻ and bicarbonate), this means that SID equals concentration of A⁻ and of bicarbonate.

$$\text{SID} = \text{HCO}_3^- + A^- \quad (1)$$
Equation 1 is practically equal to the line 6 of figure 1 (the original Stewart model), since all the omitted terms are negligible and do not change the numerical solution (Constable, 1997). The model assumes a single non-bicarbonate buffer (equation) lumping all protein buffers and phosphate in the plasma.

Fig. 1

<table>
<thead>
<tr>
<th>Reaction Equations</th>
<th>Mathematical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{H}^+ + \text{OH}^- \rightleftharpoons \text{H}_2\text{O}$</td>
<td>$[\text{H}^+]:[\text{OH}^-] = K_w$</td>
</tr>
<tr>
<td>$\text{H}^+ + \text{A}^- \rightleftharpoons \text{HA}$</td>
<td>$[\text{H}^+]:[\text{A}^-] = K_a : [\text{HA}]$</td>
</tr>
<tr>
<td>$\text{H}^+ + \text{HCO}_3^- \rightleftharpoons \text{H}_2\text{O} + \text{CO}_2$</td>
<td>$[\text{H}^+]:[\text{HCO}_3^-] = K_c : \text{pCO}_2$</td>
</tr>
<tr>
<td>$\text{H}^+ + \text{CO}_3^{2-} \rightleftharpoons \text{HCO}_3^-$</td>
<td>$[\text{H}^+]:[\text{CO}_3^{2-}] = K_c : [\text{HCO}_3^-]$</td>
</tr>
<tr>
<td>$[\text{SID}]:[\text{H}^+]:[\text{HCO}_3^-]:[\text{A}^-]:[\text{CO}_3^{2-}]:[\text{OH}^-] = 0$</td>
<td></td>
</tr>
</tbody>
</table>

Equations playing a significant role are shown in black. In the electroneutrality equation, terms that are several orders lower in magnitude than the rest can be omitted, and are shown in gray. Equation constants have the following values: $K_w = 4.46 \times 10^{-14}$ (eq/L)$^2$, $K_a = 3 \times 10^{-7}$ (eq/L), $K_c = 2.46 \times 10^{-11}$ (eq/L)$^2$/mmHg, $K_i = 6 \times 10^{-11}$ Eq/L.

Values of the independent variables are $\text{pCO}_2 = 5.3$ kPa, SID = 41.7 meq/l and $A_{sat}$ = 19 meq/l in normal conditions (Stewart, 1983). Solving Stewart’s equations (lines 1–6 of figure 1) for these values gives the values of the dependent variables: $\text{pH} = 7.40$; $\text{A}^- = 16.8$ meq/l; $\text{HA} = 2.2$ meq/l; $\text{HCO}_3^- = 24.9$ meq/L, and $\text{CO}_3^{2-} = 0.04$ meq/L.

One of Stewart’s state variables is $\text{pCO}_2$, rather than total $\text{CO}_2$ (in form of $\text{CO}_2$, $\text{HCO}_3^-$ and other). This variable is selected because the buffering system of bicarbonate behaves as an open system in the body, equilibrating back to a given $\text{pCO}_2$ after any buffering has occurred.

3. INTERPRETATION OF CLINICAL SITUATIONS USING STEWART’S MODEL

This section describes the interpretation of physiological disorders using Stewart’s model, these being changes in ventilation, the onset of anaerobic metabolism and the loss of plasma protein in two distinct ways during for example nephrotic syndrome, burn trauma or sepsis. These are explained both in terms of the reaction equations included in figure 1, and exemplified using an implementation of these reactions. This implementation is part of a broader atlas of physiological models available online to the reader at www.physio.me/cz/atlas/acidobaze/04/index.htm

3.1 Changes in Alveolar Ventilation

Changes in the acid-base status of plasma due to hyperventilation can be seen in figure 2a, with the direction in these changes shown by the bold arrows. Increased ventilation reduces $\text{CO}_2$ levels (black arrow), which in turn causes reaction 1 to proceed to the right, bicarbonate and hydrogen ions reacting in equimolar amounts (gray arrows). Physiological $\text{H}^+$ concentrations are in the nanomolar range meaning that the shift in reaction equation 1 to the right must be compensated by an equivalent shift in reaction equation 2 to the left (empty arrow). The result is that the sum of $\text{A}^-$ and $\text{HCO}_3^-$ is constant at the level of millimoles, i.e. SID stays constant. In addition the proportions of $\text{A}^-$ relative to HA have changed but the total concentration $A_{sat}$ remains constant. Changes in ventilation are therefore well described by Stewart’s 3 independent variables: a changing $\text{pCO}_2$, with constant SID and $A_{sat}$.

Fig. 2a

![Image](247)

Fig. 2b

![Image](247)

Fig. 2. Respiratory alkalosis: A typical situation is simulated whereby $\text{pCO}_2$ changes from normal level of 5.3 to 2.7 kPa. Numbers represent before and after situations, and are given in meq/L, except for $\text{H}^+$ (neq/L), and $\text{CO}_2$ (kPa). Arrows represent the direction of change with black arrow representing $\text{CO}_2$ disturbance, gray arrows the reaction of the bicarbonate system and empty arrows the reactions of non-bicarbonate buffers. Resulting $\text{pH}$ corresponds to the after concentration of $\text{H}^+$.

Figure 2b shows situation after $\text{pCO}_2$ has approximately halved due to increased alveolar ventilation. Bicarbonate decreases and $\text{A}^-$ increases by 1.0 meq/l, the increase in $\text{A}^-$ being fuelled by an equivalent decrease in HA. Values of both SID and $A_{sat}$ are represented graphically in the figure: SID is the sum of the cups containing $\text{HCO}_3^-$ and $\text{A}^-$, while $A_{sat}$ is the sum of cups containing $\text{A}^-$ and HA.
3.2 Metabolic Production of Lactic Acid

Changes in the acid-base status of plasma due to increases in strong acid can be seen in figure 3a. Dissociation of strong acid produces H+ which is bound to either HCO₃⁻ or A⁻ causing both reactions to move to the right, maintaining H⁺ concentrations in the nanomolar range. The net reduction in the sum of HCO₃⁻ and A⁻ (i.e. SID) is equivalent to the amount of H⁺ added from the metabolism (at the milimolar level). Assuming adequate ventilation, the increased CO₂ is removed, maintaining normal pCO₂ levels. The proportions of A⁻ relative to HA have changed, but the total concentration Aₜot remains constant. Adding strong acid produced in the metabolism is therefore well described by Stewart’s 3 independent variables: a changing SID, with constant pCO₂ and Aₜot.

Fig. 3a. Metabolic acidosis: A typical situation where 10 meq/L of strong acid is added and SID changes from normal level of 41.7 to 31.7 meq/L. All the numbers are in meq/L, except for H⁺ (neq/L), and CO₂ (kPa). Black arrow represents original disturbance, gray arrows reactions of the bicarbonate system and empty arrows reactions of non-bicarbonate buffers. Resulting pH corresponds to the after concentration of H⁺.

Stewart’s model describes a decrease in Atot with constant SID as a hypoproteinemic alkalosis. Such a situation arises when Atot is reduced by loss of the acidic form of protein HA only. One of the pathological states where this could be implied is nephrotic syndrome (Kurtz et al, 2008). In nephrotic syndrome, plasma albumin is filtered into urine and lost. The albumin loss can be described as a loss of HA, whenever pH of urine is acidic enough (around 5.0), because albumin is known to be in its acidic form in such a low pH environment. Selective removal of the HA component of Aₜot causes reaction 2 to proceed to the right, A⁻ and hydrogen ions reacting in equimolar amounts. The reaction replenishes lost HA from A⁻ at the cost of loss of hydrogen ion. Physiological H⁺ concentrations are in the nanomolar range meaning that the shift in reaction equation 2 to the right must be compensated by an equivalent shift in reaction equation 1 to the left – bicarbonate is formed from CO₂, with pCO₂ returning to its original value due to the open system. The result is the sum of A⁻ and HCO₃⁻ being constant at the level of milimoles, i.e. constant SID. In addition, the proportion of A⁻ relative to HA changes and Aₜot is reduced.

It should be noted that the commonness of loss of the HA form of the protein in nephrotic syndrome (or in any other disorder – see section 3.4) could be questioned. While there are examples of patients with nephrotic syndrome hypoproteinemia and metabolic alkalosis described in literature (McAuliffe et al., 1986), the most common pH seen in this group of patients is within physiological limits (Brenner, DuBose et al., 2004). This is in agreement with filtering of albumin into urine in both HA and A⁻ forms. A⁻ form is only converted into HA in renal distal tubuli and collecting ducts, where H⁺ ions are actively secreted into the tubular lumen and bind with the A⁻ to convert it into HA. Due the availability of albumin as an extra urinary buffer system, the luminal pH decreases slowly and the secretion of H⁺ tends to proceed in excess to the body’s need (Steinmetz et al., 1971; Steinmetz, 1986) and can thus cause a tendency to alkalosis. However, this secretion is soon readjusted, HA and A⁻ are lost in proportion (section 3.4), and pH returns to normal.

Primary changes in HA, whilst potentially uncommon, are therefore well described by Stewart’s 3 independent variables: a changing Aₜot with constant SID and pCO₂.

Fig. 3b shows situation after addition of 10 meq/L (=10,000,000 neq/L) of acid. Both bicarbonate and A⁻ decrease, the decrease in A⁻ causing an equivalent increase in HA. pCO₂ remains constant and pH changes accordingly to changes in HCO₃⁻, A⁻ and HA.

3.3 Protein changes with loss of the acidic form only
Fig. 4. Loss of the acidic form of protein: We simulate a situation of loss of 9.5 meq/L of HA, this loss actually exceeding the original concentration of HA. All values are in meq/L, except for H⁺ (neq/L), and CO₂ (kPa). Black arrows represent original disturbance, empty arrows reactions of non-bicarbonate buffers and gray arrows reactions of the bicarbonate system. Resulting pH corresponds to the after concentration of H⁺.

Figure 4 illustrates an example of HA loss. Aᵦ is reduced by 9.5, as seen by both reduction in A⁻ and HA. The reduction in A⁻ of 8.2 meq/L is equivalent to the increase in bicarbonate, giving constant SID. pCO₂ remains constant.

3.4 Protein changes with loss of both acid and base form

Disorders such as sepsis or burn trauma are associated with increased permeability of capillary membranes in damaged areas, causing both forms of protein buffers (HA and A⁻) to be lost. Changes in the acid-base status of plasma due to these disorders might therefore be represented as removal of the HA and A⁻ in the proportion seen in plasma, as illustrated in figure 5. Removal of HA and A⁻ in proportion causes no disturbance of the equilibrium of reaction 2, and as a consequence no change in the equilibrium of reaction 1. A⁻ is, however reduced, but at constant values of HCO₃⁻, meaning that the net concentration of A⁻ and HCO₃⁻, i.e. SID, is reduced.

It should be noted that patients with burns and sepsis often present with metabolic changes of pH ranging from acidosis to normal pH and even alkalosis. This has been ascribed to the production of metabolic acids like lactate due to the underlying disorders (section 3.2) as well as the infusion therapy. However, in both of these disorders and in most of the other causes of protein loss, the underlying mechanism (generally filtration) gives no reason to believe that HA is lost alone and not in the current proportion to A⁻, whatever the proportion of these forms in plasma is.

Changes in HA and A⁻ in proportion are therefore poorly described by Stewart’s 3 independent variables with changes of both Aᵦ and SID in conditions of constant pCO₂.

Figure 5 illustrates a typical situation of A⁻ and HA loss in proportion. Aᵦ decreases by 9.5 meq/L, as seen by both reduction in A⁻ and HA. The reduction in A⁻ of 8.4 meq/L gives an equimolar reduction in SID. pCO₂ remains constant.

Fig. 5. Loss of both forms of protein: We simulate a typical situation of loss of 9.5 meq/L of HA and A⁻ in proportion. All the numbers are in meq/L, except H⁺, (neq/L), and CO₂ (kPa). Black arrows represent original disturbance. No compensatory buffering reactions occur. Resulting pH corresponds to the after concentration of H⁺.

4. DISCUSSION

There has been a considerable debate as to whether the modern or the traditional approach to modeling the acid-base of plasma provides a better diagnosis of patients with acid-base disturbances (McAuliffe et al., 1986; Severinghaus, 1993; Siggaard-Andersen et al., 1995; Constable, 2000; Fencel et al., 2000; Dubin et. al, 2007, Kurtz et al., 2008). One of the important improvements proposed by Stewart was the representation of three ‘independent variables’ (SID, Aᵦ
and pCO₂), values of which enable unique solution of all Stewart’s equations describing acid-base chemistry of plasma. These variables have been postulated as an improvement in helping to distinguish between different causal mechanisms of changes in acid-base chemistry, such that if values of SID, A\\text{tot} and pCO₂ are known then the complete causal mechanisms behind changes in acid-base status can be understood. This postulate relies upon the assumption that each independent variable is affected by different causal mechanisms. If the same causal effect changes more than one of the independent variables then the independent variables are no longer capable of complete separation of all causal mechanisms. This paper explores this postulate, simulating typical clinical examples and illustrating independence between the variables, or otherwise, in these examples.

In most of the examples described here the assumption of independence between the variables is true. These examples are changes in alveolar ventilation, production of strong acid, and changes in protein concentration in physiological situation where the non-charged part of protein buffers (HA) is selectively removed. However this is not always true, and in the situation where protein buffers are non-selectively removed, both SID and A\\text{tot} values change as a result without the need for a change in pH.

Lack of independence between these variables can lead to misdiagnosis. For example, some authors applying Stewart’s model (Fencel et al., 2000) claim that patients with a low SID and low A\\text{tot} should be interpreted as having a low SID acidosis due to a primary electrolyte problem, combined with hypoalbuminemic alkalosis (low A\\text{tot}) due to protein loss. These authors postulate that the combined effects of the acidosis and alkalosis lead to a normal value of pH. It can be seen from this analysis presented in this paper that this situation can be equally well described by loss of protein such that A’ and HA forms are lost in proportion. In this context, the electrolyte disturbance is secondary to the effects of A’ change on protein loss and A\\text{tot} and SID are not independent. Indeed, Siggaard-Andersen and colleagues have previously criticized the concept of alkalosis or acidosis as being caused by changes in total concentration of protein buffers, i.e. albumin, or in the case of whole blood, haemoglobin (1995).

ACKNOWLEDGEMENTS

This work was supported by the project MSM 0021620806 of the Ministry of Education, Youth and Sports of the Czech Republic and by aid grant MSMT 2C06031 (Creative Connections s.r.o. company). It was also partially supported by the Programme Commission on Neuroscience, Biotechnology and IT under Danish Council for Strategic Research. The authors would like to thank Mr. Vojtech Rejl for the help with the graphics of the figures.

REFERENCES


Studies in Health Technology and Informatics., vol. 84, 2001

MEDINFO 2001


GOLEM - Multimedia simulator for medical education

Kofránek, Jiří; Lu Danh Vu; Snášelová, Hana; Kerekeš, Roman; and Velan, Tomáš

pp. 1042-1046
GOLEM - Multimedia Simulator for Medical Education

Jiří Kofránek\textsuperscript{a}, Luu Danh Anh Vu\textsuperscript{a}, Hana Snášelová\textsuperscript{a}, Roman Kerekeš\textsuperscript{a}, Tomáš Velan\textsuperscript{b}

\textsuperscript{a} Department of Pathological Physiology, 1\textsuperscript{st} Medical Faculty, Charles University, Prague, Czech Republic
\textsuperscript{b} Department of Laboratory Medicine, University of Washington Medical School, Seattle, USA

Abstract
We created multimedia medical training simulator “GOLEM” for learning diagnostics and therapy of the critical clinical disorders. The theoretical basis of the simulator is the mathematical formulation of the relationship of homeostasis of the internal environment (acid/base and electrolyte equilibrium, of transport of blood gases, of osmotic and volume homeostasis), respiration, circulation and kidneys including regulatory influence of relevant hormones and the influence of some therapeutic procedures. Mathematical description consists of 39 non-linear differential equations and containing 89 input and 179 output variables. For the development of the simulation models developer’s tools from MathWorks (Matlab and Simulink) has been used. The integration of the multimedia components, hypertext and simulation models interface was achieved by using Control Web, developed by Moravian Instruments, originally designed for long distance controls using PC and Internet. We have used our simulator as an efficient educational tool to help medical students learn circulatory, respiratory, acid-base, electrolyte, osmotic and volume disorders and train the diagnostic and therapeutic decisions by executing simulated interventions on virtual “patients”.

Keywords:
Models Educational, Computer Simulation, Hypermedia, Acid-Base Equilibrium, Water-Electrolyte Balance.

Aim of the project
The author’s intention was to create the multimedia medical simulator as an interactive teaching tool to help medical students learn acid-base, electrolyte, osmotic and volume disorders, respiratory and circulatory insufficiency. Our ambition was to allow students training in diagnostic and therapeutic decision-making process by executing simulated interventions on virtual “patients”.

Materials and Methods
Two different, but consequential tasks must be solved when developing a medical simulator

1. The development of the simulation model of physiological functions – the theoretical work itself, based on formalization of physiological relations.

2. The development of the simulator of physiological functions (the working title "GOLEM" is used) and the use of the simulator in education – practical application of the theoretical results. In addition, we have always supported the Internet accessibility of the project results.

Each of the problems is specific and requires using different developer’s tools.

Simulation model design with simulation chips
Whereas developing an educational simulator is a programmer’s work, formalization of the physiological relations is rather than creating software complicated scientific problem. To solve that effectively adequate tools are needed. Luckily for us, in the last few years substantial progress has been made in the field of software development tools for the design and testing of simulation models and we could utilize the advantage of professional tools by Mathworks (Matlab, Simulink and related libraries). These tools allow creating simulation models in an easy way illustrating their structure graphically as linked simulation circuits associated into “simulation chips”. In the same way as in electrical circuits electrical current is conveyed, in simulation chips information about the values of the variables is carried. The behavior of the “software chips” can be easily tested with the help of adequate software tools for computer simulation: similarly to an electric scheme, where it is possible to carry current on individual “pins” of the chip, the same manner can be used to carry selected value (or some time course of values) of an appropriate variable to the software chip. At the same time “measuring displays” can be “connected” to the outputs to register the values as long as the simulation is running.

Importantly, schematic representations using simulation chips are understandable for physiologists. It seems therefore that expression of the simulation models of physiological systems with the interlinked simulation chips could be a suitable way of standardization.

Simulator design with industrial developed tool
When generating a simulator we face a rather difficult task.
On one hand, we need to create a comprehensive graphical interface (with the integration of interactive animations, charts etc.). On the other hand, the structural complexity of the simulation model (which is the heart of the simulator) has high demands on the numerical performance. The requirements for sophisticated graphics and the demands on the high numerical performance of the running complex simulation model are contradictory – both are demanding on the processor performance.

When searching for the acceptable developer’s software tools for the development of the simulator we have looked in a field, where analogous contradictory requirements can be met. We have found one such a field and that is the generation of controlling and measuring applications for the industry. Here, similarly, the primary goal is to maintain high numerical output for the control and measuring application (which cannot be interrupted by drawing the graphics) even with the requirements for the graphical level of the user interface. After a certain time of searching, we have chosen the Control Web developer’s environment by Moravian Instruments.

When incorporating the simulation model into the core of the generated simulator the following trick was used: instead of the measuring/controlling adapter driver by which Control Web communicates with an industrial device in an industrial application, we have programmed a virtual driver for an (nonexistent) measuring/controlling adapter that was based on the simulation model. The Control Web application now “thinks” that it sends some controlling signals through the driver to an industrial device, but in reality the signals are inputs into the simulation model. Similarly, instead of measured output signals from the periphery, the Control Web application registers the values of the output variables of the model. This allowed us to employ the abundant options of the Control Web developer’s environment for the generation of our own simulator. The generation of the simulator is also facilitated by the options of the Mathworks special toolbox that allows automatic conversion of the graphical schemes from Simulink (the simulation circuits) into the source code of the program in C++.

Results

The theoretical concept of the simulator GOLEM is the large simulation model based on mathematical formalization of the blood gasses transfer, acid-base, electrolyte, volume and osmotic equilibrium, the function of the respiratory and urinary system, including the influence of hormones. The model design is based on classical models of Guyton et al. [3,4], Cameron [1], Ikeda et al. [5] and Coleman et al. [2], which we have sufficiently redesigned and extended. It consists of 39 non-linear differential equations and contains 89 input and 179 output variables.

The simulator GOLEM enables to visually demonstrate the mutual relations between the individual physiological subsystems and the manifestation of these relations in individual pathological conditions. The simulator enables the computer modeling of various pathological conditions and the influence of appropriate treatment. The simulator therefore becomes a visual learning aid for better understanding of the nature of physiological regulations and the manifestation of their malfunctions.

In connection to the development of distance studies via the Internet, we would also like to create an Internet version of the simulator, which would be accessible from an outside computer via a standard web browser. Simulator is drawn up so that its controlling would be as easy as possible.

The following simple example of an acid-base disorder should demonstrate how easy it is to manipulate the simulator. By changing the value for metabolic production of strong acids we can cause a metabolic acidosis in our “virtual patient” (fig. 1).

![Figure 1 – Turning the knob we can increase the metabolic production rate of non-volatile acids.](image1)

Ratio of metabolic production and renal excretion of strong acids is highly increased. H⁺Na⁺ and H⁺K⁺ exchange on the cell membrane is activated. Intracellular and extracellular buffers buffer H⁺ ions (fig. 2).

![Figure 2 – Acid-base balance in acute metabolic acidosis.](image2)

Our “virtual patient” is showing signs of metabolic acidosis - the blood has been acidified, Base Excess and actual bicarbonate concentration are decreasing, and pCO₂ is also slowly decreasing (fig. 3).
Acid-base values on compensatory diagram are in the acute metabolic acidosis range. This is the beginning of a progressive reaction by the ventilation center to counteract metabolic acidosis (fig 4). Respiratory compensation is at a maximum in about 12 hours. Decreasing pCO2 is leading to a rise of arterial blood pH. Acid-base parameters are approaching a compensated metabolic acidosis range (fig. 5). The slow response of the respiratory system on metabolic acidosis is due to the relative impermeability of bicarbonate across the blood-brain barrier (fig. 6). While CO2 penetrates easily and pCO2 in blood and cerebrospinal fluid is at a similar level, this is not the case for bicarbonate. Thus bicarbonate reaches equilibrium, cerebrospinal pH decreases, and the respiratory center is
more stimulated, and alveolar ventilation increases resulting in a decrease of pCO₂. Values of acid-base parameters are slowly approaching to compensated metabolic acidosis range. The kidney's response progressively develops. Titratable acidity and NH₄ excretion increases, urine pH decreases. Renal response is in its maximum in 3-5 days (fig. 7).

The simulator allows “virtual therapy” so that we can start an alkaline infusion for metabolic acidosis. To help the strong acid input/output balance, we can start bicarbonate infusions by simply adjusting the appropriate rate of bicarbonate intake in the simulator and alkaline infusion therapy of metabolic acidosis of our “virtual patient” is initiated (turning the knob we are starting bicarbonate infusion therapy - see figure 7).

BE and pH slowly increases after bicarbonate infusion. pCO₂ remains stable for a while (thanks to respiratory compensation), at its low level. We must take pCO₂ into account when choosing doses of alkaline infusion in order not to overdose (fig. 8).

In virtual therapy we can allow ourselves to make a mistake such as overdosing the alkaline infusion which in real life can be dangerous. If we overdose the infusion (fig. 9), we correct BE, but hyperventilation leads the patient from acidemia to alkalemia, which can be dangerous.

---

**Figure 9** – Alkaline infusion has been overdosed.

During acidemia (see figure 2) the cell membrane exchanges potassium ions for hydrogen ions entering into cells to be used by buffers. If acidosis lasts too long, the supply of potassium in the cells decreases, resulting in the depletion of K⁺. Inadequate therapy would quickly lead the patient from acidemia to alkalemia, as the cell exchanges K⁺ for H⁺ (from the intracellular buffers). Because extracellular stores of potassium are limited, its plasma concentration will quickly and dangerously decrease. (see figure 10).

It is necessary to replace lost potassium. We can then try correcting it, or simply pressing “Restart” button to try the all simulation run again. To correct the K⁺ depletion, we must use a potassium infusion in glucose and insulin - insulin takes glucose into cells but also increases the entry of potassium into cells resulting in a faster correction of K⁺ depletion (fig. 11). The infusion must not contain too large concentrations of potassium (as this would increase K⁺ to dangerous level).

---

**Figure 10** – Potassium concentration decreases dangerously.

---

**Figure 11** – To correct K⁺ depletion, a potassium infusion in glucose and insulin is used: - insulin takes glucose into cells but also increases the entry of potassium into cells resulting in a faster correction of K⁺ depletion.

A mistake is no reason to get upset since the patient is virtual (and so his death is just virtual). A real patient would not take our carelessness so easily, however. The simulation can be stopped at any time by using the Stop “switch” on the simulation clock and we can then take our time to analyse the many variables in the individuals physiologic subsystems windows.

**Discussion**

The simulators bring entirely new possibilities into the education of physicians. Not only they allow the students to try various therapeutic interventions (and to return to the starting point if necessary) without any threat for the patient, but also they are helpful teaching tools for gradual explanation of complex physiologic regulatory relations.

We can disconnect individual physiological subsystems (only on computer, not in reality, obviously) from their regulatory loops and observe their actions separately. In the “simulation game” with the individual subsystems, the
student can better understand the importance of the individual physiological relations and their involvement in all sorts of pathological conditions. That is why we have enabled the disconnection of the regulatory loops of each subsystem in our simulator of physiological functions GOLEM so that their behavior can be observed separately.

With the help of the simulation games with progressively interconnected subsystems, the student acquires a dynamic perspective of the relatively complex problems of the homeostasis of the internal environment, which is greatly important in the better understanding of the dynamical mutual relations between the regulations of acid-base, ion, osmotic and volume homeostasis. For example, the student can step by step observe, how high aldosterone level can causes metabolic alkalosis (fig. 12).

**Conclusion**

We have tested our simulator and multimedia computer aided learning tool in educational practice of our faculty. It seems that *multimedia combined with the simulation* are becoming an *efficient teaching tool* to help medical students in learning complicated pathophysiological disorders. Due to the interactive features of the simulation model user interface, it enables a *better perception* of the complicated dynamic relationships within complex physiological structures.

**Acknowledgments**

This research was partially supported by the Charles University Grant No. 217/1998, by the Research and Development Initiative MSM-111000008 (physiome.cz) and by BAJT servis Ltd.

**References**


**Address for correspondence**

Dr. Jiří Kofránek, Institute of Pathological Physiology, U Nemocnice 5, 128 53 Praha 2, Prague, Czech Republic.

E-Mail: kofranek@cesnet.cz
Modelling and Simulation 2002 Proceedings of 16th European Simulation Multiconference

edited by Amborski, K. and Meuth, H.
Darmstadt, SCS Publishing House, 2002

From Simulation chips to biomedical simulator

Kofránek, Jiří; Andrlík, Michal; Kripner, Tomáš and Mašek, Jan

pp. 431-436
Abstract

Computer simulation in physiology is an interdisciplinary problem. It is very important to provide good readability of used models. Programmer understands the architecture of models and the algorithms of programs. On the other hand physiologist usually can judge how closely the behavior of system correspond to the nature. In this text we will show how some known programming methods can help us find common language between programmers and physicians.

Simulation models

Simulation models in physiology are based on differential equations, regulation mechanisms and often on some mathematical algorithms like finding value of inverse function or solving some optimization problem.

In scientific literature there are many articles about partial simulation models programmed in classical sequential languages like FORTRAN, MODULA, ADA, etc. Those products are hardly understandable for physiologists and non-programmers. Speak nothing of linear structure of these programs, which usually excludes any updating, adaptations or expansion.

For these reasons our team found another way. We use MATLAB & SIMULINK graphical programming environment. This software is developed for use in control industry and for using advanced mathematical methods and has a strong toolset for constructing, developing and testing of various models.

Simulation chips

Step of high importance is creating of simulation chips. The idea borrows from the concept of electronic chips and electronic circuits where the circuitry lies hidden inside the packages. Simulation chip is a subsystem with exactly defined function. Mask of this subsystem contains brief description of function and meaning, and description of inputs and outputs.

This is actually a method of decomposition of a model to the parts.

It is clear that physiologists can look at this chip like a black box with some physiological function and programmers look at chip like a subprogram or algorithm that is part of a more complex system.

Example of one very simple chip can be seen on figure 1.

Simulation chip is a basic construction element. As physiological systems (and models too) may have several levels, it is natural that chips can be ordered hierarchically. It means,
system with several chips inside can be masked and can be represented as one “bigger” simulation chip. So construction of chips fully traces hierarchical organization of nature.

Example of hierarchical structure of chips can be seen on figure 2.

Figure 1: The simple example of simulation chip.

Figure 2: The sketch of hierarchical structure of chips.
**Chips library**

We have many different models that sometimes contain identical parts. Furthermore one system or chip can internally include the same chip in several places and contexts. From whence it follows that one simulation chip can be copied into many places in many different models. When we want to update the content of the chip, we must replace chip in all of its places (and it is very probable to forget about some of them).

Next, if we update or modify a chip, we don’t know if somewhere else there exists a copy of this chip. Very simply it may happen that we will have several different versions of a subsystem in several places. It very often leads to disorder.

Therefore our team applied the idea of a large library. It is possible to create libraries of blocks (in our case libraries of chips) in SIMULINK. A model then contains no chips but just links to chips into a library. All the same, user can open chips and pass through the model without any limitations.

Let’s now consider several models containing links to the same chip in a library. When we update or replace the chip in the library, then by opening or executing “Model Update” command model automatically reloads all links from libraries. So we have the latest versions of chips in a model all the time. Certainly a block in a library can contain a link to another chip, which is in the same or another library.

On opening a library it refreshes all links inside and loads the latest version of blocks. If we need to make a mutation of a linked chip in a single system only, and we don’t want to modify the chip in the library, it is possible to disable or break this link. It can be used for developing new model or developing any special modification of chip.

Example of a library is on figure 3.

*Figure 3: The library is a set of standalone chips.*
Our team used this way to develop “GOLEM” simulator, for simulation of inner body environment. See more on www.physiome.cz.

**Simulators**

One thing is to develop a simulation model and second thing is to build an interface between the model and a user. We must consider users who don’t know how to work in SIMULINK or don’t want to buy all MATLAB & SIMULINK products. Or the user has no interest in the structure of the model and chips but he needs to comfortably monitor the behavior of this model.

These observations lead our team to conclusion that it is necessary to be able to run the model outside the SIMULINK. After a certain time of searching, we have chosen the Control Web developer’s environment from Moravian Instruments (see http://www.mii.cz/).

On one hand, we need to create a comprehensive graphical interface (with the integration of interactive animations, charts etc.). On the other hand, the structural complexity of the simulation model (which is the heart of the simulator) has high demands on the numerical performance. The requirements for sophisticated graphics and the high execution speed of the simulation model are contradictory – both are demanding on the processor performance.

The tools for creating industrial controlling and measuring applications can handle this problem – these applications maintain real time computational throughput, while servicing input / output requests and updating graphics.

A little trick had to be used when incorporating the SIMULINK model into the core of the simulator. Instead of the measuring/controlling adapter driver by which Control Web communicates with a controlled device in an industrial application, we have programmed a virtual driver for a (nonexistent) measuring/controlling adapter. This virtual device driver mimics the behavior of the simulation model. The Control Web application now “thinks” that it sends some control signals through the driver to an industrial device, but in reality the signals are inputs to the simulation model. Similarly, instead of measured output signals from the periphery, the Control Web application registers the values of the output variables of the model (See figure 4). This enabled us to employ the abundant options of the Control Web developer’s environment for the generation of our own simulator.

The generation of the simulator is also facilitated by the capabilities of a special MATLAB toolbox - Real Time Workshop (RTW). RTW is intended for development of real-time systems running on many software and hardware platforms. As an intermediate step, RTW translates a SIMULINK model into C-language.

Through customization of the C source code, different targets are supported, depending on the programmer's needs. An important achievement of our team is a creation of a new RTWtarget, Control Web driver target, which serves the purpose of making a Control Web virtual device driver with an embedded SIMULINK model.

Gradual generation of the simulator is sketched in figure 5.

**Simulation chips – standardized communication tool**

In through all the development stages of the simulator the simulation chips proved to be very useful. They promote structured design of a model, serve as an up-to-date documentation of the model and succeeded as a standardized communication tool between technicians and physiologists.
From simulation chips to biomedical simulator

**Figure 4:** Communication of the system Control Web with driver measuring/controlling adapters during creation of industrial applications and inserting of simulating model to the driver of the “virtual card” during creation of simulator Golem in the system Control Web.

**Figure 5:** Gradual generation of the simulator
References


MEFANET report, vol. 3, 2010

Edited by Dušek, Vladimír Mihál, Stanislav Štípek, Jarmila Potomková, Daniel Schwarz, Lenka Šnайдrová

Institute of Biostatistics and Analysis. Masaryk University, 2010

ISSN 1004-2961

Web simulator creation technology

Kofránek, Jiří; Mateják, Marek and Privitzer, Pavol

pp. 12-97
MEFANET report 03

Medical teaching with the use of advanced technology

Editors:
Daniel Schwarz
Martin Komenda
Stanislav Štípek
Vladimír Mihál
Ladislav Dušek
WEB SIMULATOR CREATION TECHNOLOGY

Kofránek J., Mateják M., Privitzer P.

Laboratory of Biocybernetics, Dept.of Pathophysiology, 1st Faculty of Medicine, Charles University, Prague

Abstract
This document includes description of technologies that we use for the creation of web-based tutorials, educational and teaching simulators. When creating a web simulator, two types of problems must be taken into consideration. The first problem is the creation and identification of the mathematical model. This work is more of a research than development work, based on the creation of formalized mathematical description of the modeled reality. For the creation, tweaking/debugging and verification of simulation models, special software development tools are used. For a long time we have mostly been using Matlab/Simulink models, made by Mathworks, for the development process. Simulink belongs to block-oriented simulation languages, which enables the user to assemble computer models from each block with defined inputs and outputs, interconnect these blocks into computer networks and group computer networks into blocks with higher hierarchy. From the description of the block-oriented structure it is clear, how the values of each variable parameter is calculated in the model, that is, what is the algorithm for the relevant calculation process. Recently, we have been using a simulation environment based on the Modelica language. The most important innovation in Modelica is the option to describe each part of the model as a set of quotations and not as an algorithm used to solve these quotations. Models created in Modelica are well-arranged and better reflect the structure of the modeled reality. The other problem apparent during the creation of tutorial and educational simulators is the creation of the tutorial software itself. It is a very demanding development work, which requires the combination of ideas and experiences of teachers who create the script of the tutorial application, the creativity of art designers who create the multimedia components interconnected with the simulation model in the background, as well as the efforts of programmers who finally “sew up” the final masterpiece into its final shape. To automate the model debugging transfer from the simulation development environment (using Simulink or Modelica) into the development environment where the development application is programmed, specialized software tools (developed by us) are used. We have been creating tutorial simulators in ControlWeb development environments (originally designed for the creation of industrial control and measuring applications), in Microsoft .NET and Adobe Flash environments. Recently, we began using the Microsoft Silverlight platform, which enables distribution of simulators over the internet and may be executed directly into the internet browser environment (even on computers running various operating systems).

Keywords
Modeling, Simulation, Simulator
In place of an introduction – a web of physiological regulations

Thirty-six years ago the Annual Review of Physiology published an article (Gyton et al. 1972) which at a glance was entirely different from the usual physiological articles of that time. It was introduced by a large diagram on an insertion. Full of lines and interconnected elements, the drawing vaguely resembled an electrical wiring diagram at first sight (Fig. 1). However, instead of vacuum tubes or other electrical components, it showed interconnected computation blocks (multipliers, dividers, adders, integrators, functional blocks) that symbolized mathematical operations performed on physiological variables (Fig. 2). Bundles of connecting wires between the blocks indicated the complex feedback interconnection of physiological variables at first glance. The blocks were arranged in eighteen groups that represented individual interconnected physiological subsystems. In the centre was a subsystem representing circulatory dynamics – linked through feedback links with other blocks: From the kidneys to tissue fluids and electrolytes to autonomic nervous control and hormonal control including ADH, angiotensin and aldosterone (Fig. 3).

In this entirely new manner, using graphically represented mathematical symbols, the authors described the physiological regulations of the circulatory system and its broader physiological relations and links with the other subsystems in the body – the kidneys, volumetric and electrolyte balance control, etc. Instead of an extensive set of mathematical equations, the article used a graphical representation of mathematical relations. This syntax allowed depicting relations between individual physiological variables graphically in the form of interconnected blocks representing mathematical operations. The whole diagram thus featured a formalized description of physiological relations in the circulatory system using a graphically represented mathematical model.

The actual description of the model in the article was mostly represented by the elementary (but fully illustrative) drawing. Comments on and reasons for the formulation of the mathematical relations were very brief, e.g.: “Blocks 266 through 270 calculate the effect of cell pO2, autonomic stimulation, and basic rate of oxygen consumption by the tissues on the actual rate of oxygen consumption by the tissues”. This required exceptional concentration (and certain physiological and mathematical knowledge) from the reader to be able to understand the formalized relations between physiological variables.

A monograph (Guyton et al. 1973) published a year later, in 1973, explained a number of the adopted approaches in greater detail.

We do not see a mathematical representation of reality very often in biology and medicine. It should be noted that the process of formalization, i.e. the translation of a purely verbal description of a given network of relations into the formalized language of mathematics, is delayed in biological and medical sciences in comparison with engineering sciences, physics or chemistry. While the process of formalization in physics started as early as the seventeenth century, in medical and biological sciences it has been relatively delayed due to the complicacy and complexity of biological systems and has only advanced with cybernetics and computer technology. The methodological tool here is computer models built on a mathematical description of biological reality.

Formalized descriptions in physiology have been used since the late sixties (since the pioneering works of Grodins et al., 1967, describing respiration). Guyton’s model was the first extensive mathematical description of the physiological functions of interconnected body subsystems and launched the field of physiological research that is sometimes described as “integrative physiology” today. Just as theoretical
physics tries to describe physical reality and explain the results of experimental research using formal means, “integrative physiology” strives to create a formalized description of the interconnection of physiological controls based on experimental results and explain their function in the development of various diseases. From this point of view, Guyton’s model was a milestone, trying to adopt a systematic view of physiological controls to capture the dynamics of relations between the regulation of the circulation, kidneys, the respiration and the volume and ionic composition of body fluids by means of a graphically represented network.

Guyton’s graphical notation of a formalized description of physiological relations provides a very clear representation of mathematical correlations – the blocks in network nodes represent graphical symbols for individual mathematical operations and the wires represent individual variables. Guyton’s graphical notation was soon adopted by other authors – such as Ikeda et al. (1979) in Japan and Amosov et al. (1977) in the former USSR. However, the graphical notation of the mathematical model using a network of interconnected blocks was only visualization when created – Guyton’s model and later modifications (as well as the models of other authors that adopted Guyton’s representative notation) were originally implemented in Fortran and later in C++.

Today’s situation is different. Today, there are specialized software simulation environments available for the development, debugging and verification of simulation models, which allow creating a model in graphical form and then testing its behaviour. One of these is the Matlab/Simulink development environment by Mathworks, which allows building a simulation model gradually from individual components – types of software simulation elements that are interconnected using a computer mouse to form simulation networks. Simulink blocks are very similar to the elements used by Guyton for the formalized representation of physiological relations. The only difference is in their graphical form (see Fig. 4).

This similarity inspired us to use Simulink to revive Guyton’s good, classic diagram and transform it into a working simulation model. When implementing the model in Simulink, we used switches that allow us to connect and disconnect individual subsystems and control loops while the model is running. We strove to keep the appearance of the Simulink model identical to the original graphic diagram – the arrangement, wire location, variable names and block numbers are the same.

The simulation visualization of the old diagram was not without difficulties – there are errors in the original graphic diagram of the model! It does not matter in the hand-drawn illustration but if we try to bring it to life in Simulink, the model as a whole collapses immediately. There weren’t too many errors – switched signs, a divider instead of a multiplier, mixed-up interconnections between blocks, a missing decimal point in a constant, etc. However, there were enough to prevent the model from working. Some of the errors could be seen at first sight (even with no knowledge of physiology) – it is obvious from the diagram that the value of some variables in some integrators would quickly grow to infinity in operation (because of incorrectly drawn feedback) and the model would collapse. With a knowledge of physiology and system analysis, however, all of the errors could be identified with some work (Fig. 5). A detailed description of the errors and their corrections is in Kofranek et al. 2007).

It is interesting that Guyton’s diagram as a complex drawing was reprinted many times in various publications (recently see e.g. Hall, 2004, Van Vliet and Montani, 2005). However, nobody mentioned the errors or made an effort to correct them. This was understandable at the time the diagram was created. Drawing software did not exist – the diagram was created as
Figure 1. Guyton’s blood circulation regulation diagram from 1972.

Figure 2. Individual elements in the block diagram of Guyton’s model represent mathematical operations, the interconnection of elements represents equations in a graphically expressed mathematical model.
Figure 3. Individual interconnected subsystems in Guyton's model.

Figure 4. Appearance of blocks in Guyton's original graphical notation and in Simulink.
Figure 5. Correction of errors in Guyton's original diagram.
a complex drawing – and redrawing the complex diagram manually was not easy. It is also possible that the authors of the model did not wish to correct the errors – those who went to the trouble of analyzing the model could spot the drawing’s “typos” and those who would just like to unthinkingly copy the diagram were out of luck. After all, the authors used to send out the source codes of Fortran programs for their model – so if anybody wished just to test the behaviour of the model, they did not have to program anything (at most, they had to routinely convert the program from Fortran to another programming language).

Our Simulink implementation of Guyton’s (corrected) model (Figs. 6 and 7) is available for download at www.physiome.cz/guyton. Also available at that address is our Simulink implementation of a much more complex, later model from Guyton et al. There is also a very detailed description of all applied mathematical relations with an explanation.

**Block-oriented simulation networks**

Guyton’s block diagram augured the rise of visual, block-oriented simulation languages. However, Guyton and his colleagues implemented the model in Fortran back in 1972 – Simulink version 1 was released eighteen years later (in 1990). Block-oriented simulation languages, of which Simulink is a typical example, allow assembling computer models from individual blocks with defined inputs and outputs. The blocks are grouped in libraries; when building a model, a computer mouse is used to create individual block instances, with inputs and outputs connected through wires that “conduct” information.

A Simulink network can be arranged hierarchically. Blocks can be grouped into subsystems that communicate with their ambient environment through defined input and output “pins”, making “simulation chips” of a sort. A simulation chip hides the simula-
tion network structure from the user, much like an electronic chip hiding the interconnection of transistors and other electronic elements. Then the user can be concerned just with the behaviour of the chip and does not have to bother about the internal structure and calculation algorithm. The behaviour of a simulation chip can be tested by monitoring its outputs using virtual displays or virtual oscilloscopes connected to it. This is very useful especially for testing the behaviour of a model and expressing the mutual relations of variables.

The entire complex model can be then visualized as interconnected simulation chips and the structure of their interconnection clearly shows what effects are taken into account in the model, and how (Figs. 8–11). This is very useful for interdisciplinary collaboration – especially in borderline fields such as biomedical system modelling (Kofránk et al., 2002). An experimental physiologist does not have to examine the details of mathematical relations hidden “inside” a simulation chip; however, from the mutual interconnection of simulation chips they will understand the model structure and will be able to check its behaviour in the appropriate simulation visualization environment. Simulation chips can be stored in libraries and users can create their instances for use in their models (Fig. 12). For example, we created a Physiolibrary for modelling physiological regulations (http://www.physiome.cz/simchips).

Hierarchical, block-oriented simulation tools are thus used advantageously in the description of the complex regulation systems that we have in physiology. A formalized description of physiological systems is the subject matter of PHYSIOME, an international project that is a successor to the GENOME project. The output of the GENOME project was a detailed description of the human genome; the goal of the PHYSIOME project is a formalized description of physiological functions. It uses computer models as its methodological tool (Bassingthwaighte, 2000; Hunter et al., 2002).
Several block-oriented simulation tools developed under the PHYSIOME project have been used as a reference database for a formalized description of the structure of complex physiological models. These include JSIM (http://www.physiome.org/model/doku.php) and CElML (http://www.cellml.org/). Prof. Guyton’s disciples and followers have expanded the original, extensive simulator of the circulatory system (Quantitative Circulatory Physiology (Abram et al. 2007) with an integrated connection of all important physiological systems. The latest result is the Quantitative Human Physiology simulator (Hester et al. 2008), now also distributed as “Digital Human”, which represents today’s most comprehensive and largest model of physiological functions. The model can be downloaded from http://physiology.umc.edu/themodelingworkshop/. The authors developed a special block-oriented simulation system to represent the complex model structure.

Causal and acausal approaches

Block-oriented tools work with hierarchically interconnected blocks. The connections between blocks “conduct” signals that transmit the values of individual variables from the output of a block to the inputs of other blocks. The blocks process input information into output information.
The hierarchically arranged block-oriented description clearly shows how the values of individual variables are calculated in the model – i.e. what the calculation algorithm is. However, the interconnection of blocks in a network of relations cannot be completely arbitrary. Interconnected elements may not include any algebraic loops – i.e. cyclic structures where an input value fed to the input of a calculation block depends (through several intermediate blocks) on the block’s output value in the same time step.

For illustration, let us consider the small example of an algebraic loop in Simulink, a block-oriented language. A model of the kidneys uses a simulation chip calculating the glomerular filtration rate. The individual inputs and outputs of that chip are shown in Fig. 13. The inside of the simulation chip consists in elementary blocks performing mathematical operations. The value of GFR, a variable representing the glomerular filtration rate, is calculated from the value of NETP; to calculate NETP it is necessary to know the value of PTP, which is however calculated as the quotient of GFR and TUBC (Fig. 14). Our Simulink diagram contains an algebraic loop that must be broken. Therefore we solve an implicit equation in the blocks identified as “Algebraic Constraint” in Fig. 15 to calculate GFR in each integration step. Therefore, a Simulink network does not constitute the graphical representation of mathematical relations in a model; rather, it is the graphical representation of a chain of transformations from input values to output values through Simulink elements where loops are not allowed.

If we focus on the representation of a structure of mathematical relations rather than the algorithm of calculations when building a model in Simulink, we can easily introduce algebraic loops into the model.
There are methods that can be used to get rid of algebraic loops (e.g. see Dabney and Harman, 2004) – however, they lead to transformations that make the model structure even more complex and the model more difficult to understand. The need to have a fixed direction of connection from inputs to outputs with no algebraic loops also makes model building more difficult. The interconnection of blocks in Simulink thus reflects the calculation procedure rather than the actual structure of the modelled reality. We call this causal modelling. In complex systems, the physical reality of the modelled system becomes somewhat lost under the structure of calculation with this approach.

New, “acausal tools have recently been developed for the creation of simulation models. The major innovation brought about by acausal modelling tools is the possibility to describe the individual parts of a model directly as a system of equations rather than an algorithm for solving the equations. The notation of models is declarative (we describe the structure and mathematical relations, not the calculation algorithm) – thus the notation is acausal. Acausal modelling tools work with interconnected components that are instances of classes in which equations are directly defined.

The components (i.e. instances of classes with equations) can be interconnected by
Simulink blocks reflects the calculation procedure rather than a graphical representation of mathematical relations.

Figure 14. The interconnection of individual blocks inside the “Glomerular Filtration” simulation chip graphically represents individual mathematical relations for the calculation of the glomerular filtration rate. However, there is an algebraic loop, it is necessary to break the loop.

Figure 15. Breaking the algebraic loop in the calculation of the glomerular filtration rate. The interconnection of Simulink blocks reflects the calculation procedure rather than a graphical representation of mathematical relations.
means of precisely defined interfaces – connectors; this defines a system of equations. The latest version of Simulink provides certain options for using acausal tools as well. Mathworks, the producer of the Matlab/Simulink simulation tools, responded to the new trends by creating a special acausal Simulink library – Simscape – and related domain libraries such as SimElectronics, SimHydraulics, SimMechanics, etc.

A modern simulation language that is built directly on the acausal notation of models is Modelica (Fritzson, 2003). It was originally developed in Sweden and is now available both in an open-source version (developed under the auspices of Modelica Association, http://www.modelica.org/) and in two commercial implementations.

The first commercial implementation is made by Dynasim AB – which has been bought by Dassault Systemes, a multinational corporation (sold under the name of Dymola, currently in version 7.1), and the other commercial implementation is made by MathCore (sold under the name of Math-Modelica). Dynasim’s Modelica has a good connection to the Matlab and Simulink simulation tools, while MathModelica can connect to the Mathematica environment made by WoModelica works with interconnected components that are instances of individual classes. Unlike the implementation of classes in other object-oriented languages (such as C# or Java), classes in Modelica have an additional special section in which equations are defined.

The equations do not mean assignment (i.e. storing the result of the calculation of an assigned command in a variable) but rather the definition of relations between variables (as is common in mathematics and physics). For example, the following notations of relations between variables expressing the resistance (R), flow (F) and pressure gradient (P) are equivalent:

\[
F = P / R \\
F = P / R = F \\
P = R * F \\
P = R * F = P \\
R = P / F \\
P = P / F = R
\]

Components (class instances) in Modelica can be interconnected by means of precisely defined interfaces – connectors.

Generalized system properties:
- \(e\) means generalized effort – corresponding to force in mechanics, voltage in electrical diagrams, pressure in hydraulics, etc.
- \(f\) is generalized flow – corresponding to velocity in mechanics, current in electrical diagrams, flow rate in hydraulics, temperature flow in thermodynamics, etc.
- \(q\) is generalized accumulation or deflection, representing the integral of the generalized flow. It corresponds e.g. to the stretching of a spring in mechanics, fluid volume in hydraulics, charge in electrical diagrams, accumulated heat in thermodynamics, etc.
- \(p\) is generalized momentum (inertance) – the integral of the generalized effort, representing kinetic energy; in hydraulics it represents the change of the flow rate proportional to the pressure difference (flow momentum), in electrical circuits it is the potential needed to change an electric current (induction), etc.
- \(R\), \(C\) and \(L\) represent constants of proportionality between the generalized system properties. They correspond e.g. to resistance, capacitance or weight.

**Figure 16. Relations between generalized system properties.**
What is important is that the interconnection of components actually interconnects systems of equations in the individual components with one another. By interconnecting Modelica components, we do not define the calculation procedure but rather the modelled reality. The method of solving the equations is then “left to the machines”.

Generalized system properties

The representation of a model in an acausal simulation environment resembles the physical reality of the modelled world more than the standard interconnected block diagrams in causal modelling tools. This is associated with the generalized system properties of the real world (Fig. 16), where an important role is played by generalized effort (corresponding to force, pressure, voltage, etc. in the real world) and generalized flow (corresponding to current, flow rate, etc. in the real world). The integral of generalized flow is generalized accumulation or deflection (in the real world, this can be e.g. an electrical charge but also the volume of a liquid or gas, stretching of a spring, accumulated heat, etc.). The integral of generalized effort is generalized momentum (this represents flow momentum in hydraulics, induction in electrical circuits, etc.).

Also related to the generalized system properties is the fact that descriptions of models of biological or physiological processes often use electrical or hydraulic analogies for reasons of clarity.

Let us illustrate the utilization of generalized system properties and the difference between modelling in block-oriented simulation tools and in Modelica with a physiological reality modelling example – a model of simple pulmonary ventilation mechanics. Let us consider a simple pulmonary mechanics model that is schematically shown in Fig. 17. With a high level of simplification, the lungs can be seen as three bags interconnected through two tubes. The lungs are connected to the fan of the artificial pulmonary ventilation equipment, which periodically drives air into the lungs with the pressure $PAO$. $P0$ is the pressure of the ambient atmosphere. Airflow $Q$ runs through the upper respiratory tract that has the resist-
ance $RC$. From the upper respiratory tract, air forces its way through the lower respiratory tract to the alveoli. The resistance of the lower respiratory tract is $RP$, the pressure in the central parts of the respiratory tract (at the boundary between the upper and lower respiratory tracts) is $PAW$, the pressure in the alveoli is $PA$. Air expands the pulmonary alveoli, whose compliance is $CL$ (as the total compliance of the lungs). The interpleural cavity is in between the lungs and the rib cage. The pressure in it is $PPL$. The chest has to expand as well during artificial pulmonary ventilation when air is forced into the lungs – the chest compliance is $CW$. The small portion of air that does not reach the alveoli expands the respiratory tract instead – its compliance is $CS$.

Now we can set up our equations. According to Ohm’s law, it must be true that:

\[
PAW - PA = RP \cdot QA \tag{1}
\]

The relation between the compliance, pressure gradient and volume (expressed as the integral of the flow rate) is expressed by these equations:

\[
PA - PPL = \frac{1}{CL} \int QA \, dt \quad \text{(2)}
\]

\[
PPL - P0 = \frac{1}{CW} \int QA \, dt
\]

\[
PAW - P0 = \frac{1}{CS} \int (Q - QA) \, dt
\]

According to the generalized Kirchhoff’s law, the sum of all pressures (voltages) along a closed loop must be equal to zero, i.e. the following must hold true for the loop along the node $PAW$ and along the node $PAO$:

\[
(PAW - PA) + (PA - PPL) + (PPL - P0) + (P0 - PAW) = 0
\]

\[
(PAO - PAW) + (PAW - P0) + (P0 - PAO) = 0 \tag{3}
\]

Substituting from the equations for Ohm’s law and compliances, we get:

\[
\begin{align*}
RPQA + \left( \frac{1}{CL} + \frac{1}{CW} \right) \int QA \, dt - \frac{1}{CS} \int (Q - QA) \, dt &= 0 \\
QRC + \frac{1}{CS} \int (Q - QA) \, dt + (P0 - PAO) &= 0
\end{align*} \tag{4}
\]

**Causal approach – implementation of the pulmonary ventilation mechanics model in Simulink**

When building a model in Simulink, we have to define precisely the *procedure of calculation* from input variables to output variables. If we wish to calculate the reaction of the air flow to/from the lungs ($Q$) to the input – i.e. to the changes in pressure at the beginning of the respiratory tract ($PAO$) caused by the artificial pulmonary ventilation apparatus – the Simulink model will look like Fig. 18. We can also simplify the Simulink model. First we obtain a differential equation (input variable $PAO$, output $Q$) from equations (4):

\[
\frac{d^2 PAO}{dt^2} + \frac{1}{RP \cdot CT} \frac{dPAO}{dt} = RC \cdot \frac{d^2 Q}{dt^2} + \left( \frac{1}{CS} + \frac{RC}{RP \cdot CT} \right) \frac{dQ}{dt} + \frac{1}{RP \cdot CS} \left( \frac{1}{CL} + \frac{1}{CW} \right) Q \tag{5}
\]

When we enter the numeric parameters of resistance (in units cm H₂O/L/sec) and compliance (in units L/cmH₂O) (Khoo, 2000):

\[
RC = 1; \quad RP = 0.5; \quad CL = 0.2; \quad CW = 0.2; \quad CS = 0.005
\]

the equation (5) simplifies:
In the Laplace transform of the equation (7), we get:

\[
\frac{Q(s)}{PAO(s)} = \frac{s^2 + 420s}{s^2 + 620s + 4000}
\]  

This allows simplifying the Simulink model (Fig. 19):

However, when the values of the parameters change, the transform function (6) must be recalculated and the Simulink model will change.

Now we will make the model a little more complex by taking \textit{air inertia} in the upper respiratory tract into account (Fig. 20).

In addition, we will now take into account the inertial element \( LC = 0.01 \text{ cm H}_2\text{O s}^2 \text{ L}^{-1} \):

\[
PLC \frac{dQ}{dt} = \Delta P
\]

where \( P \) is the pressure gradient and \( \frac{dQ}{dt} \) is the flow acceleration, or:

\[
\Delta P = LC \frac{dQ}{dt}
\]

Then we get this instead of the system of equations (4):

\[
\begin{align*}
RP \frac{dQ}{dt} \left( \frac{1}{CL} + \frac{1}{CW} \right) Q_A - \frac{1}{CS} (Q - QA) &= 0 \\
RC \frac{dQ}{dt} + LC \frac{d^2Q}{dt^2} + \frac{1}{CS} (Q - QA) + \frac{dP}{dt} - \frac{dPAO}{dt} &= 0
\end{align*}
\]

Instead of the equation (7), we get:

\[
\frac{d^2Q}{dt^2} = 2,5
\]

\[
+ 620 \frac{dQ}{dt} + 4000 Q
\]

and in the Laplace transform, we get:

\[
\frac{Q(s)}{PAO(s)} = \frac{s^2 + 420s}{s^2 + 620s + 4000}
\]

This Simulink model will change (Fig. 21): Since we always have to take into account the direction of the calculation in Simulink, the actual Simulink diagram is rather dissimilar to the physical reality of the described system. Even a small change in the model, such as the inclusion of the inertial element, requires careful calculation and a change in the model structure. The model will change significantly even if we consider spontaneous breathing instead of artificial pulmonary ventilation. The model input will be not the pressure \( PAO \) generated by the artificial pulmonary ventilation respirator but e.g. the compliance of the thoracic wall \( CW \) (a cyclic variation in the compliance can be used to model the function of the respiratory muscles).

**Acausal approach – implementation of the pulmonary ventilation mechanics model in Modelica**

Comparing the model structure in Figures 17 and 20, formulated by means of generalized state variables, to the implementation of the model in Simulink (Figures 18, 19, 21), we can see that the interconnected blocks in Simulink express the structure of the calculation procedure rather than the structure of the modelled reality.

In Modelica, this is different.

Acausal modelling tools, of which Modelica is a typical example, work with interconnected components that are instances of special classes in which \textit{equations} are defined. When modelling in Modelica, the first task is to formally express the modelled reality by means of equations.

In our simple pulmonary mechanics model, we describe the resistances of the respiratory tract, expansible elastic bags, and we
Figure 18. Simulink model implementation according to the equations (4).

Figure 19. Simulink model implementation using the Laplace transform according to the equation (7).

Figure 20. A simple pulmonary mechanics model taking inertia into account (hydraulic and electrical analogy).
might take into account air flow inertia (see Figures 17 and 20). The description of the air flow in the lungs belongs in the pneumatic domain. However, if we disregard the compressibility of gases, we can describe the model using the hydraulic domain. The same formal expression can be provided by an electrical analogy.

It is interesting that the individual fundamental elements have the same formal expression (Fig. 22) in different domains (electrical, hydraulic or pneumatic). This is due to the general system properties of the real world, where voltage or pressure correspond to generalized effort and electric current or medium flow correspond to generalized flow, as the case may be.

To build the pulmonary mechanics model in Modelica, we will need to define the equations of three elementary classes, whose instances we will use in the model. To express the resistance of the respiratory tract, we will use an instance of the Resistor class. The elastic respiratory tract, alveoli and chest will be described as elastic bags using an instance of the Capacitor class and the air flow inertia will be expressed using an instance of the Inductor class.

The fragment with an equation notation in the “Resistor” class, describing the relation between variables expressing the resistance (R), pressure gradient (v) and flow (i) in Modelica, is simple, according to Ohm’s law:

\[ R \cdot i = v; \]

\textit{end Resistor;}

The “Capacitor” class is used to describe an elastic bag expanded by air flow at the input. The compliance (C) characterizes the level of “expansibility” of the bag wall due to the pressure difference (v) between the air pressure forcing air into the bag and the pressure outside the elastic bag. The flow rate of air coming to the bag (i) is then described by the following equation in the Modelica language (where “der” means derivative):

\[ i = C \cdot \text{der}(v); \]

\textit{end Capacitor;}

The inertial element will be implemented in the model by means of the “Inductor” class. The force that accelerates air flow is the pressure gradient. According to Newton’s law, the acceleration of flow, i.e. the first derivative of flow \( \text{der}(i) \), is proportional to the pressure gradient (v) and inversely proportional to the weight of the selected gas column, called inertance (L). We can thus describe the relation between a change in the flow rate (i) and the pressure gradient (v) depending on inertance (L) using a simple equation in the “Inductor” class:

\[ L \cdot \text{der}(i) = v; \]

\textit{end Inductor;}

Instances of the above-mentioned fundamental elements are interconnected in a network by means of connectors – two interconnecting connectors, labelled “p” and “n”, are defined for each of the elements. Voltage, or pressure for the hydraulic or pneumatic domain, is fed to each of them (p.v, n.v) when connected and an electric current or medium flow (p.i, n.i) can flow through the connectors.

Connectors are instances of special connector classes, in which the variables used for interconnection are defined. Components can be interconnected by means of connectors that are instances of the same connector classes (the “interconnection sockets” must be of the same type). In our case, connectors “p” and “n” are instances of the “Pin” connector class, which is able to interconnect voltages or pressures (p.v, n.v) and flows (p.i, n.i) with the environment. Values from the connectors are interconnected with the values of the variables (i) and (v) inside the individual fundamental elements.

As a rule, flow does not disappear anywhere in any of the above-mentioned fundamental elements – what flows into an element also
flows from it \( (i=p.i=n.i) \), and the appropriate gradient is calculated from the difference in voltages or pressures \( (v=p.v-n.v) \). Implementing this requirement is simple – since Modelica is an object-oriented language, all three of the above-mentioned classes of fundamental elements will have a common ancestor \( \text{OnePort} \) from which they will inherit connectors “p” and “n” as well as the following equations:

\[
\begin{align*}
ev & = p.v - n.v; \\
0 & = p.i - n.i; \\
i & = p.i;
\end{align*}
\]

\text{end OnePort;}

The equations will thus connect the values of the pressures or voltages fed from the environment to connectors “p” and “n” \( (p.v, n.v) \) with the pressure or voltage gradient \( (v) \) and express the same value of (electrical or hydraulic) flow at both connectors \( (p.i, n.i) \) and inside the component \( (i) \).

Connector classes define the manner in which Modelica components communicate with one another. Figuratively speaking, by defining connector classes we define the types of “sockets”. In connectors we define individual variables that the connector will use to interconnect a component with its environment.

It is defined for each variable in a connector whether it represents a flow (then the variable is identified with a “flow” attribute) or not (“non-flow” variables). This differentiation is important for the correct interpretation of the interconnection of individual components (instances of element classes) through the appropriate connectors (see Fig. 23). For flow variables, it is obvious that we must make sure the entity in question (whose flow the variable characterizes) neither disappears nor accumulates anywhere in the interconnection. Therefore, the sum of all interconnected variables with the “flow” attribute must be zero (as according to Kirchhoff’s law in the electrical domain). For non-flow variables, an interconnection defines that their values must be the same for all interconnected connectors (according to Kirchhoff’s first law). By interconnecting the instances of individual fundamental elements through connectors, we express the requirement of the zero algebraic sum of the values of interconnected flow variables and the requirement of the equality of the values of interconnected non-flow variables.

Each Modelica class can have a graphical representation – this is important especially for depicting the interconnection of instances where components are interconnected to create a clear graphical structure of a model. That is why we can also define an icon for each class in Modelica. The icon can be animated.

We can then create a model graphically in Modelica, by interconnecting the instances of individual elements that we select from a library with the mouse and setting the val-
ues of the appropriate parameters in a dialogue box.

For the implementation of our pulmonary ventilation model, we need to interconnect instances of the “Resistor”, “Capacitor” and “Inductor” elements. However, we do not have to program the fundamental elements we need from the very beginning – Modelica includes extensive libraries from various physical domains (electric, hydraulic, mechanic, etc.) where such elements can be found.

In our specific case, we can take advantage of e.g. the visual components of electrical circuits for a quick solution – we will create the individual instances (RC, RP, CL, CW and CS), enter the appropriate values of parameters (C and R) and interconnect the components with a connector.

The result is shown in Fig. 24. Comparing the model structure implemented in Modelica with the original schematic drawing showing the model structure (Fig. 17), we can see that the Modelica solution is straightforward and (unlike the Simulink implementation – see Figs. 18 and 19) the structure of the model corresponds to the structure of the modelled reality.

Increasing the complexity of the model by including an inertial element does not cause

---

**Figure 22.** The hydraulic and electric elements are from different domains but have the same formal description. The analogy of voltage (v) in the hydraulic domain is pressure, the analogy of current (i) in the hydraulic domain is a stream of fluid (and a stream of gas in the pneumatic domain). Hydraulic resistance (R) follows Ohm’s law in the same way as electric resistance (the voltage difference is just replaced with the pressure gradient and the current is replaced with the flow rate). The hydraulic analogy of a capacitor is an elastic bag expanded by the difference in pressures inside and outside the bag. The analogy of the electric capacity of a capacitor (C) is the compliance of the elastic bag wall. When we include inertia in a hydraulic system, the force that accelerates fluid flow is the pressure gradient. According to Newton’s law, the acceleration of flow, i.e. the first derivative of flow $\text{der}(i)$, is proportional to the pressure gradient (v) and inversely proportional to the weight of the selected fluid column, called inertance (L). In the electrical domain, inertance corresponds to coil inductance. Each element from the hydraulic or electric domain has two interconnecting connectors through which electric current or medium flow (p.i, n.i) flows in and out; as a rule, the running flow (i) never disappears in the element (i.e. $i = p.i = n.i$). Simultaneously, voltage or pressure (p.v, n.v) is connected to the connectors by interconnecting into a network, and a voltage gradient or a pressure gradient (v) builds up in the element.
any significant trouble – we just pick up the appropriate inertial component (LC) from the library with the mouse, set the value of its parameter (L) and interconnect it in the model. The structure of the model implemented in Modelica, shown in Fig. 25, corresponds to the structure of the modelled reality (see Fig. 20), while the structure of the Simulink implementation (Fig. 19) corresponds more to the method of solution for the model’s equations.

The fundamental elements of the simulated reality can have very trivial notation of relations between the variables in question. A resistor, capacitor or coil from the electrical physical domain or their hydraulic analogies are illustrative examples of this. A complex system for calculation will ensue from interconnecting the fundamental elements in networks – a system of equations will result from their mutual interconnections. Their numerical solution in causal simulation tools may not be trivial at all – e.g. more complex R-C-L models of circulation or respiration implemented in Simulink are very complex (see e.g. circulation models in our Simulink library, Physiolibrary – http://www.physiome.cz/simchips).

In Modelica, we do not have to bother with the method of solution for equations. Instead, more attention should be paid to the definition of equations in individual elements and interconnection of their instances (individual components). In Modelica, the acausal tool itself will take care of the algorithm for solving the resulting system of equations and we can monitor the appropriate flows and pressures in various places in the simulated circuit when the simulation is launched.

### Causal and acausal connectors

The acausal connector interconnection of components is implemented by means of two types of variables: one representing a flow – for this, it holds true that the sum of flow values in all connected nodes is zero (because no medium accumulates in the area of branching into connected nodes); and one whose value remains the same in all connected nodes. It is advisable that each variable with the flow attribute is accompanied by a non-flow variable representing the generalized effort in relation to the flow variable in the connector interconnection.

Unlike Simulink components (which have defined component inputs and component outputs), we do not define what is an input and what is an output in an acausal interconnection. An acausal Modelica component does not calculate output values from input values. The interconnection of Modelica components by means of acausal connectors interconnects the equations in individual components into systems of equations.

In addition to acausal linking connectors, Modelica classes may include causal input connectors that are used to feed actual input variables from the environment, as well as causal output connectors that serve to send output variables to the environment.

In addition to equations, Modelica classes may also include a precisely defined algo-
Figure 24. The implementation of the pulmonary mechanics model (according to Fig. 17) in Modelica resembles the modelled reality much more than the implementation in Simulink.

Figure 25. In Modelica, implementing the pulmonary mechanics model taking into account the inertial element (according to Fig. 20) merely requires adding the LC inertial component.
Algorithm for the calculation of output values from input values (a typical example is the modelling of functional dependencies). Modelica components are thus interconnected using both acausal links and causal, directional inputs and outputs. Causal connectors usually distribute signals – e.g. in a blood circulation model, signal causal inputs may contain signals used to set resistance values in components representing the resistance of the circulatory system.

Consequently, a Modelica model is usually represented by a graphical set of components interconnected using both acausal and causal links. Components are instances of Modelica classes whose structure may also be represented as a network of interconnected instances.

An example of the definition and use of an acausal element – elastic compartment

Let us see a simple example of the definition and use of a Modelica class. When modelling the dynamics of blood vessels, we often need an elastic (inflatable) compartment. Therefore we will define a class named `VascularElasticBloodCompartment` whose instances will be elastic, acausally interconnected compartments that can be connected to the “distribution” of a fluid through an acausal connector – the fluid may flow to/from the compartment at a certain rate and under a certain pressure. We can assign a graphic icon to each class representing a model or connector in the programming environment. We can create an icon for our elastic compartment, too (Fig. 26).

This is not just a school example – we take this compartment into account in our Modelica implementation of an extensive model of physiological functions, “Quantitative Human Physiology” (Abram 2007, Coleman et al, 2008). Fig. 28 shows an example of the use of instances of the elastic compartment in our implementation of this extensive model.

We can imagine the elastic vascular compartment (Fig. 27) as an inflatable bag with one acausal interconnecting connector (let us name it e.g. “ReferencePoint”) that we will use to connect to the environment – this connector will provide us with two variables:

- flow “ReferencePoint.q”
- pressure “ReferencePoint.pressure”

If the connector is connected to the environment through a connector, the pressure value will truly be the same in all nodes connected to the compartment, and the flow will be distributed to all connected nodes so that its algebraic sum will be zero (nothing ever accumulates in the area of branching) – see the example of the component connection in Fig. 28.

Three signal (causal) inputs will enter the compartment from the outside:

- Basic charge “V0” – the value of the volume that must be reached before
the pressure in the elastic compartment starts increasing. If the volume is less than zero, the pressure in the compartment will be zero.

- Outer, external pressure “ExternalPressure” – the pressure of the ambient environment on the elastic compartment.

- “Compliance” of the elastic compartment – the pressure in the compartment will be inversely proportional to it if the compartment volume exceeds the basic charge.

Two (causal) signal outputs will go from the compartment to its environment:

- Information about the compartment’s current volume, “Vol”

- Information about the pressure inside the compartment, “Pressure”

It is useful to design another parameter for the compartment (whose value will be read before the start of simulation), which would specify its initial charge:

Initial compartment volume, “initialVol”

We can also design an icon to display the elastic component in the programming environment.

The actual fragment of code describing the behaviour of the elastic compartment looks like this in Modelica:

```model VascularElasticBloodCompartment extends QHP.Library.Interfaces.BaseModel;
```

Figure 27. The concept of an elastic vascular compartment is based on the idea that when a blood vessel fills with blood, the pressure in the vessel is determined only by the external pressure on the vessel until a certain residual volume ($V_0$) is achieved; the elastic and muscle fibres in the blood vessel will then start to tense and compress the volume of blood in the vessel with the VascularPressure pressure. If we label the volume of fluid in the blood vessel $V$, then the volume of blood stressing the vessel ($\text{StressedVolume}$) will determine the Pressure inside the vessel depending on its Compliance and on the external pressure on the vessel (ExternalPressure). The vascular compartment is connected to the system by means of the ReferencePoint connector, through which blood may flow into the compartment (at the rate of referencePoint.$q$) under pressure (referencePoint.pressure).
The first line declares the model class; in addition, there is the declaration of a real variable, “StressedVolume”, whose physical units will be checked. This is not just a question of code clarity and readability. The check of unit compatibility will enable us to avoid a very hard-to-find error, when we exchange connectors in interconnections by mistake (if units are found to be incompatible, the check will not allow us to create the wrong interconnection at all).

Then there is the declaration of an “InitialVol” parameter, whose physical units will be checked as well. And then there is the equation section. The initialization of the compartment’s initial volume, i.e. the variable “Vol”, is declared first. The other lines in the equation section declare four equations. The first one is a differential equation – the derivative of the volume “der(Vol)” equals the inflow “q” from the connector “referencePoint”. The next equation declares that the value of the elastically stressed volume “Stressed-
Volume will be calculated as the difference between the compartment volume \( \text{Vol} \) and the value of its basic charge \( \text{V0} \) (which is an input); the equation also says that the value of the compartment volume may never drop down to negative values. The third equation declares the relation between the “Pressure” in the compartment, the value of the “StressedVolume”, the “Compliance” and the “ExternalPressure”. We would like to repeat that these are not assignments but equations. The equation could also be written like this in Modelica:

\[
\text{Pressure} - \text{ExternalPressure} = \frac{\text{StressedVolume}}{\text{Compliance}};
\]

The last equation interconnects the value of “Pressure” in the compartment with the value of the pressure interconnected with its environment by the acausal connector through the “referencePoint.pressure”. The value of “Pressure” is also a signal output from the compartment – as a signal, it can be fed to other blocks – but it is a causal output (signal) variable and its value cannot be affected by what we connect it to. However, the situation is different with the interconnection from the acausal connector. When we interconnect an instance of the elastic compartment with other elements through the acausal connector, the four equations in the compartment will become part of the system of equations defined by the interconnection and the values of the variables in the elastic compartment instance will depend on the solution of the originated system of equations.

**Hybrid models**

Continuous dynamics expressed by a system of algebraic differential equations is often enough for the mathematical description of real-world models. However, we frequently need to represent discontinuous, discrete behaviour (which is often an approximation of quick continuous processes in physical systems) and continuous dynamic systems themselves are not enough for the description of real-world processes – examples include the opening and closing of valves in the hydraulic and pneumatic domain, the behaviour of diodes in the electrical domain or the switching on/off of genes, the creation and transmission of nerve impulses or the opening and closing of ion channels in the biological domain. Discrete event dynamic systems are frequent in the description of technical applications. Discrete hierarchical state automata are a very powerful tool for the formalized description of processes and their interactions (Harel, 1987).

When modelling large systems, it is often useful to combine discrete and continuous description to a lesser or greater extent. Such “hybrid” models can combine discrete and continuous time variables, and generate and react to various events (see Fig. 29). Hybrid models are supported in modern development simulation environments. For example, a continuous dynamic system model in Simulink can be combined with hierarchical state automata created in a special modelling tool, Stateflow – the values of variables in Simulink can change the states of automata in Stateflow, and Stateflow can switch calculation blocks in Simulink by means of generated events, changing the calculation procedure.
However, acausal development tools can directly change the used equations (not just the method of solution). A small illustrative example can be the modelling of the average blood volume in a ventricle (see Fig. 30). A ventricle is modelled as a continuous pump with a variable internal volume. The ventricle model is connected to the circulation by means of acausal connectors “q_inŽ and “q_outŽ. These connectors interconnect the flow of blood into (“q_inŽ) and out of (“q_outŽ) the ventricle. The change of blood volume in the ventricle will be determined by the algebraic sum of flows in both acausal connectors. In Modelica, this will be written as follows:

\[
der(\text{Volume}) = q_{\text{in}}.q + q_{\text{out}}.q;
\]

The model has two causal inputs – one is the current flow in the ventricle (“BloodFlowŽ) and the other is the required volume of blood in the ventricle in the steady state (“VentricleSteadyStateVolumeŽ). If that volume is greater than the current ventricle volume (“VolumeŽ), then inflow to the ventricle will be set to a larger value than the outflow, proportionally to the difference between the required value and the actual value:

\[
q_{\text{in}}.q = \text{BloodFlow} + (\text{VentricleSteadyStateVolume} - \text{Volume}) * K;
\]

Outflow from the ventricle (“q_out. qŽ will be set to the value of “BloodFlowŽ with a negative sign, because it flows out of the compartment:

\[
q_{\text{out}}.q = -\text{BloodFlow};
\]
Alternatively, when the required value of blood volume in the ventricle ("Ventricle-SteadyStateVolume") is less than the actual value ("Volume"), the inflow of blood will be set to "BloodFlow" and the outflow of blood will be set to a larger value than the inflow, proportionally to the difference between the actual value and the required value. The equation notation fragment in Modelica then looks like this:

```model VentricleVolumeAndPumping;

  .

  equation
  der(Volume) = q_in.q + q_out.q
  if (SteadyStateVolume >= Volume) then
    q_in.q = BloodFlow + (VentricleSteadyStateVolume − Volume) * K;
    q_out.q = −BloodFlow;
  else
    q_in.q = BloodFlow;
    q_out.q = (BloodFlow + (Volume − VentricleSteadyStateVolume) * K);
  end if;
end VentricleVolumeAndPumping;
```

Two equations are then switched over in the model’s system of equations, depending on the values of the variables "Volume" and "VentricleSteadyStateVolume". At first sight, the notation looks like an assignment (as in standard programming languages) but they are equations. An equivalent notation may look like this:

```model VentricleVolumeAndPumping;

  .

  equation
  delta = (VentricleSteadyStateVolume − − Volume) * K;
  der(Volume) = delta;
  q_in.q + q_out.q = delta;
  if (delta<0) then
    q_in.q = BloodFlow;
  else
    q_in.q = BloodFlow + delta;
  end if;
end VentricleVolumeAndPumping;
```

Because they are equations, their order does not matter; nor does it matter whether the value of the variable "delta" in the third equation is on the right or on the left. The actual notation of the equations used in Modelica is even more compact:

```model VentricleVolumeAndPumping;

  .

  equation
  delta = (VentricleSteadyStateVolume − − Volume) * K;
  der(Volume) = delta;
  q_in.q + q_out.q = delta;
  q_in.q = if (delta<0) then BloodFlow else BloodFlow + delta;
end VentricleVolumeAndPumping;
```

Modelica allows describing discrete and continuous systems acausally, providing many possibilities of combining models with discrete and continuous parts. Details can be found in Fritzon, 2003.

Combining acausal and causal (signal) connections in hierarchically arranged models

Modelica makes modelling large systems easier and more controllable and supports their hierarchical decomposition. Modelica’s object-oriented architecture supports the structuring of models into suitable parts having a coherent meaning so that they can be examined separately under certain conditions or re-used (whether in a different place in the same model or in another model), greatly enhancing the clarity of the created models. That is why we create large, reusable libraries of Modelica “simulation chips” in Modelica and each model is usually accompanied by an extensive, hierarchically arranged library of elements. Hierarchical components can be clicked to expand, which will reveal their internal structure.
An example of the hierarchical structure of a Modelica program is the “Vascular-Compartments” class (see Fig. 31), which implements a part of the blood circulation subsystem and makes use of an instance of the above-mentioned class “Vascular-ElasticBloodCompartment”. Blood flows through acausal connectors between elastic compartment instances, resistances of individual parts of the vascular system and two pumps modelling the activity of the right and left ventricles. The component also uses causal signal connections. An entire set of signal connections (coming from outside the component) is distributed e.g. by the “OrganBloodFlowSignals” bus. Input signal connections control the value of the peripheral resistors and the pumping functions of the right and left ventricles. The structure of the model represents the structure of the modelled reality much better and much more clearly than models created in causal modelling environments. Just compare the Modelica model in Fig. 31 to the model shown in Fig. 6, implemented in Simulink. The two models represent roughly the same – the flow through an elastic vascular system and a heart pump (however, the Modelica model has more details). The Simulink model represents the calculation procedure rather than the structure of the modelled system. The advantage of acausal modelling tools is particularly evident in more complex models, where the possibility of hierarchical model decomposition is crucial for success, as it is important for the interconnection of components to always express in an aggregated manner the cardinal relations at a given hierarchical level while details can be obtained by digging deeper.
deeper into the structure of individual components, which will reveal the aggregated structure of the modelled reality at a lower hierarchical level.

For instance, the component representing the pump of the right ventricle is connected to the elastic compartment of the right atrium and the elastic compartment of the pulmonary arteries by means of two causal connectors (distributing the blood flow and blood pressure). Causal signal control inputs are connected to it from the “organ-BloodFlowSignals” bus. The “inside” of the component is shown in Fig. 32.

The heart is a pulsation pump that first draws blood from the atria into the ventricles during “diastole” – at the end of diastole, the volume of blood in the ventricle equals the end-diastolic volume (EDV). After the end of diastole, the valves between the atrium and the ventricle close and the ventricle starts contracting during “systole”. The appropriate valves open and the right ventricle starts pumping blood out to the pulmonary artery (the left ventricle to the aorta). At the end of systole, the valves between the ventricle and the pulmonary artery in the right ventricle (and between the ventricle and the aorta in the left ventricle) close – the volume of blood in the ventricle at the end of systole is called end-systolic volume (ESV). The ventricle muscles relax, the pressure gradient between the atrium and the ventricle opens the atrioventricular valves and diastole begins again.

In the ventricle model, the end-diastolic volume (EDV) is calculated in a “diastole” component and the end-systolic volume in a “systole” component. Modelica allows not only designing of the graphic form of icons representing the individual components but also animating the icons (to improve clarity). In the given example, both components have animated curves during simulation, which represent the relation between pressure in the ventricle and the values of ESV and EDV. A dot on the curves represents the current value of EDV/ESV. Blood pressure in the ventricle is derived from the value of the supply pressure in the atrium (this gets to the component from “q_in” by means of an acausal connector) and the value of the external pressure in the pericardium – this gets to the ventricle from the signal bus

**Figure 32.** The “inside” of an instance of the right ventricle pump (component “rightVentricle”) from Fig. 30. The ventricle is modelled as a continuous pump with variable internal volume.
“BloodFlowSignals” by means of a causal connector, “Pericardium_Pressure”. In the “systole” component, blood pressure in the right atrium at the end of systole is derived from the value of the counter-pressure in the pulmonary artery (or pressure in the aorta in the left ventricle) – by means of an acausal connector, “q_out”, and the value of the external pressure in the pericardium (by means of the causal connector “Pericardium_Pressure”). During systole, the dependency of the ESV value on the end-systolic pressure is also affected by the stimulation (or blocking) of “beta receptors”, which results in changes in the contractile power of the heart muscle. A detailed description of equations that describe this dependency is contained in “BetaReceptorsActivityFactor”, a component whose output is the causal input for the “systole” component.

The ventricle model in Fig. 32 is not expressed as a pulsation pump but rather as a continuous pump with variable internal volume. We do not model pumping “beat by beat” but by the average cardiac output per minute. The systolic volume is calculated first (in the component “StrokeVolume”), as the difference between the end-diastolic (EDV) and end-systolic (ESV) volumes. The value of the blood flow per minute is calculated (in the “BloodFlow” multiplier) from the systolic volume multiplied by the heart rate. The value of the heart rate (“HeartVentricleRate”) comes from the outside, from the “bloodFlowSignals” bus. The average volume of blood in the ventricle is estimated as the arithmetic mean (“Vol_SteadyState”) of the maximum heart charge in diastole (EDV) and heart volume at the end of systole (ESV).

Figure 33. A ventricle model with valves, which generates a pulsating blood flow beat by beat. It has the same outer interface for interconnecting into the model of a higher hierarchical level as the pump model.
The ventricle is represented (by the “ventricle” component) as a continuous pump that has variable internal volume (the component is an instance of the model from Fig. 30). The pump is connected to the blood circulation by means of acausal connectors (“q_in” and “q_out”). It receives the calculated value of the cardiac output (Blood_inflow) and the required average value of the pump’s internal volume (“Volume_SteadyState”) by means of two causal connectors.

The model of the heart approximated as a continuous pump is sufficient (and sufficiently quick) for a number of applications in medical simulators. However, if we wish to model e.g. various valve defects, we have to use a more detailed model, describing the behaviour of the ventricle beat by beat.

Replacing a simpler component with a more complex component does not have to mean reworking the entire model. Model notation in Modelica allows a very elegant exchange of components as different variants of classes with the same interface.

For example, it is possible to exchange the instances of the left and right ventricle models (“rightVentricle” and “leftVentricle” components) inside the blood circulation subsystem model (see Fig. 31): Instead of the continuous pump model of the ventricles (Fig. 32), we can insert instances of a more complex model into the diagram, generating blood flow beat by beat. We just have to cast the instances of the left and right ventricles.

The basis of the ventricle model with valves that generates a pulsating blood flow beat by beat (Fig. 33) is an elastic compartment (“ventricle”), which has a generated oscillating value of compliance (unlike the elastic compartment used in the blood vessels). The frequency of the oscillations is determined by the number of heartbeats.
per minute. The shape of a single compliance change period (the “curve” component) expresses the properties of the heart muscle. The amplitude is affected by the stimulation and blocking of the beta receptors. At last, the direction and rate of blood flow in the ventricle is derived automatically from the properties of the valve components (“valve1” and “valve2” components) and from the pressure gradients.

A simple valve model can be represented as an analogy of a series connection of an ideal diode with a resistor. An alternative (more complex) model of the valves will allow the modelling of various valve defects.

By exchanging components of different complexity with the same interface, we can create model instances of different complexity as needed for their application use. Modelica supports the possibility to exchange individual components by allowing defining interfaces with a variable number of control (input, causal) signals. Depending on the number of control signals, components may be more complex or, conversely, simplified, and their function may be tested when connected to a model of a higher hierarchical level. This major advantage can be used efficiently not only when debugging complex models but also when identifying a model from experimental data.

Making use of the hierarchy and component structure of models is very important in Modelica (see Fig. 34). For the model construction architecture, it is advisable to follow the rule stating that the structure of a component should always fit in a single screen. A complex tangle of connections is not the sign of a good design and calls for trouble.

The purpose of this chapter was not to describe the physiology of blood circulation. We just wanted to use the rather detailed description of the structure of some components to illustrate how acausal modelling tools allow the creation of richly hierarchically structured, easily modifiable, “self-documenting” models.

When modelling extensive systems, such as the models of interconnected physiological regulations as a basis for medical simulators, the acausal modelling environment of the Modelica language is a great help.

**From Simulink to Modelica**

It is simply amazing how fast the new Modelica simulation language adopted various commercial development environments. Only recently, just two commercial implementations of this language existed (Dymola from Dynasim and MathModelica from Mathcore), today (February 2010), the Modelica language is already used by simulation environments LMS Imagine.Lab AmeSim from the LMS company (http://www.lmsintl.com), MapleSim from Maplesoft (http://www.maplesoft.com/), Mosi-lab from the Frauenhofer company (http://www.fraunhofer.de) and SimulationX from ITI (http://www.itl.de).

Modelica is being used more and more in industrial applications. This modern simulation language is used by large corporations such as Siemens, ABB and EDF. Well-know automakers, such as AUDI, BMW, Daimler, Ford, Toyota, VW, use Modelica to design economic cars and air-conditioning units. The advancement of the development environments and technologies that use the Modelica language and the development of relevant application libraries is a part of the European research projects EUROSYSLIB, MODELISAR and OPENPROD financed by a total sum of EUR 54 million (see http://www.modelica.org/).

However, Modelica is still not used as much in biomedical applications. The vast majority of biomedical simulation applications are still done in casual, block-oriented environments. These include referencing database development environments for biomedical models (such as the JSIM language http://physiome.org/).
A frequently used environment in biology and medicine is Matlab/Simulink – monographs dedicated to biomedicine models are usually equipped with additional software used in this environment, but so far without the use of new acasual or non-casual Simulink libraries, such as Wallish et al. 2008; Logan & Wolesensky, 2009; Oomnes et al. 2009.

However, already in 2006, Cellier and Nebot pointed out the benefits of Modelica, when used for clear implementation of physiological systems descriptions and interpretations. The classic McLeod’s circulation system model was implemented by PHYSBE (PHYSiological Simulation Benchmark Experiment) (McLeod, 1966; McLeod, 1967; McLeod, 1970). The difference is clearly seen, if we compare the Cellier model implementation (Cellier & Nebot, 2006) with the freely downloadable version of the PHYSBE model implementation in Simulink http://www.mathworks.com/products/demos/simulink/physbe/.

Haas and Burnham, in their recently published monograph, pointed out the benefits and large potential of the Modelica language used for modeling medically adaptive regulatory systems (Haas & Burnham, 2008). The most recent, Brudgád a spol. publication (2009) talks about work on the implementation of the SBLM – http://sbml.org/, in the Modelica language. This would enable us in the future, to simply
run models, whose structure is described in the SBLM language, on development platforms, based on the Modelica language. However, acasual models may be created in Simulink today as well, by using new acasual libraries (Simscape and others).

We have been using Matlab and Simulink for years to create and develop models of physiological systems (Kofránek et al. 2001, 2002, 2007) and have also been developing the relevant application Simulink library – the Physiolibrary http://physiome.cz/simchips). We have also developed the relevant software tools that simplify the transfer of models implemented in Simulink over to development environments (ControlWeb and Microsoft .NET), where we create our own tutorial and education simulators (Kofránek et al. 2008). Our development team gained priceless experience in previous years working with the Matlab/Simulink development environment made by the renown company MathWorks. On the other hand, we were also
attracted by the new development environments using the Modelica language. We were facing a decision whether to continue with the development process of physiological system models in Simulink (using new acausal libraries), or to make a radical decision and switch to the new Modelica language platform.

Our decision was affected by our efforts to implement a large model made by Guyton’s coworkers and students (Hester et al. 2008). Their Quantitative Human Physiology (QHP) model is an extension of a tutorial simulator called the Quantitative Circulatory Physiology (QCP) (Abram et al. 2007).

The QHP model contains more than 4,000 variables and at the present time, it probably represents the largest and most extensive model of physiological regulations. It enables the user to simulate a wide range of pathological stages and statuses, including the effects of the relevant applied therapy.

Compared with the previous QCP simulator, whose mathematical background is hidden from the user in its source code written in C++, the QHP simulator uses a different approach. The QHP authors decided to separate the simulator implementation and description of the model quotations, in order to make the structure of the model more clear and apparent for the larger scientific community.

In 1985 the architect of this model, Thomas Coleman, had already created a special language used to write the model structure, as well as the element definitions into the simulator user interface. The language is based on modified XML notation. Model is then written by using XML files. A special converter/decoder (DESolver) converts XML files into executable simulator code.

A detailed description of this language and DESolver converter, as well as the relevant educational tutorial, is freely accessible on the web page of the University of Mississippi (Fig. 35). The new QHP model is writ-
ten in the XML language as well. Its structure with all details may be found at (http://physiology.umc.edu/themodelingworkshop), published as an open source. Therefore, the user can modify this model as he wishes. However, the model description has been divided into more than 2,833 XML files in 772 directories, from which the special solver creates and executes the simulator (Fig. 36).

The entire structure of the model and following links and references are not easily identifiable. That is why the international research and development team in its SAPHIR project (System Approach for Physiological Integration of Renal, cardiac and respiratory control) decided to use the old Guyton models from 1972 (Guyton, Coleman & Grander, 1972) and the Ikeda model from 1979 (Ikeda, Marumo & Shirsataka, 1979) for the creation of its new and extensive model of physiological functions instead of the freely available QHP model. The source codes of the QHP model appeared unclear or hard-to-understand to those involved in this project (Thomas et al. 2008).

The extensive QHP model is still at the testing, modification and expansion stage. We have been able to agree on a long-term cooperation with the main architect of this simulator, Thomas Coleman, as well as with other co-authors from the University of Mississippi, focusing on the future development of this model. We have been able to create a special software tool called QHPView (Fig. 37), which is able to create a clear and legible overview of mathematical relations and connections from thousands of source codes.

We are offering this tool as an open source on the QHP web page at (http://physiology.umc.edu/themodelingworkshop/).

First, we tried to implement the QHP model in the Simulink environment. The model contains a wide range of relations that offer solutions for implicit quotations. That is why the implementation of this block-oriented model (outputs from one block are used as inputs for the next blocks) is very difficult and as the implementation got more and more complex, the transparency of this model went down quickly. The use of new acausal libraries in this complex model proved to be problematic and the transparency of the model improved only a little bit. Therefore, we decided to stop using the Simulink implementation and began to implement the Modelica language (using the Dymola environment (http://www.3ds.com/products/catia/portfolio/dymola)).

Very quickly we discovered that the implementation of a large and extensive model in Modelica is much more effective than using acausal libraries in Simulink. When we compared the Simulink and Modelica implementations we also discovered a significant difference. Mainly due to the fact that the new acausal libraries are only acausal superstructure of Simulink and not an objectively oriented modeling language based on quotation, as the Modelica language is.

Therefore, if we compare the development environments based on the simulation language Modelica with the Matlab/Simulink development environments made by Mathworks, we may say the following:

- contrary to Simulink, the model implemented in Modelica much better reflects the essentials and base of the modeled reality and the simulation modes are more clear, readable and less prone to errors;
- the object architecture in Modelica enables the user to build and tweak models with an hierarchical structure gradually, while using reusable element libraries;
- contrary to Simulink (which is the industrial standard for Mathworks), Modelica is a normalized programming language and therefore, it may contain various commercial and non-commercial developing environments competing between each other. This language is used for specific problem solutions originating in various application fields.
(for commercial and non-commercial specialized libraries);

• in Modelica it is possible to combine casual (mostly signals) and acasual links non-invasively; and unlike in Simulink, it is also possible, (within interconnected blocks) to create algebraic loops fairly easily – the assembler in Modelica contains symbolic manipulations on the background and therefore the disconnection of algebraic loops is the task for the development environment and not for the programmer.

The above specified reasons led us to use, as the main implementation tool for the model creation, the Modelica language and we also gradually stopped using the Matlab/Simulink environment, (Kofránek, Mateják, & Privitzer, 2008).

As far as the creation of application libraries and tools used in the Modelica development environments is concerned, we are involved in international cooperation.

The mutual efforts of 12 companies and 9 universities grouped together in the so-called Open Modelica Source Consortium, contributed to the development of the Open Modelica environment, distributable as an open source at (Open Modelica Source Consortium – see http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html). Our development team cooperates with Creative Connections s.r.o., which is a member of this consortium (see http://www.creativeconnections.cz/). We are currently developing a tool which would enable us to generate a source code/text in the C# language from a model created and debugged in Modelica.

QHP in a modeling coat
The implementation of the QHP model clearly shows the benefits of the model creation process when done in the Modelica language.

If we compare the complex structure of the QHP model by using the visualization option in QHPView (Fig. 37) with examples of implementations done in the simulation language Modelica, shown in previous pictures 26–34, we can see that the acasual implementation done in Modelica creates a transparent and legible model structure and therefore offers easier model modifications.

The QHP model implemented in Modelica is being currently modified and extended. Modifications and extensions of QHP were partially taken from our original model Golem (Kofránek et al. 2001) and further modified according to newest findings and experiences.

Our modifications are mainly extensions, which improve the usability of the model during the modeling of difficult breakdowns in acidobasic (acid-based), ionic, volume and osmotic homeostasis of inner environments, which is very important for urgent medicinal statuses.

Our modification of the QHP model is based mainly on the process of re-programming the acidobasic subsystem balance, which is based in the original QHP on the so-called Stewart acidobasic balance theory. Simply put, the so-called “modern approach” of Stewart (Stewart, 1983) and his followers (e.g. Sirker et al. 2001; Fencl et al. 2000) explaining breakdowns in the acidobasic balance, uses mathematical relations calculating the concentration of hydrogen ions $[H^+]$ from partial pressure CO2 in plasma ($p$CO2), total concentration ($[BFT]$), weak (partially dissociated) acids ($[HBU]$) and their base ($[BFT]$), where $[BFT] = [BFT] + [HBU]$ and from the difference between the concentration of fully dissociated cations and fully dissociated anions in SID (strong ion difference):

$$[H^+] = \text{Function} (p\text{CO2, SID, BFT})$$

The problem of this approach is that the precision of acidobasic calculations in the model depends on the precision of the SID calculation, that is the difference between the concentration of fully dissociated cations (that is mainly sodium and potassium)
and fully dissociated anions (mostly chlorides). Imprecision that is created during the modeling of sodium, potassium and chlorides intake and excretion are transferred and reflected by the imprecision in the modeling process of the acidobasic status. Even though Coleman et al. (2008), significantly improved the modeling of reception and excretion of sodium, potassium and chlorides in kidneys in his QHP model, if we model a long-term status (when nothing is happening with the virtual patient), the virtual patient (in the current model version) has a tendency to fall into slight and steady metabolic acidosis after one month of the simulated time.

Our evaluative approach towards the modeling and evaluation of breakdowns in acidobasic balance (Kofránek, 1980; Kofránek et al. 2007; Kofránek, 2009) is based on the modeling and evaluation of two flows – the creation and excretion of CO2 and the creation and excretion of strong acids, connected through the purification systems of each part of the bodily fluids. This approach, according to our opinion, better explains the physiological causality of acidobasic regulations, rather than direct modeling of acidobasic breakdowns through the balancing of accompanying electrolytes. Besides, the fidelity and truthfulness of the modeling process is getting better; mainly in mixed (acidobasic and electrolyte) breakdowns in inner environments.
Another important modification of the QHP, is the fact that the model was extended by adding the dependency of the potassium flow on the intake of glucose as a result of insulin, which enables us to model (besides other things), the influence of potassium solution infusions with insulin and glucoses, which are distributed in acute medicine for treating potassium depletions.

We have been using this “balancing and evaluation” approach towards the modeling of acidobasic balance in our old “Golem” simulator (Kofránek et al. 2001). The extended QHP model serves as the base for the educational simulator “eGolem”, used in medical tutoring in clinical physiology of urgent statuses, which is being currently developed under the research project of MŠMT No. 2C067031. On the webpage of this project you may find the updated and current structure of our implementation of the QHP model (http://patf-biokyb.lf1.cuni.cz/wiki/projekty/e-golem).

From a model to the simulator

A simulation model, implemented in the most sophisticated development environment, cannot be used as an education aid alone. It is the implementation of the formalized description of the modeled reality that enables testing of the behavior of the mathematic model during various input values and the search for model quotations and parameters, which within the established precision range, can ensure the sufficient compatibility of the behavior of the model with the modeled system (model identification).

Even after this goal is reached, there is still a long road ahead from the identified model to the educational or tutorial simulator (see Fig. 38).

The creation of a multimedia educational and tutorial simulator is very demanding developmental work requiring a combination of the ideas and experiences of all the teachers who create the script of the tutorial program, the creativity of art designers who create the interactive multimedia components, as well as the effort of programmers who create the necessary user interface and put together the final masterpiece and its final shape.

Thanks to advances in software technologies, we now have new tools available used not only for the creation of more effective simulation models but also tools that simplify the creation of own simulators equipped with attractive user interface graphics.

The skeleton of simulation application – the script

Simulation technologies have improved a great deal and have become more effective, while prices for equipment and software keep falling. The development of the Internet, virtual 3D environments such as the already mentioned software Second Life, or even the increasing range of medical simulators using robotized dummies, representing the patient, opened new options and potential for the medical education process.

The use of a medical simulator during studies and education is becoming more and more popular and greatly utilized (mainly in the USA and Israel).

From the pedagogical point of view, an education process aided with a simulator is a very effective approach, but it is also a very demanding and challenging task for the teacher. Therefore, the most complex or sophisticated simulator equipped with the most attractive interface, is not guaranteed to be the most effective way of learning. Pedagogical efficiency depends on the teacher, who must have a clear vision and know what is the most effective and most suitable way to use simulation model in the class.

The more complex the simulator, the clearer the picture the teacher must have in his mind, being sure of how the class should look and what types of simulation games he wants to use. This is also confirmed by
our practical experiences with the use of complex simulators during classes (e.g. with Golem or QHP simulators). We have learned that a user interface offering many complex options or previews of hundreds of variables and parameters, takes away the concentration of students. Without clear pedagogical guidance pointing out what to look at or look for during the relevant simulation game while working with a complex simulator and without knowing how to interpret achieved results, the use of simulators is very ineffective.

However, from the pedagogical point of view, it is necessary to think in advance about how to use the simulator, before we actually begin with the creation. This is mostly true if we want to create multimedia and interactive educational programmes, available through the internet and using simulation games helping students understand and practice the subject better. The key for success is a good script.

The first person the success depends on, is an experienced teacher, who must have a very clear picture in his mind as to how he wants to explain the problem to his students, how the multimedia application should work, where they may be used and how to use the simulation model to help him explain the subject better.

The skeleton (the backbone) of educational application is the script. The base is usually studying text – script, chapter in a textbook, etc. However, during the creation process of multimedia tutorial application, we have to imagine how the tutorial program will be displayed on the screen, how each screen should follow after another (the sequence), what the graphical design will be like, where interactive elements will be placed, where the audio input may be, how each animation will look, where the simulation model will be inserted, where the knowledge test will be inserted, how it will be evaluated and what reactions on the test results will be required.

Figure 39. The pedagogue task – the role of the script creator in the development process of tutorial simulators is very important. The script author must together with the model creator precisely define how the model will be used in the tutorial application and how the user interface will look. They must prepare sufficient materials and specify the graphical requirements of all needed graphical elements and animations and present them before the art designer. Testing of the created simulator in class is very important as it usually opens new ideas and points out improvements and modifications.
The final shape of graphical elements is completed by a professional art designer. Therefore, a **good communication between pedagogical expert/teacher – the script creator and the designer is necessary.** The teacher doesn’t need to draw perfectly, but he has to have a clear and well-planned image in his mind and be able to explain to the designer what he wants from him. The biggest roadblock from the beginning was the constant need to redraw already drawn animations, usually due to the pedagogue’s/teacher’s fault, as he did not have a clear vision or image as to how to create the multimedia element in the script. Therefore, it is well-worth paying attention to careful planning and preparation before the actual beginning of the project (Fig. 39).

During the creation process of the tutorial application script, we found it very useful to use a procedure that is applied during a regular animation film making process – that is to draw (best together with an artist) a pictorial script, a so-called “Storyboard” – an approximate sequence of each screen and then by using a regular text editor, write the relevant commentary (or a link reference pointing to the relevant part of the text) underneath.

Interactive multimedia programs are not scripts re-written and converted into a computer format. It is not a linear sequence of texts, audio sounds and moving pictures, as a typical animated movie is. The significant feature of an educational and tutorial program is **its interactivity** and the possibility of branching and mutually interconnecting individual parts. To remake or transform text and picture scripts into a branching script linked with hypertext links of the accompanying interactive program is not easy at all.

One of the methodological problems that we had to solve during the creation of our scripts for tutorial application, was the problem how to display the structure of the tutorial program in the script, including interpretations, interaction with the user, program, branching, etc. The easiest way is to use a text or picture editor and by using regular block diagrams or structure diagrams, to describe the applicable branching, decision blocks, etc. with the relevant links on text pages and other relevant pictures stored in other files.

During the script writing, we also found using the abilities of modern text editors to create the required hypertext links helpful – which already gives the script some characteristics of its future interactivity.

We also tried using the **Adobe Captivate** tool to write the script for the tutorial application, see (http://www.adobe.com/cz/products/captivate/), which enabled us to create professional looking e-learning contents with advanced interactivity, without the need to have superb programming knowledge. We also found out that in order to write the script in the form of an interactively branched storyboard, this tool is too complicated. On the contrary, we learned that in order to share continuously created scripts between members of the development team, using simple tools is sufficient.

To specify the requirements for the creation of the graphical elements given to designers, and for monitoring the results of their work and for monitoring other results, **Microsoft OneNote** was sufficient.

Modern interactive tutorial program is not an animated film converted into a computer form – the best advantage of interactivity offered by a computer, is the option to use **simulator**, which through the use of simulation games enables the user to explain the problem in virtual reality. The script of the tutorial program must take this advantage into consideration. The author must answer these following questions – what type of simulation experiments should be offered with the simulation model, what the user interface of the simulation game will be like, and finally what are the requirements for the simulation model running in the background.

Therefore, it is necessary for the “scriptwriter” to communicate with the model creator and to know the structure of the model,
so he may propose possible modifications and be able to explain the specifications that the model should have and comply with. A key factor is also pedagogical experience. Sometimes elements or issues that appear simple and easy-to-follow during the development of tutorial application, may become difficult and hard-to-follow issues once integrated into the pedagogical process. Besides that, during the use of simulation games in the education process, the modification of either the user interface or the simulation model on the application background is necessary.

It is also necessary to create script proposals in close cooperation with pedagogical experts and based on pedagogical experiences. Therefore, we found it very helpful to first try the simulation application “on students” in class and based on the results, create the explanation text, modify the simulator and propose the final form of the user interface for the production version of the relevant internet-based e-learning application.

**Muscles of tutorial simulation applications – interactive multimedia components**

To create the user interface of the tutorial simulator, it is rather impressive to present the simulator as a set of moving pictures, controlled by the simulation model. The controlled animations may graphically represent the meaning of numeric values – e.g. a schematic picture of a blood vessel may be extended or compressed, the heart may pulse quicker or slower, the lungs may breath deeper, the arm of a measuring apparatus may move and display a value of some variables of the model read from the simulation model running on the background, etc. On the other hand, we may enter various inputs into the simulation model (using various press buttons, buttons, levers and so on).

The graphic look/design is important as well, as it may determine how the tutorial and educational application will be accepted by potential users.

To achieve a professional final look of the tutorial simulator, it is necessary to have an artist create the animations – the results are much better than if animations are created by a programmer, even with graphical talent. However, the art designer must be able to use interactive graphics tools. But art designers with these skills are very hard to find and artists who are able to work with these tools are in demand and are usually highly paid members of professional teams that produce computer games, web portals and multimedia commercials and applications, etc.

To accomplish this, we had to shift a great deal of our attention to training and school instructors in order to make sure that they know how to use these tools. Therefore, we began to work closely with the Václav Hollar College of Arts, where we opened an interactive graphics laboratory, as a detached workplace of the Charles University.

We spent a great amount of time trying to teach professional artists to work with interactive animation development tools, such as Adobe Flash, Microsoft Expression Blend, etc., or to work with 3D graphics development tools (e.g. Adobe Premiere), then to integrate it all into applications accessible through the internet (e.g. Adobe Flex) and finally teach them about the basis of program controlled interactive animations and about the creation of interactive web pages. Our efforts were rewarded. Graphic artists stopped being shy in front of the computer and quickly understood that “a digital brush” is just another creative tool, giving them a way to express themselves and that mastering a digital brush gives them an opportunity to succeed in the professional field.

We have also helped with the establishment of a “higher professional school”, where interactive graphics are taught in three-year courses. Workers of our Bio-cybernetics department and workers of the compu-
ter support system classes, also teach here, (http://www.hollarka.cz/).

On one side, the artist must work with the pedagogue, who creates the script of the tutorial application and also with the programmer who makes sure that interactive animations behave as required (e.g. he interconnects interactive animations with the simulation model). Therefore, even the art designer must have basic knowledge of programming, so he may communicate with the programmer effectively.

In the past, we have used the Golem simulator for the simulator interface, (Kofránek et al. 2001) and the ControlWeb software environment, originally designed for visualization and control of industrial processes (see http://www.mii.cz/cat?id=1 &lang=409), which offers a wide range of preset elements – virtual instruments, which enabled us to create the user interface quickly and comfortably. However the quickly and comfortably created simula-

Figure 40. Visual interface of tutorial simulator demonstrating kidney functions, created in the ControlWeb environment. Outputs from the model are displayed on arm-gauges and at the same time they affect the shape of the inserted animated picture of the kidney glomeruli (artery diameters, arrow thickness and numeric value, etc.), created in Adobe Flash.

To create interactive multimedia components, connectable with the simulation model on the background, we use two tools (Fig. 41):

- The first one is Adobe Flash, were we can create animated interactive components, which may be programmed (and may be connected with the simulation model in our applications). The created components may be easily
played back in the internet browser (if the freely available Adobe Flash Player is installed) and on various operation system platforms. Also, less demanding numerical simulators, easily playable directly from the web page browser, may be created in Flash. We also used Flash components as the visual interface, communicating with the simulator core (through the ActiveX component) which were created in the ControlWeb and .NET environments.

- The second tool that we have started to use recreantly for the creation of simulator graphical components is the Microsoft Expression Blend development environment. This environment, while using the Microsoft .NET development environment, enables us to create applications that may be played or run directly in the internet browser, providing that the Microsoft Silverlight add-on is installed. The Microsoft Silverlight platform is capable of running numerically demanding applications with an interactive multimedia interface. This new Microsoft platform enables the user to distribute numerically demanding simulators over the internet, playable directly from the internet browser.

Adobe Flash Animation brush designed for artists

Flash has been developed over a long period of time. First, it was made by Macromedia and used only for the creation of animated pictures. At that time, we were creating tutorial animations in a different product, the Macromedia – Director (Fig. 42), which enabled us to control animations through scripts (Kofranek & Svačina, 2001).

During that time, the option to control animations through scripts became available for Flash as well, as the syntax got gradually richer and better. Starting with version 7 (sold and known as Macromedia Flash MX 2004), the Flash environment already contained an object control language (ActionScript) with syntax very similar to Java, which offers rather sophisticated and comfortable control of the visual interactive elements.

The huge success of Macromedia Flash is partly because its creators were able to successfully define an interface for artists and art designers (enabling them to create basic animation elements), as well as for programmers, who by using the above-mentioned objective language can “vitalize” and give interactivity to these components.

The basic component of the Flash application is a film/movie. Film may be divided into each scene, which may be played back in a programmed sequence or at random. Scenes are composed of sequences, which contain individual frames. Films may be linked – from each movie you may call a link from another movie and initiate playback of the movie. This is very useful in internet applications, when the first part of the animation is played and the other part of the movie is being downloaded in the background.

The creation of computer animations is based on the creation of classical animated films, where each frame is drawn on transparent foils and stocked above each other. Some frames may not be completely redrawn and they are only moved (e.g. the background is moved), while other frames must be redrawn completely or partially (e.g. just the moving figure, etc.) in order to create animation – moving pictures.

In Flash, (Fig. 43) each scene consists of several layers, which works similarly as the foils used in animated films.

Each frame or scene has several layers, where each picture element is stored. These visual elements may be drawn in each layer separately – Flash includes a powerful tool for the creation of vector pictures (vector pictures may also be imported from other external painting or drawing applications). Or the picture may be selected from a library and the required shape created. However, the sample shape may not only be represented by a still picture. You may select
a clip from a movie, (MovieClip), which in reality is an instant classic of a previously created film. For example, if we want to create a picture of a plane, we can select a rotating propeller from a library and insert it into one of the layers, as one element of the picture. A special type of movie clips are buttons. The shape and graphics may be designed (when the cursor is moved over the button or when the button is pressed), as well as the button action, behavior and function. The movie clip may have a rather complex structural hierarchy – the film that creates the clip may contain instances from other clips. For example, a MovieClip of a car may contain movie clips of spinning wheels. Each instance of a MovieClip has its parameters (coordinates specifying its location on the screen, size, colors, transparency, etc.), which may be changed dynamically in the program. Besides that, the MovieClip class offers many methods that may be used (e.g. a method which detects the collision between two instances of a MovieClip, etc.).

During the creation of a MovieClip we can also program specific methods, which may be recalled from its instances. We may also program the complex behavior of visual components. It is rather easy to create special MovieClips as real components and then set their parameters and properties in a special component editor and recall their methods while the clip is running. This enables manufactures to create (and distribute and sell) various visual (and non-visual) components and helped with the introduction of Flash in the artistic community.

The graphical and the programming part of the movie is created in the development environment. Then it may be tested or translated into a sub-language (in .swf format), which may be interpreted by using the freely available and downloadable interpreter (the so-called Flash player) and played back as an individually executable animation or it may be viewed in the internet browser (see Fig. 41).

Besides that, the created .swf file may be interpreted by using the special ActiveX...
component, which may be integrated into another program – e.g. into an application created in ControlWeb or in Microsoft Visual Studio. The important thing is that this component may exchange messages with the application, enabling us to comfortably control the behavior of the interactive application through another application. The application may also receive messages from the interactive animation, describing or referring to the user action/intervention.

The huge success of Flash caused that the Adobe company bought Macromedia and Flash became one of the integral parts in portfolios containing computer graphic tools made by this manufacturer.

Today, Flash components may be used in so-called RIA formats (Rich Internet Application) – a new generation of multiplatform web applications with superb complex user interface design, created with Adobe Flex or as a desktop application created with Adobe Air.

Figure 42. A multimedia tutorial program with interactive graphical animations created in Macromedia Director in 2000. At that time, Director offered stronger support for interactive programming than Flash. But Flash soon overcame this handicap and offered richer options of interactive controls than Director. Besides that, Flash offered a more intuitive user interface, resembling tools used for creation of animated films and became very popular among art designers. Therefore, we stopped using Director in 2000 and began using Flash for interactive animations.

Figure 43. Adobe Flash environment offers tools for vector picture painting. Each layer of movie frames (as seen on the picture) may be also inserted with a sample of movie clip, selected from a library. The behavior of each visual and non-visual component may be programmed in a special programming window.
The speed of the .swf file interpreter (Flash player) has improved and the ActionScript language can now be used to create the simulation core of tutorial simulators. The advantage of Flash tutorial and education applications (which may contain complex RIA applications, compiled in the Adobe Flex environment) is that these applications may be executed directly from the internet browser (providing that the relevant plug-in is installed) and run on all platforms.

We have created some tutorial simulators and multimedia interactive applications with simulation games in this environment (see Fig. 44).

In 2004, using Flash, we created (besides other applications) an interactive multimedia tutorial application called “Music as seen by physics and physiology. Tuning/tweaking Theory” (http://patf-biokyb.lf1.cuni.cz/~obdrzalek/ladeni.htm) for which we have received the “TECHFILM Laureate” reward at the TechFilm festival, (Obdržálek & Kofránek, 2004). Using Flash, we have also created a twenty-minute film called “The Historical Meeting” (Kofránek, 2006), dedicated to the history of relationship between the Czech Republic and Sweden during the Thirty-year war.

Our Atlas of physiology and pathology was also implemented using the Flash platform (see http://www.physiome.cz/atlas/index_en.html).

However, the Flash player environment is still an environment based on the interpretation of .swf flash files. For numerically demanding calculation used in more complex simulators we encounter certain...
**performance barriers.** For more complex simulators, the Adobe Flash environment is (so far) insufficient. **More complex simulators** are created in the **.NET environment** (and in the past they were created in ControlWeb). Flash components are integrated into these simulators by using the ActiveX component. But to bridge between the two different “worlds” of Adobe Flash and .NET, and to make sure that both operate in unison, hard programing work is involved.

**Microsoft Expression Blend – a tool for the creation of “graphical puppets“ for simulators**

However, the Microsoft .NET platform may be used directly for graphical applications (without the need to interconnect Adobe Flash components). Thanks to **WPF technology – Windows Presentation Foundation** (Sells & Griffins, 2007) we can use the .NET platform to create complex graphic components containing animations, vector graphics, 3D elements, etc. Graphical elements may be created similarly as in Adobe Flash but with potentially better options for controlling their behavior, than in Adobe Flash. Besides that, Microsoft swiftly reacted to the widely spread addition of Adobe Flash internet simulators and created its own **Silverlight** platform, which similarly as Flash Player, is capable of running complex applications combining text, vector and bitmap graphics, animations and videos from internet browser. The application runs as the primary application in the internet browser without the need to install it (the only required application to be installed is the Silverlight plug-in). Therefore, by installing a small component, Silverlight enables the user to interactively control applications in the majority of web browsers (Internet Explorer, Firefox, Safari) and on various hardware and software platforms. Now Windows and Mac operational systems are directly supported for the most popular browsers and fully compatible open source implementation is under development for Linux OS. Applications created for this platform use a significant part of the .NET framework, which is a part of the plug-in (and therefore these applications may handle quite complex calculations).

Silverlight is a platform which is capable of distributing simulators that run directly in the internet browser via the internet (and even on computers with various operating systems – the only requirement is installing the relevant add-on (plug-in)). The important parameter of **Silverlight** is that it **includes native support of animations** (Little, Beres, Hinkson, Rader & Croney, 2009). Therefore, animations are part of the application and it is not necessary to use an additional platform such as Adobe Flash for graphic layers.

![Figure 45. Key animation frames created in Microsoft Expression Blend are placed directly on the time axis. This makes the synchronization of animations with the audio track easier.](image-url)
The animation method is also more advanced. In the Adobe Flash environment, the animation is controlled by playing each frame in the preset speed (see Fig. 43 and Fig. 55). If all the frames are not rendered or loaded in time during the animation playback on a client computer, some frames are skipped over and the animation appears “jerky”. In Silverlight, the animation is driven directly by a time axis. That makes the playback smoother because the frame speed is progressively adjusted according to the resources available in the client computer – where the animation is played back.

Microsoft created a tool to create graphical elements and animations comfortably — the Microsoft Expression Blend. The graphical interface for Silverlight may also be created using this tool.

Microsoft Expression Blend offers an interface for art designers and programmers as well and works directly above the application created in Visual Studio .NET (Williams, 2008). That means that communication between the programmer and the art designer is much easier and design proposals don't need to be transferred over to the application project.

Key animation frames in Microsoft Expression Blend are not created by each frame (as in Adobe Flash) but according to the time axis (Fig. 45). The drawn animations may then be expressed or described as the object property. The size of the animated project may be set according to value derived from the animation time axis (Fig. 46). Therefore, by entering the value of the created property we may control the shape of the animated graphical element.

The created and controllable animated graphical object may be used as a component for another, more complex animated object and it may be controlled or its shape may be changed by changing or by setting the values of the applicable property. This way we can create an animated puppet, whose final shape depends on the current setting of its component values.

To simplify the communication between the art designer and the programmer who implements his own simulator and also between the art designer and the author of the tutorial application, we have created a software tool called Animtester (Kofránek 2009). Using this software, graphic designers may tweak and create these puppets, without the need of additional programming work (see Fig. 47–48). Animester as a component is inserted into the Microsoft Expression Blend development tool and enables the user to generate properties of the graphical element from the created
animations, controlled from outside, by using control elements (buttons and sliders). When the application is generated, the art designer and the author of the tutorial application (the pedagogue) may check and verify how the animated component will behave.

This enable the art designer to be shielded from the programming work details. And similarly, the author of the proposal of the tutorial application doesn’t need to focus on the implementation details of the graphical proposal and may communicate with the art designer easier and thus reach his vision and goal quicker.

The task of the programmer implementing his own simulator is to interconnect the generated graphical object with the simulation model on the background. The animated “puppets” created in this way and controlled via values that control their shape, may be directly connected to model outputs, without the need to add another programming „middle“/sub-layer for data propagation, as is necessary in Flash animations.

The use of graphical options in the Silverlight platform replaces and compensates greatly for the original approach using animations based on the Adobe Flash platform. Therefore, during the creation of animations as a visual interface for simulators, we do not need the Adobe Flash platform, which may be fully replaced with the new animation tools from Microsoft.

The brain of the tutorial and education application – the simulation model

Implementation of simulation models in the education and tutorial program is not a simple issue.

To create simulation models, we use special development tools, designed for tweaking, tuning and verification of simulation models (Matlab/Simulink or acasual development using the Modelica language), which we discussed in previous chapters.

Debugged models must be converted from the development environment where they were created, debugged and verified into the environment where the tutorial simulator itself is being created.

This may be done manually for simple models – as is often done in purely Flash tutorial simulators, where the development environment for the creation of simulators is Adobe Flash only.

However, for more complex models we created software tools which automate this work for us (see Fig. 49). In the Golem simulator, implemented in the ControlWeb environment, the model was represented as a controller of virtual measuring/control card (see Fig. 50). To automate the transfer or conversion of the simulation model from Matlab/Simulink we have created a generator, which creates the controller source text in the C language directly from the Simulink model. (Kofránek et al. 2002).

And to simplify the creation of simulators done in Visual Studio .NET (that is to eliminate “manual” programming of the debugged simulation model in Visual Studio .NET) we developed a special software tool (Kofránek et al. 2005; Stodulka et al. 2007), which automatically generates from the Simulink simulation model in a component form useable in the .NET environment. The output in Modelica is the generated simulator program in C++. If we are ok with a simulator that needs to always be installed in the client computer then the program in C++ is sufficient. But, if we want to make use of the new options available in the .NET environment, which enables the user to create applications in Silverlight and run in the internet browser, then we have to create a tool which will generate the model source text in C# from Modelica, which is also our current goal at the international Open Modelica Source Consortium as mentioned earlier – (see http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html).
The body of the tutorial simulator – installed program or web application

The creation of a tutorial application is quite demanding programming work, based on the creation of the simulation core of the developed application (unless this core was already automatically generated from the applicable simulation development tool) and its interconnection with graphical elements of the visual user interface (see Fig. 51).

For simple tutorial applications with simple models we can use Flash player with the ActionScript language, which we use to program the simulator itself and the entire application may be run in the internet browser.

However, this is not enough for more complex applications.

In the past, we were creating simulators in the ControlWeb development environment, made by the Czech company Moravské přístroje (Kofránek et al. 2001; Kofranek et al 2002). The created application had to be installed in the client computer or (in case of web-distributed applications), at least the ControlWeb runtime environment had to be installed.

For the development of simulators today, we use the Microsoft .NET platform and for programming work, the Microsoft Visual Studio .NET programming environment, which offers great options for programming. We can also use the graphical component of the user interface created in Adobe Flash, which we can interconnect with (via ActiveX) the core of the simulator, which is represented by the simulation model and graphical components may then behave as puppets controlled by the simulation model. This method was used during the realization of our Atlas of physiology and pathology, dedicated to the basic dynamic properties of physiological regulatory systems – http://physiome.cz/atlas/sim/RegulaceSys/ or the blood gas transfer tutorial simulator – http://physiome.cz/atlas/sim/BloodyMary_cs/.

The disadvantage of this approach is the necessity to install the program (offered through the internet interface) into the client computer. In such scenario the client needs to have the relevant installation rights applicable to the computer that he works
**Figure 48.** Animation of a beating heart. Outputs from the model affect the phases of the heart pulse, opening and closing of cardiac valves, etc. Auxiliary Animtest control elements are above the animation and enable the graphic designer to set and tweak each sub-animation. The graphical designer is completely shielded from the programming process. In the final simulator, the “control ropes and levers” are pulled by the simulation model, programmed on the background.
However, this is not the case in computer classes, where computers are protected from the installation of unwanted software, and the user must first ask the administrator for permission to install the educational program.

Therefore, it is desirable to be able to run and control even complex models directly from the web browser. This is possible if the entire simulator can be executed in the Silverlight environment, that is, if the entire core has been created in the form of a control code designed to be used in the .NET environment (in .NET assembly), and the graphic components are created in the Microsoft Expression Blend environment.

### Simulator structure – MVC architecture

If the architecture is more complex, the logic used to interconnect the visual user interface with the simulation model may be quite difficult. Therefore, it is better to insert a control layer between the visual element layer and the simulation model layer, which solves the communication logic between the user interface and the model.
and where the relevant context is stored. In literature (Collins, 1995; Leff & Rayfield, 2007) we may find the so-called MVC architecture (Model – View – Controller).

This setup is necessary mainly for more complex models and simulators, whose user interface is represented by many virtual instruments displayed on several interconnected screens. The advantages of this setup are seen mainly when applying model or user interface modifications (Fig. 52).

When designing the control layer, which connects the simulation model layer with the user interface, we found using interconnected status instruments very useful, which are able to memorize the relevant context of the model, as well as the context of the user interface.

For this reason, we have created a special software status tool called Statechart Editor, which enables us to visually design the interconnected status instruments, inter-
actively test their behavior and automatically generate their source code for use in the Microsoft .NET environment. This tool makes programming the links connecting the simulation model with visual objects of the user interface in the tutorial simulator possible.

**Interconnection of platforms used for the creation of models, simulators and animations**

When creating simulators, we are forced to work with three types of different software tools.

- Software tools for the creation and debugging of mathematical models, which will be the base of the simulator – Matlab/Simulink and recently with acasual tools using the Modelica language. It is beneficial and effective to create simulation models in this environment, but it is problematic to operate simulators in this environment.
- A software tool for the development of own simulator – mostly the Microsoft Visual Studio .NET environment is used. In the past we were using the ControlWeb development environment, made by the Czech company Moravské přístroje, mainly because it offers excellent options for the quick creation of a simulator user interface – however, this interface has an overly technical character. Simple models are implemented in the ActionScript language and therefore it is sufficient to use the Adobe Flash environment, added with Adobe Flex.
- Software tools for creation of interactive multimedia graphics – user interface for simulators. Here, we have been using Adobe Flash (formerly Macromedia Flash) for a long time. Interactive animations, which may be programmed
Figure 53. The original solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is a high-quality script, created by an experienced pedagogue. The creation of animated pictures is done by artists who create interactive animations in Adobe Flash. The core of simulators is the simulation model, created with special development tools, designed for the creation of simulation models. For a long time, we have been using Matlab/Simulink made by Mathworks for the development of models. The simulator development process is a demanding programming work. To make this task easier, we have developed special programmes that simplify the automatic transfer process of simulation models created in Matlab/Simulink over to ControlWeb or over to the Microsoft .NET environment.
Figure 54. New solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is still a high-quality script, created by an experienced pedagogue. The creation of animated pictures is done by artists who create interactive animations in Expression Blend. To create and test animations that will be controlled by the simulation model, art designers use the Animtester software tool, developed by us. The core of simulators is the simulation model, created in the Modelica simulation language environment. Within the project Open Modelica Source Consortium, we are in the process of creating a tool which will be able to generate the source text from Modelica in the C# language. This will enables us to generate a component from .NET used in the final application on the Silverlight platform, which will enable us to distribute the simulator as a web application, running in the internet browser (even on computers with various operating systems).
by using the special programming language ActionScript, may be created here. The important thing is that animations may communicate via the software (thanks to the above-mentioned programming option in the ActionScript language) with their surroundings, using ActiveX components. Today, we prefer the **Microsoft Expression Blend** development environment before Adobe, because we can create graphic components for the **Silverlight platform**.

Because we use different development tools for simulation models and different tools for the simulator, we had to make sure that the results are flexibly transferred from one development environment into the other one — that is for example, automated model transfer from the Matlab/Simulink environment over to the Visual Studio Microsoft .NET (in the past over to ControlWeb). These connection tools enabled us to develop and continuously update the mathematical model in the most suitable environment designed for mathematical model development and at the same time, to develop our own simulator in Visual Studio .NET (or in ControlWeb), without the need to “manually” reprogram the mathematical model.
Also, these tools made multi-disciplinary cooperation between members of the solution team easy – system analysts creating mathematical models and programmers implementing the simulator (see Fig. 53). On the other hand, this method required working in three software environments and during each innovation we had to innovate the relevant connection tools.

From the user point of view, simulators are best distributed via the relevant web interface, which may be used also for the relevant interactive interpretation. Web based interpretation application may be easily interconnected with simple models implemented directly into ActionScript on a flash animation background (using this method we have created, for example, the tutorial

**Figure 57.** The Atlas of physiology and pathology combines interactive interpretations with sound, animations and simulation games. It has been created in Czech and (gradually) in an English version as well. It is freely available at: www.physiome.cz/atlas.

More complex models such as the complex model of blood gas transfer, (http://physiome.cz/atlas/sim/BloodyMary_cs/) required before the actual installation execution of the model on the client computer (and also the installation of .NET platform, which if not installed, is automatically downloaded from the Microsoft server).

The program installation process requires the user to have the relevant administration rights. Besides that, the model which runs as an independent application is connected indirectly with the web interface, where the multimedia interpretation is realized. This problem solves our new technology of simulator creation, using the Silverlight platform (see Fig. 54). Graphical elements are created in the Microsoft Expression Blend environment.

But the simulation core needs to be realized as controlled code in the .NET environment – this should be ensured by our newly developed application Modelica .NET, which will be able to generate model code in C#.

To design the inner logic of the application we use hierarchical status instruments which are able to memorize the relevant context of the model, as well as the context of the user interface. The Statecharts editor environment, developed by us, enables the user to graphically design the instruments, to generate their code and tweak/debug them.

The benefit is that the graphical interactive elements and the simulation core are created on one platform – therefore the need to bridge between .NET and Adobe Flash, by using ActiveX components is eliminated.

The simulator must be easily combined with the interpretation chapter. The final application (the simulator and the interpretation chapter) may be realized as a web application executable directly in the web browser, without the need to install it in the client computer. It may run on various operation systems – the only requirement is to have the Silverlight plug-in installed in the relevant web browser.

**Packaging of simulation games into multimedia interpretation**

A simulator without an interpretation part requires an experienced pedagogue during use. Therefore, it is beneficial to combine simulators with explanatory lectures.

Interactive programmes using virtual reality models offer great tutorial and educational benefits, because they combine tutorial text with animated pictures and with simulation games and therefore, better demonstrate and present the studied problem.

Our technology includes simulation games as a part of the e-learning multimedia tutorial lesson, based on a script created by an experienced teacher. The teacher or pedagogue compiles and proposes the wording of the text, as well as the shape of the pictures and animations.

Animations are designed by artists, closely cooperating with the pedagogue working in Adobe Flash and designing them for Flash Player, operating in the internet browser or (newer technology) the Microsoft Expression Blend platform in Silverlight.

The text is then read and synchronized with each animation and with simulation game references and links. Components are then compiled into study lessons.

However, to **synchronize animation with audio in Adobe Flash** is not that easy. Animations in Adobe Flash are created as in regular animated film – by each frame (see Fig. 55) and therefore, the timing sequence of the visualization depends on the time, when the “playback head” (at the preset playback speed), reaches the relevant scene/frame.

Audio track may be stored within the movie frame layer. When the synchronization process is done and animations with sound are played back, it is necessary to make sure (by using a command in ActionScript) that the played clip, playing the relevant anima-
tion, starts playing at the precise and correct time, when the playback head reaches its position.

To make the synchronization process easier, we have created a special tool called the PlayDirector – an element in the library, inserted into the created flash movie clip (Fig. 56). When the clip is played back in Flash Player, the audio tracks may be assigned interactive stoppers, according to which data are generated and used in the inserted script, which will make sure that the relevant animations will begin at the correct time.

To compile multimedia tutorial lessons created in Adobe Flash, we are using the Adobe Presenter development tool, supplied as software environment in the Adobe Connect server (Adobe Presenter may be purchased separately today – see http://www.adobe.com/products/presenter/).

However, the synchronization of animations with sound is much easier in the Silverlight environment. Because the animation approach in Silverlight is based on graphical property changes of the graphic objects within a certain time (unlike in Adobe Flash), the synchronization of animations with audio sounds is linear and therefore, the Microsoft Expression Blend environment is sufficient for this job and no other tools needs to be used or created.

Simulation games on the Web

One of the solid results of our efforts is our internet Atlas of physiology and pathology, created as a multimedia tutorial aid, which through the internet and through the use of the simulation model helps the teacher to explain the functions of individual physiological systems, causes and breakdowns. See http://physiome.cz/atlas.

Our Atlas is a part of the MEFANET network (MEDical FACulties NETwork), collecting electronic study textbooks and texts of medical universities in the Czech Republic and Slovakia (http://www.mefanet.cz/). Currently, an integration of English texts “Teaching Resources” of the American physiological community into our Atlas is under way, see http://www.apsarchive.org/).

The Atlas combines interactive lectures and chapters and simulation games with models of physiological systems (see Fig. 57). During the creation of the model user interface, used as a base for simulation games, the atlas looked more like an atlas of animated pictures from the regular, printed Atlas of physiology (Silbernagl & Despopoulos, 2003, Czech publication 2004) or the printed Atlas of pathology (Silbernagl & Lang, 1998, Czech publication 2001), rather than abstract regulation schemes used during biomedical classes. However, contrary to the printed illustrations, pictures creating the multimedia user interface in simulators are „alive“ and interactive – changes in parameters or variables will change the picture look as well. Thanks to interactive illustrations we may create simulation games which, better than regular still pictures or simple animations, explains the dynamical relations in physiological systems and help students to understand the causes or reasons that are involved in the development of pathogeneses in various diseases.

The Atlas project is an open project – the results are freely accessible on the internet. It is created in a Czech and English version. It is an electronic tutorial aid and a part of the All educational text network. Interactive animations and simulation models, including their source codes are available to all who are interested. We would appreciate any cooperation and help from all who would like to be involved in the creation process.

Conclusion – from enthusiasm to technology and multidisciplinary cooperation

Far gone is the time when a handful of enthusiasts, excited about the new possibilities of personal computers sold during the eighties, were making their own tuto-
rial and education programmes. Computers are much more powerful today, the numeric and graphical possibilities of computers are enormous in comparison with the computers of the eighties, a huge network of high-speed internet surrounds the entire planet and represents huge potential in the modern education process.

Also development tools and the methods of the software creation process are much more powerful and advanced. At the same time, users of these software applications must be much more experienced and educated.

The creation of high-quality software capable of utilizing the huge potential offered through the information and communication technologies of today, depends on the enthusiasm and hard work of individuals. It is a demanding and complicated development process, involving all kinds of professionals and experts:

- experienced teachers, whose script is the base for high-quality tutorial application
- system analysts working with professionals and experts on the relevant field and who are responsible for the creation of simulation models used in simulation games;
- art designers who create the outer look and visual shape;
- information specialists (programmers) who will “stitch up” the application together into its final shape.

To make sure that the work of all kinds of specialist is efficient, it is necessary to have many interconnected development tools and methods which make the cooperation between all involved members easier and helps them overcome the differences and barriers in their fields. A great deal of hard work and effort needs to be put in to master these tools, but the final results are well worth it.

The tutorial software creation process is slowly becoming a blend and a combination of pedagogical experiences and the creativity of enthusiasts. It is mostly work for specialized teams using highly specialized development tools and it is beginning to look more and more like an engineering project.

References

25. Kofránek, J., Andrlík, M., Kripner, T., & Stodulka, P., “From Art to Industry: Development of Biomedical Simulators”, The IPSI BgD Transactions on Advanced Research, 1 #2(Special Issue on the Research with Elements of Multidisciplinary, Interdisciplinary,
26. Kofránek, J. (Director), Klucina, P. (Script), Kofránek, J. (Producer), Rejl, V. (Designer), and Obdržálek, J. (Music), Mötten i historien, Historica encounter, Dějinné setkání, Motion picture film. Bajt servis s.r.o. in cooperation with Charles University. Prague, 2006.

Acknowledgement
Work on the development of medical simulators is supported under the National Research Programme, Project No. 2C06031 and by Creative Connections s.r.o.
E-learning materials focused on cancer management and control are supported by grant “Addressing Cancer Disparities in Central and Eastern Europe” Bristol-Myers Squibb Foundation, 2009–2011 (Project: National Information System for the Assessment and Communication of Cancer Care Results and Quality in the Czech Republic).
Computer Methods and Programs in Biomedicine, 2011

The Atlas of physiology and pathophysiology: web-based multimedia enabled interactive simulations

Kofránek, Jiří; Matoušek, Stanislav; Rusz, Jan; Stodulka, Petr; Privitzer, Pavol and Mateják, Marek, Tribula Martin

Article in Press, doi:10.1016/j.cmpb.2010.12.007
The Atlas of Physiology and Pathophysiology: Web-based multimedia enabled interactive simulations

Jiri Kofranek\textsuperscript{a,}\textsuperscript{*}, Stanislav Matousek\textsuperscript{a}, Jan Rusz\textsuperscript{a,}\textsuperscript{b}, Petr Stodulka\textsuperscript{a}, Pavol Privitzer\textsuperscript{a}, Marek Matejak\textsuperscript{a}, Martin Tribula\textsuperscript{a}

\textsuperscript{a} Charles University in Prague, First Faculty of Medicine, Department of Pathological Physiology, U nemocnice 5, 128 53 Praha 2, Czech Republic

\textsuperscript{b} Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Circuit Theory, Technicka 2, 166 27 Praha 6, Czech Republic

\begin{abstract}
The paper is a presentation of the current state of development for the Atlas of Physiology and Pathophysiology (Atlas). Our main aim is to provide a novel interactive multimedia application that can be used for biomedical education where (a) simulations are combined with tutorials and (b) the presentation layer is simplified while the underlying complexity of the model is retained. The development of the Atlas required the cooperation of many professionals including teachers, system analysts, artists, and programmers. During the design of the Atlas, tools were developed that allow for component-based creation of simulation models, creation of interactive multimedia and their final coordination into a compact unit based on the given design. The Atlas is a freely available online application, which can help to explain the function of individual physiological systems and the causes and symptoms of their disorders.
\end{abstract}

\section{Introduction}

Over 350 years ago, the renowned Czech scholar and educator J.A. Comenius emphasised the importance of “schola ludus” or “playful school” as the best practice in educational effort [1]. Contemporary use of interactive simulations and serious computer games can be interpreted as bringing a whole new quality to this maxim [2–4]. The behaviour of individual physiological subsystems can be appreciated in serious games, both under normal conditions as well as in the presence of a medical disorder. Such simulators include models not only of individual physiological subsystems but also of their mutual connections into more complex units. The simulation models can also involve disconnecting the regulation circuits and separately studying the influence of individual physiological variables while the other variables are set to a chosen constant value (the so-called “ceteris paribus” principle).

Over time, a number of biomedical simulators have been developed. Coleman and Randall created a model named “Human,” intended primarily for educational purposes [5]. This model allowed for the simulation of numerous pathological conditions including cardiac and renal failure, hemorrhagic shock, and others, as well as for the effect of several therapeutic interventions such as infusion therapy, the effect of certain medication, blood transfusion, artificial pulmonary ventilation, and dialysis. Recently, Meyers and Doherty created a web-based version of the Coleman system [6]. Another complex simulator by Hester et al. [7] titled “Quantitative Human Physiology” and currently renamed as “HumMod” [8] represents probably the most complex extensive model of physiological functions at present time [7]. The
simulator is an extension of the original large circulatory sys-
tem simulator [9], and achieves an integrated connection of all
important physiological systems.

Kofranek et al. have also created a complex educational
simulator "Golem" based on an extensive model of integrated
physiological regulations [10]. The simulator was designed to
teach complex disorders of the internal environment [11]. Cur-
cently, Kofranek and Rusz are creating a complex circulatory
dynamics [12]. Complex integrative simulators of this type can
cast a general reflection on physiological regulation in patho-
physiology or pathogenesis of varied medical conditions and
syndromes.

However, our experience in using quantitative models
(e.g. of the Golem or HumMod type) shows that complex
control may impact negatively on the usability of the applica-
tion. The large numbers of input variables, as well as
the many options for observing the outputs, require a
quite thorough understanding of the very structure of the
simulator on the user's part. Thus, for better and deeper
comprehension, models should be accompanied by an expla-
nation of their use, preferably with interactive educational
multimedia.

In this paper, we present a novel web-based interac-
tive multimedia application titled "Atlas of Physiology and
Pathophysiology" (hereinafter the Atlas). The Atlas uses the
possibilities offered by the connection between interactive
multimedia and simulation models, combines the simulations
with tutorials, and simplifies the presentation layer while
keeping the complexity of the model beneath. It is conceived
as a multimedia instruction aid that should help to explain,
with visual means and using simulation models, the function
of individual physiological subsystems and the causes and
manifestation of their disorders. The Atlas thus combines an
explanation of the physiological subsystems using audio and
visual animation with interactive simulation (interactive for
presentation layer, simulation for underlying model); it can be

During the time of the Atlas's development, new technolo-
gies have emerged allowing for component-based creation of
simulation models, creation of interactive multimedia, and
their interconnection into a compact unit. The prerequisite for
development of the Atlas was also the creation of a number of
mathematical models of physiological systems and appropri-
ate tools which enable us to facilitate the design and sharing
of the multimedia interactive educational simulators.

The organisation of the rest of the paper is as follows.
In Section 2, we describe computational methods and the-
ory. Section 3 describes our software description; details
the development process, standardisation and sharing, and
the technology used. Section 4 presents a sample of typi-
cal interactive simulation. Section 5 describes the hardware
and software specifications, and Section 6 describes mode of
system availability. Finally, in Section 7, we present the con-
clusions and future plans.

2. Computational methods and theory

Several key points should be considered when creating web-
based multimedia-enabled interactive simulators, such as
those that are part of the Atlas of Physiology and Pathophysi-
ology (see Fig. 1).

1. The underlying mathematical model needs to be formu-
lated based on known physiological relationships. The
model (i.e. the set of equations that simulate the behaviour
of the underlying object) is often implemented based on
the verified models published in biomedical literature, but
it can also represent original theoretical scientific work. In
the past, simulation models were typically created in the
same environment as the simulator itself (e.g. in the lan-
guages Fortran, C++ or Java). Today, special modelling tools
can be used. Our team has been using the Matlab/Simulink
(The Mathworks Inc.) environment on a long-term basis.
We have created a Simulink library of formalised physio-
logical relationships, named Physiology Blockset (available
on http://www.physiome.cz/simchips). Recently, we have
started implementing and creating mathematical models in
an environment based on the language Modelica (The Modelica Association) [13]. An essential innovation intro-
duced by Modelica lies in its declarative and thus acasual
definition of models. Individual parts of the model are
described directly as equations and not as an algorithm
towards solving the equations. Modelica uses intercon-
ected components in which equations are defined.

2. The simulator itself is created based on the underlying
mathematical model and the teaching goals; it is more a
matter of programming rather than modelling work. In our
case, the simulator generally has a three layer architecture
known as MVC (model–view–controller) [14,15]: the lay-
ers include the user interface with interactive animations,
the control layer and the simulator core. The simulation
core is obtained by converting a debugged model from the
modelling tool (Simulink or modelling tool based on the
Modelica language) to the simulator development envi-
ronment (ActionScript, Microsoft Visual Studio, etc.). The
conversion can be done manually in the case of simpler
models, but manual conversion of more complex models
would be a tedious and error-prone job. We have created
two software tools to convert the debugged models auto-
matically (see Table 1). The control layer connects the
simulation core with the interactive animations of the user
interface and assures correct application logic. This layer
is not needed in simpler simulators. The user interface is
created in cooperation of the simulator programmer with a
designer and a teacher. It can evoke pictures from
image-based textbooks, such as the Atlas of Physiology [16]
or the Atlas of Pathophysiology [17].

3. In order for an educational simulator to have a profes-
sional appearance, a trained graphic designer should be
the author of the user-interface animations. Our team has
developed special tools that allow testing of the animation
properties and subsequent connection of these interactive
animations with the other model layers (e.g. Animate
made for Microsoft Expression Blend). Hence, the artist
is required to be proficient not only in standard technologies
(e.g. AdobeFlash or Microsoft Expression Blend), but also in
these testing tools. We have put substantial effort into the
training of our artists in both areas.
Fig. 1 – Workflow of a web-based simulator’s development and deployment. The simulation core is connected with interactive animation by a control layer. The model core is programmed manually or by means of automatic generation from a modelling tool (e.g. Matlab/Simulink or the Modelica-programming-language-based tool). Graphic components are created in Adobe Flash or Microsoft Expression Blend. Creating animations in Expression Blend offers the advantage of creating both the animations and the simulator using the same .NET platform.

4. An ideal means of educational material deployment has been the Internet. Uncomplicated user accessibility and relative ease of updates are among its advantages over portable electronic media. However, if a larger number of users become connected one-by-one to a more complicated simulation model, its placement on the server may later cause problems with the server’s performance. In this situation, it is more effective to use the computing power of the client computers for running the model. We therefore used a technology in which the simulator is automatically downloaded, transparently installed and run securely in a Sandbox.

3. System description

3.1. Development process

The creation of educational software capable of utilizing the potential offered through today’s information and communication technologies is a demanding and complicated...

Table 1 – Atlas of Physiology and Pathophysiology: technologies used in the development of interactive simulator.

<table>
<thead>
<tr>
<th>Final product</th>
<th>Modelling environment</th>
<th>Model conversion to simulator development tool</th>
<th>Simulator development tool</th>
<th>Animation development tool</th>
<th>Simulator deployment tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A simulator based on Adobe Flash</td>
<td>Simulink</td>
<td>Manual</td>
<td>Action script based (Adobe Flex, Adobe Flash)</td>
<td>Adobe Flash</td>
<td>Run in the Internet browser (Flash Player plugin needed)</td>
</tr>
<tr>
<td>2. A simulator based on .NET platform</td>
<td>Simulink</td>
<td>Automatic</td>
<td>Microsoft Visual Studio</td>
<td>Adobe Flash</td>
<td>Installed locally from the Internet</td>
</tr>
<tr>
<td>3. A simulator based on Silverlight</td>
<td>Modelica</td>
<td>Automatic</td>
<td>Microsoft Visual Studio</td>
<td>Microsoft Expression Blend, Animtester</td>
<td>Run in the Internet browser (Silverlight plugin needed)</td>
</tr>
</tbody>
</table>
development process, involving all kinds of professionals and experts:

- Experienced teachers, whose course-scripts are the base for high-quality tutorial application.
- System analysts working with professionals and experts in the relevant field, who are responsible for the creation of simulation models used in interactive educational simulations.
- Art designers who create the exterior look and visual shape.
- Information specialists (programmers) who will “stitch up” the application into its final shape.

To make sure that the work of all kinds of specialists is efficient, it is necessary to have specific interconnected development tools which make the cooperation between all involved members easier and help them overcome the differences and barriers in their fields.

3.2. Standardisation and sharing

The Atlas is a part of the MEFANET (Medical Faculties Network) structure, which secures the sharing of electronic educational content of Czech and Slovak medical faculties [18]. To facilitate sharing, the educational content had to be supplemented to involve metadata that describe the content to be shared.

There are number of standards for metadata forms that are related to e-learning objects. The main purpose is to enable transfer of e-learning content into other organisations or into other Learning Management Systems (LMS), modularise or reuse content in other e-learning courses, keep an account of versioning of content, relate content to other situation (repurposing) etc. In our case, the norm SCORM (Sharable Content Object Reference) is used for metadata description. The SCORM is created and supported by ADL (Advanced Distributed Learning) initiative [19]. The matter of this choice is given by the fact that educational portal of the Czech Faculty of Medicine uses LMS Adobe Connect which has implemented description of e-learning objects in the SCORM norm. The SCORM is not the only norm for description of e-learning content metadata; a review of contemporary technical standards enabling medical education content sharing can be found in Konstantinidis et al. [20].

The variability of possible metadata for description of e-learning objects (and mostly the missing of any description) leads to a lack of standardised content sharing mechanisms. As a result, considering the abundance of medical educational contents created by various academic institutions, it is not easy to retrieve a necessary learning object. The solution of the problem caused by the lack of standardised content sharing mechanisms is proposed by mEducator Best Practice Network (BPN), which brings together the learning material of individual academic institutions with various available standards [21]. The aim of mEducator BPN is to implement and critically evaluate existing standards and reference models in the field of e-learning in order to enable specialized state-of-the-art medical educational content to be discovered, retrieved, shared and re-used across European higher academic institutions [21].

The scientific background of (physiological) educational simulators is ensured by mathematical models which represent a formalised description of (physiological) reality. To enable sharing of models, the description of a mathematical model in “some” standardised form is essential. One possibility is to share models in the form of source code for special modelling tools such as, for example, Matlab/Simulink, which represents the standard for private industry. The disadvantage of this kind of sharing is the fact that the model can be shared only between the owners of the appropriate commercial modelling tool. For this reason, standardised markup languages have been created for the models’ description. These descriptions allow the sharing of models in extensive databases. For models of metabolism, cell signalling, and other processes, Systems Biology Markup Language (SBML) was created, a computer-readable format for representing models of biological processes. On the basis of SBML, an extensive repository of models and software tools were created allowing implementation and testing of behaviour of these models through the use of computers [22]. Another standard based on the eXtensible Markup Language (XML) for sharing physiological models is CellML [23]. The extensive model repository and software tools are created in CellML allowing the scientists to share models even when they use different modelling tools. It also enables scientist to reuse components from one model into another one, and thus accelerates model development. In the model repository of the international project Physiome, models are archived in Jsim-MML (Mathematical Modelling Language) standard [24]. JSim is an open source Java-based simulation system for building quantitative numeric models and analyzing them with respect to experimental reference data. JSim’s primary focus is in physiology and biomedicine; however its computational engine is quite general and applicable to a wide range of scientific domains. JSim also imports the SBML and CellML model archival formats. The modern standard for model description (not only in physiology) is the language Modelica. It is a non-proprietary object-oriented, equation based language that can conveniently model large-scale complex hierarchical systems. In contrast to the clearly text-oriented modelling languages, Modelica also works with a graphic simulation environment and thus is available for creation of hierarchical large-scale models of integrative physiology [25].

3.3. Technology choices

Three technology chains (and pertaining work-flows) have been used during the development of the Atlas. They are illustrated in Table 1, while Table 2 summarizes the advantages and disadvantages of each option.

Simulators based on a relatively simple mathematical model were implemented using the Flash Player platform, which allows them to be run directly in the browser window. The same platform was used for the explicatory lectures of the Atlas. The simulation kernel of the simulator was created in the Action Script language of the development tools Adobe Flash and Adobe Flex, while the same tools were used for the design of the interactive animations.

More complex simulators were implemented using the platform .NET. We developed a software tool that allows for
The ClickOnce technology provides a one-click solution for deploying and running applications on the client's computer with one click using the technological simulator kernel. The created simulators can be installed in various languages, such as C#, and thus creating automatic conversion of the mathematic model implemented in Matlab Simulink into language C#, and thus creating the simulator kernel. The created simulators can be installed on the client's computer with one click using the technological solution ClickOnce [26]. The ClickOnce technology provides a way of running an application by clicking on an Internet link. After confirming the security level, the application is automatically downloaded, transparently installed and run in a Sandbox.

The simulators implemented in the platform Silverlight are the result of our most recent work-flow and technological chain. These simulators can be computationally demanding and yet can be run in the browser window under various operating systems (OS); the prerequisite is the Silverlight plugin in the Internet browser. The underlying mathematical model is first implemented in the modelling tool based on the language Modelica. Typically, Modelica based tools can generate simulator kernels in the C++ language. However, simulators that contain parts of the code in C++ are not allowed to run in the browser window due to security requirements. As active participants in the Open Source Modelica Consortium [27], we have designed and implemented a code generator templating language that enables multi-targeting of the compiler output [28], and we have developed templates for C# code generation from Modelica models. This solution allows for automatic conversion of a Modelica formulated acausal model into a C# formulated simulator kernel, and we can produce pure .NET code able to run even under strict security requirements. Besides these improvements, the technology allows for simulators of a more compact structure. The animations are created by graphic designers in Microsoft Expression Blend – a tool that communicates well with the rest of the platform. The previously mentioned tool of Animittester provides considerable support for cooperation of graphics and programmers. Its interface separates (and connects) the graphic design and simulator programming (see Fig. 2). The artist can create complex animations comfortably and the animations can be controlled easily. The programmer specifies the animation control by connecting it to relevant simulator modules.

The Atlas is composed of explicatory chapters and web-based simulators. The explicatory chapters of the Atlas are designed as audio lectures accompanied by interactive multimedia images (see Fig. 3a). Every animation is synchronized accurately with the explanatory text. Some simulators combine the model with the explicatory part, for instance the simulator of mechanical properties of muscles (see Fig. 3b).

Other simulators can be run separately, and the scenarios used in their control are planned as part of relevant explanatory chapters. The complex model of gas transport by blood is an example; this model can be used as an instructional aid in explaining the physiology and pathophysiology of oxygen and carbon dioxide transport (e.g. to explain the consequences of ventilation-perfusion mismatch). This simulator can be downloaded from our Atlas using the following link: http://physiome.cz/atlas/sim/BloodyMary/.

### Table 2 – Atlas of Physiology and Pathophysiology: comparison of various technologies used in the simulator development.

<table>
<thead>
<tr>
<th>Final product</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A simulator based on Adobe Flash platform</td>
<td>1. No need for simulator installation, the simulator runs under various OS in the browser window (Flash Player plug-in needed)</td>
<td>1. Manual conversion from Simulink to Action Script</td>
</tr>
<tr>
<td></td>
<td>2. Rich support for the visual aspects of the user interface</td>
<td>2. The simulation kernel is relatively slow</td>
</tr>
<tr>
<td>2. A simulator based on .NET platform</td>
<td>1. Automatic generation of the simulation kernel from Simulink</td>
<td>1. Simulator runs under MS Windows OS only</td>
</tr>
<tr>
<td></td>
<td>2. Fast simulation kernel enables creation of computationally demanding simulators</td>
<td>2. Need for simulator installation on the client computer</td>
</tr>
<tr>
<td>3. A simulator based on Silverlight platform</td>
<td>1. No need for simulator installation, the simulator runs under various OS in the browser window (Silverlight plug-in needed)</td>
<td>1. Silverlight plug-in is less widespread than Flash Player</td>
</tr>
<tr>
<td></td>
<td>2. Automatic generation of the simulation kernel from a Modelica based environment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Fast simulation kernel enables creation of computationally demanding simulators</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Declarative model description (using model equations) in Modelica based environments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Common platform of the user interface animations and the simulation kernel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6. Animittester tool provides division between programming and the graphic design of the interactive animations</td>
<td></td>
</tr>
</tbody>
</table>

4. **Samples of typical interactive simulations**

As mentioned in Section 1, the interactive simulations in the Atlas are not always based on highly complex models with a large number of variables. As an example of explaining the physiology and pathophysiology of circulation with a simple aggregated model, the simplest model of circulation with disconnection of regulatory links can be demonstrated here. It has a quite intuitive control, and helps to clarify relationships among individual variables of the circulatory system (i.e. pressures and flows in the pulmonary and systemic circulation) and the essential variables affecting these pressures.
Fig. 2 – Animation of a beating heart. Outputs from the model affect the phases of the heart pulse, opening and closing of cardiac valves, etc. Auxiliary Animtester control elements are installed above the animation and enable the graphic designer to set and tweak each sub-animation. The graphic designer is thus completely shielded from interfering in the programming process. In the final simulator, the “control ropes and levers” are pulled by the simulation model, programmed in the background.
and flows which are themselves regulated by neurohumoral means (see Fig. 4). These variables including: (a) system and pulmonary peripheral resistances; (b) the pumping function of the right and left ventricles implemented simply as the slope of the Starling curve, expressing dependence of the cardiac output on filling pressures of the ventricles; (c) the elasticity of arteries and veins expressing dependence of pressure on the volume of vascular filling; and (d) the total volume of circulating blood. The organism regulates these variables as mentioned above: resistance is controlled by means of both nervous and humoral regulation; myocardial frequency and inotropy modifies the shape of the Starling curve; venous tone (again regulated by neurohumoral means) of large veins changes their elasticity; and the circulating blood volume is

Fig. 3 – (a) Audiovisual interactive lecture in the explanatory part of the Atlas of Physiology and Pathophysiology. Every audio-explanation is accompanied by synchronized animated images. An explanation can be stopped at any moment in order to take a more detailed look at the accompanying animation. It can also be moved backward using the slide at the bottom of the player and (b) mechanical properties of skeletal muscle. An explicatory chapter including various simulation games, it is a Flash application (accessible via http://www.physiome.cz/atlas/sval/svalEN/SvalEN.html)
affected especially by renal activity, the rennin-angiotensin aldosterone system, etc. However, these variables represent input (i.e. non-regulated) quantities in an aggregated model; the aim of the interactive simulation is to obtain a clear notion of the influence of these quantities on blood pressure, flow and distribution of the blood volume among individual parts of the bloodstream. Thus, interactive simulation with this model helps to explain the regulation of essential quantities in the circulatory system in the pathogenesis of various circulatory system disorders. Fig. 4b shows acute right-side circulatory failure by decreasing the slope of the right heart Starling curve, which models the reduction of contractility. The minute heart volume can drop to values such as 3.29 L/min and the mean system arterial pressure to 59.68 mmHg. To preserve perfusion of coronary blood vessels, the sympathetic nervous system responds to the blood pressure decrease by vasoconstriction, especially in the splanchnic region. Therefore, the next step lies in increasing peripheral system resistance by moving the slide to the right (see Fig. 4c). The mean arterial pressure can increase to a value approaching 89.21 mmHg; however, the minute heart volume drops further from 3.29 L/min to 3.07 L/min by the very same mechanism. The sympathetic nervous system does not cause only the vasoconstriction of arterioles and the subsequent increase of peripheral resistance, but also increases the tone of large veins, increasing the pressure in them with the same blood filling. The increase of the venous tone can be modelled by reduced elasticity of systemic veins (see Fig. 4d). Reduced elasticity increases not only the pressure in large system veins but also the filling pressure in the right atrium, leading to increased cardiac output. At the same time, the increase of venous pressure leads to higher filtration in capillaries and oedemas. The mean arterial pressure increases to its normal value and it is not necessary to maintain the resistance value in the system bloodstream at a high level in order to preserve the pressure value. Therefore, the resistance can be decreased to previous levels using the slide (e.g. from 28.37 to 19.24 Torr/L/min). This principal example illustrates how interactive simulations can contribute to a better understanding of the role of a particular regulation circuit in the pathogenesis of various medical conditions, as well as to a better understanding of subsequent therapeutic interventions.

5. Hardware and software specifications

The Atlas is currently designed as a web-based application. Interactive simulation models and tutorials implemented as Flash or Silverlight applications do not need to be installed separately and can be run in an Internet browser on different devices.
Fig. 5 – (a) Interactive educational model of the buffering plasma system. Fluid level values represent concentrations. Initial condition; (b) dilution can be invoked using the control slide; levels of all substances, including the CO2 concentration and hydrogen ion concentration, are reduced. (c) Chemical equilibrium in the buffering system is reached by pressing the button “Buffering Equilibration.” In our example, plasma pH returns to 7.4; (d) respiration brings the CO2 concentration back to the original level, 1.2 mmol/L, after it was reduced by the dilution. When a new chemical equilibrium is established, the hydrogen ion concentration increases and the plasma pH value decreases to acidic values typical of dilutional acidosis.

operation systems. Adobe Flash Player plugin and (for current application) Silverlight plugin installed in the browser are the only real prerequisites.

Part of the simulation models are based on Microsoft .NET framework and can be run only in MS Windows. Installation of these models is offered directly from the Internet browser (if .NET framework is not installed, its installation is offered before installing the first simulator, which requires .NET). In the near future, we will redesign these models into the Silverlight application, which can be run directly in the Internet browser.

6. Mode of availability of the system

The Atlas is a web-based application freely available from the project homepage on http://physiome.cz/atlas. It is created in Czech and English versions. All educational text, interactive animations, and simulation models, including their source codes, are available on demand to all who are interested.

7. Conclusions and future plans

This paper has presented the Atlas, a new interactive multimedia application primarily designed for biomedical education, but also broadly usable for general reflection on physiological regulations and processes. Through development of the Atlas, new technologies have been created for interconnection between simulations models and interactive multimedia. Our latest technology allows for the creation of relatively large-scale models of physiological systems in the language Modelica; from these models we subsequently generate simulation cores of educational interactive multimedia simulators that are created as Silverlight applications, which can be run in the Internet browser. In these applications, it is possible more easily to combine interactive simulations with explicatory instructions.

In the future research, the Atlas will be expanded to include additional topics such as an explicatory lecture including interactive simulations of pulmonary ventilation, circulation, kidney functioning, etc. We furthermore plan to translate the entire Atlas gradually into the English language. Also, we plan
to accompany each educational topic with a detailed description in the form of metadata to allow for its easy connections to international networks for sharing of medical educational content. The interactive simulations available through the Atlas have been already put into use during pathophysiology seminars; so far, the teachers’ experience is positive and the interactive simulations have met with visibly high student approval. Therefore, we believe that the future research will show the impact of the Atlas interactive simulations on effectiveness of medical education. Finally, our future plan is to create a web-based simulator of virtual patient on the basis of a large-scale model for education of emergency medicine.

During the course of further development, we welcome cooperation with anyone who would like to take part in its gradual building process.

Acknowledgements

Development of the Atlas of Physiology and Pathophysiology and educational simulation models is supported by the grants MSM 2006031, MSM 0021620806, MSM 684070012, GACR 102/08/H008, SGS10/180/OKH32/13, SVV-2010-254260507, and by the company Creative Connections s.r.o.

Appendix A. Appendix

Simulators of the Atlas can at first sight resemble simple interactive animations, where an uninformed user might not be aware of the model layer in the background. Perhaps this impression is given by the model of acid–base equilibrium in plasma (see Fig. 5). However, using interactive simulations with this model, it is possible to explain visually the development of various acid–base equilibrium disorders. Buffering systems are displayed as interconnected containers of individual substances; the level of liquid in these containers represents their concentration. Chemical reactions are shown as “flows of liquid” between the containers – i.e. individual buffering system components. The metabolism, the respiratory system or the kidneys can all provide substances that are inside or outside the containers. Dilution of individual buffer components is shown as an expansion of the containers. Since the amount of substance in the containers remains the same during dilution, the level of liquid (representing concentration) drops. At the start, as well as during dilution, the level of hydrogen ions drops (see Fig. 5b). By pressing the button “buffering equilibration,” chemical reactions are started in the buffering systems, visualized as flowing in and out of individual substances. Upon disassociation of carbonic acid and weak buffering acids (denoted as HBuf in the model – represented especially by albumin and phosphates in reality) according to buffering equilibriums, the hydrogen ion level settles on the original value again (see Fig. 5c). The value of carbonic acid, just like the value of carbon dioxide (CO₂), remains reduced due to dilution. However, the CO₂ level is a regulated quantity and respiration soon brings it back to the original value. By pressing the button “respiratory regulation,” the CO₂ level increases back to its value before dilution. By pressing the button “buffering equilibration,” a chemical reaction takes place again, establishing a new chemical equilibrium with increased concentration of hydrogen ions (see Fig. 5d). The simulator thus shows the same underlying principles of this acid–base disturbance as those described in [29].

REFERENCES


Large Scale Physiological Models in Modelica.

Kofránek, Jiří; Mateják, Marek; Privitzer, Pavol

12 str., v tisku
Hummod - Large Scale Physiological Models in Modelica

Jiří Kofránek  Marek Mateják  Pavol Privitzer
Institute of Pathophysiology, First Faculty of Medicine, Charles University
U nemocnice 5, 128 53 Praha 2, Czech Republic
kofranek@gmail.com matejak.marek@gmail.com pavol.privitzer@lf1.cuni.cz

Abstract

Modelica is being used more and more in industrial applications, but Modelica is still not used as much in biomedical applications. For a long time we have mostly been using Matlab/Simulink models, made by Mathworks, for the development of models of physiological systems. Recently, we have been using a simulation environment based on the Modelica language. In this language, we implemented a large scale model of interconnected physiological subsystems containing thousands of variables. Model is a richly hierarchically structured, easily modifiable, and “self-documenting”. Modelica allows a much clearer than other simulation environments, to express the physiological nature of the modeled reality.

Keywords: simulation; physiology; large-scale model

1 Introduction

It is simply amazing how fast the new Modelica simulation language adopted various commercial development environments. Modelica is being used more and more in industrial applications. However, Modelica is still not used as much in biomedical applications.

The vast majority of biomedical simulation applications are still done in casual, block-oriented environments. These include referencing database development environments for biomedical models (such as the JSIM language - http://physiome.org/model/doku.php or CELML language - http://www.cellml.org/).

A frequently used environment in biology and medicine is Matlab/Simulink – monographs dedicated to biomedicine models are usually equipped with additional software used in this environment, but so far without the use of new acasual or non-casual Simulink libraries, such as [23, 27, 31].

However, already in 2006, Cellier and Nebot [5] pointed out the benefits of Modelica, when used for clear implementation of physiological systems descriptions and interpretations. The classic McLeod’s circulation system model was implemented by PHYSBE (PHYS-

2 Web of physiological regulations

Thirty-nine years ago, in 1972 Guyton, Coleman and Granger published an article in the Annual Review of Physiology [9] which at a glance was entirely different from the usual physiological articles of that time. It was introduced by a large diagram on an insertion. Full of lines and interconnected elements, the drawing vaguely resembled an electrical wiring diagram at first sight (Fig. 1). However, instead of vacuum tubes or other electrical components, it showed interconnected computation blocks (multipliers, dividers, adders, integrators and functional blocks) that symbolized mathematical operations performed on physiological variables. In this entirely new manner, using graphically represented mathematical symbols; the authors described the physiological regulations of the circulatory system and its broader physiological relations and links with the other subsystems in the body – the kidneys, volumetric and electrolyte balance control, etc. Instead of an extensive set of mathematical equations, the article used a graphical representation of mathematical relations. This syntax allowed depicting relations between individual physiological variables graphically in the form of interconnected blocks representing mathematical operations. The whole dia-
gram thus featured a formalized description of physiological relations in the circulatory system using a graphically represented mathematical model.

Guyton's model was the first extensive mathematical description of the physiological functions of interconnected body subsystems and launched the field of physiological research that is sometimes described as "integrative physiology" today. Just as theoretical physics tries to describe physical reality and explain the results of experimental research using formal means, "integrative physiology" strives to create a formalized description of the interconnection of physiological controls based on experimental results and explain their function in the development of various diseases.

From this point of view, Guyton's model was a milestone, trying to adopt a systematic view of physiological controls to capture the dynamics of relations between the regulation of the circulation, kidneys, the respiration and the volume and ionic composition of body fluids by means of a graphically represented network.

Guyton's graphical notation was soon adopted by other authors – such as Ikeda et al. (1979) in Japan [13] and Amosov et al. (1977) in the former USSR [2]. However, the graphical notation of the mathematical model using a network of interconnected blocks was only a graphical representation – Guyton's model and later modifications (as well as the models of other authors that adopted Guyton's representative notation) were originally implemented in Fortran and later in C++.

Today the situation is different. Now, there are specialized software simulation environments available for the development, debugging and verification of simulation models, which allow creating a model in graphical form and then testing its behavior. One of these is the Matlab/Simulink development environment by Mathworks, which allows building a simulation model gradually from individual components – types of software simulation elements that are interconnected using a computer mouse to form simulation networks.

Simulink blocks are very similar to the elements used by Guyton for the formalized representation of
physiological relations. The only difference is in their graphical form. This similarity inspired us to use Simulink to revive Guyton’s good, classic diagram and transform it into a working simulation model. When implementing the model in Simulink, we used switches that allow us to connect and disconnect individual subsystems and control loops while the model is running. We strove to keep the appearance of the Simulink model identical to the original graphic diagram – the arrangement, wire location, variable names and block numbers are the same.

The simulation visualization of the old diagram was not without difficulties – there are errors in the original graphic diagram of the model! It does not matter in the hand-drawn illustration but if we try to bring it to life in Simulink, the model as a whole collapses immediately. A detailed description of the errors and their corrections is in [17].

Our Simulink implementation of Guyton’s (corrected) model (Figs. 2 and 3) is available for download at www.physiome.cz/guyton. Also available at that address is our Simulink implementation of a much more complex, later model from Guyton et al. There is also a very detailed description of all applied mathematical relations with an explanation.

3 Block-oriented simulation networks for physiology

Block-oriented simulation languages, of which Simulink is a typical example, allow assembling computer models from individual blocks with defined inputs and outputs. The blocks are grouped in libraries; when building a model, a computer mouse is used to create individual block instances, with inputs and outputs connected through wires that “conduct” information. A Simulink network can be arranged hierarchically. Blocks can be grouped into subsystems that communicate with their ambient environment through defined input and output “pins”, making “simulation chips” of a sort. A simulation chip hides the simulation network structure from the user, much like an electronic chip hiding the interconnection of transistors and other electronic elements. Then the user can be concerned
just with the behavior of the chip and does not have to bother about the internal structure and calculation algorithm.

The behavior of a simulation chip can be tested by monitoring its outputs using virtual displays or virtual oscilloscopes connected to it. This is very useful especially for testing the behaviour of a model and expressing the mutual relations of variables.

Simulation chips can be stored in libraries and users can create their instances for use in their models. For example, we created a Physiolibrary for modelling physiological regulations.

Hierarchical, block-oriented simulation tools are thus used advantageously in the description of the complex regulation systems that we have in physiology.

A formalized description of physiological systems is the subject matter of PHYSIOME, an international project that is a successor to the GENOME project. The output of the GENOME project was a detailed description of the human genome; the goal of the PHYSIOME project is a formalized description of physiological functions. It uses computer models as its methodological tool [3, 12].

Several block-oriented simulation tools developed under the PHYSIOME project have been used as a reference database for a formalized description of the structure of complex physiological models. These include JSIM (http://www.physiome.org/model/doku.php) and CEIIML (http://www.cellml.org).

4 From Simulink to Modelica in modeling of large-scale physiological systems

We have been using Matlab and Simulink for years to create and develop models of physiological systems [15,16,17] and have also been developing the relevant application Simulink library – the Physioli-Web and Microsoft .NET), where we create our own tutorial and education simulators [19].

We have also developed the relevant software tools that simplify the transfer of models implemented in Simulink over to development environments (ControlWeb and Microsoft .NET), where we create our own tutorial and education simulators [19].

Our development team gained priceless experience in previous years working with the Matlab/Simulink development environment made by the renown company MathWorks. On the other hand, we were also attracted by the acausal development environments using the Modelica language.

In the Modelica language environment the essence of physiological regulation is much clearer than in Simulink causal network (see Figures 3 and 4). We were facing a decision whether to continue with the development process of physiological system models in Simulink (using new acausal libraries), or to make a radical decision and switch to the new Modelica language platform.

Our decision was affected by our efforts to imple-
Figure 5: All necessary files of the Quantitative Human Physiology tutorial simulator (called the Hummod by the authors in the last version). This simulator has been designed for the Windows operating system and does not require special installation. Only zip files must be unzipped into a selected folder. After you click the DigitalHuman.exe icon, the translator translates the source text embedded within thousands of directories and more than two thousand files and initiate its own simulator. Even though the source text of the simulator and the entire mathematical model on the background is offered as an open source (and in theory, the user may modify the model), the navigation through thousands of mathematical relations and viewing thousands of XML and interconnected files is rather difficult.

A large model made by Guyton’s disciples and followers. Their Quantitative Human Physiology model is an extension of a tutorial simulator called the Quantitative Circulatory Physiology (QCP) [1]. Quantitative Human Physiology (QHP) simulator [11], which is now distributed as “Hummod” [7], represents today’s most comprehensive and largest model of physiological functions.

The Hummod model contains more than 4000 variables and at the present time, it probably represents the largest and most extensive model of physiological regulations. It enables the user to simulate a wide range of pathological stages and statuses, including the effects of the relevant applied therapy. The authors
developed a special block-oriented simulation system to represent the complex model structure. Compared with the previous QCP simulator, whose mathematical background is hidden from the user in its source code written in C++, the Hummod simulator uses a different approach. The Hummod authors decided to separate the simulator implementation and description of the model quotations, in order to make the structure of the model more clear and apparent for the larger scientific community.

In 1985 the architect of this model, Thomas Coleman, had already created a special language used to write the model structure, as well as the element definitions into the simulator user interface. The language is based on modified XML notation. The model is then written by using XML files. A special converter/decoder (DESolver) converts XML files into executable simulator code.

A detailed description of this language and DESolver converter, as well as the relevant educational tutorial, is freely accessible on the web page of the University of Mississippi (http://physiology.unc.edu/themodelingworkshop). The new Hummod model is written in the XML language as well. Its structure with all details may be found at (http://hummod.org), published as an open source.

Therefore, the user can modify this model as he wishes. However, the model description has been divided into more than three thousand XML files in more than thousand directories, from which the special solver creates and executes the simulator (Figure 5).

The entire structure of the model and following links and references are not easily identifiable. That is why the international research and development team in its SAPHIR project (System Approach for Physiological Integration of Renal, cardiac and respiratory control) decided to use the old Guyton models from 1972 [9] and the Ikeda model from 1979 [13] for the creation of its new and extensive model of physiological functions instead of the freely available QHP model. The source codes of the QHP model appeared unclear or
hard-to-understand to those involved in this project [30].

We have been able to create a special software tool called QHPView (Figure 6), which is able to create a clear and legible overview of mathematical relations and connections from thousands of source codes. We are offering this tool as an open source on the web page at (http://physiome.cz/hummod). First, we tried to implement the QHP/Hummod model in the Simulink environment.

The model contains a wide range of relations that offer solutions for implicit equations. That is why the implementation of this block-oriented model (outputs from one block are used as inputs for the next blocks) is very difficult and as the implementation got more and more complex, the transparency of this model went down quickly. The use of new acasual Simulink libraries in this complex model proved to be problematic and the transparency of the model improved only a little bit.

Therefore, we decided to stop using the Simulink implementation and began to implement the Modelica language (using the Dymola environment). Very quickly we discovered that the implementation of a large and extensive model in Modelica is much more effective than using acausal libraries in Simulink. When we compared the Simulink and Modelica implementations we also discovered a significant difference. Mainly due to the fact that the new acausal libraries are only acausal superstructure of Simulink and not an objectively oriented modeling language based on equations, as the Modelica language is.

Therefore, if we compare the development environments based on the simulation language Modelica with the Matlab/Simulink development environments made by Mathworks, we may say the following:

- contrary to Simulink, the model implemented in Modelica much better reflects the essentials and base of the modeled reality and the simulation modes are more clear, readable and less prone to errors;
- the object architecture in Modelica enables the user to build and tweak models with an hierarchical structure gradually, while using reusable element libraries;
- contrary to Simulink (which is the industrial standard for Mathworks), Modelica is a normalized programming language and therefore, it may contain various commercial and non-commercial developing environments competing between each other. This language is used for specific problem solutions originating in various application fields (for commercial and non-commercial specialized libraries);
- in Modelica it is possible to combine casual (mostly signals) and acausal links non-invasively; and unlike in Simulink, it is also possible, (within interconnected blocks) to create algebraic loops fairly easily – the assembler in Modelica contains symbolic manipulations on the background and therefore the disconnection of algebraic loops is the task for the development environment and not for the programmer.

The above specified reasons led us to use, as the main implementation tool for the model creation, the Modelica language and we also gradually stopped using the Matlab/Simulink environment [20].

5 Hummod in Modelica

The implementation of the Hummod model clearly shows the benefits of the model creation process when done in the Modelica language. If we compare the complex structure of the Hummod model by using the visualization option in QHPView (Figure 5) with examples of implementations done in the simulation language Modelica, shown in Figures 7–13, we can

Figure 7: Structure of Hummod model. Model consist of cardiovascular component (CVS), nutritient and metabolism component, water and osmolarity component, proteins component, O₂, CO₂ and acid-base regulation component, electrolyte component, nervous regulation component, hormone regulation component, status of virtual patient component and setup component. All components are connected with bus connectors.
Figure 8: Structure of cardiovascular component (CVS class).

Figure 9: Structure of systemic circulation component (SystemicCirculation class).

Figure 10: Structure of systemic peripheral circulation component (Peripheral class).

Figure 11: Structure of splanchnic circulation component (SplanchnicCirculation class).

see that the acasual implementation done in Modelica creates a transparent and legible model structure and therefore offers easier model modifications.

The Hummod model implemented in Modelica is being currently modified and extended.

Modifications and extensions of Hummod were partially taken from our original model Golem [15, 16] and further modified according to newest findings and experiences.

Our modifications are mainly extensions, which improve the usability of the model during the modeling of difficult breakdowns in acid-base, ionic, volume and osmotic homeostasis of inner environments, which is very important for urgent medicinal statuses. Our modification of the Hummod model is based mainly on the process of **re-programming the subsystem of acid-base balance**, which is based in the original QHP on the so-called Stewart acid-base balance theory. Simply put, the so-called “modern approach” of Stewart [29] and his followers (e.g. Fencel et al. [8], Sirker et al. [28]) explaining breakdowns in the acid-base balance, uses mathematical relations calculating the concentration of hydrogen ions \([H^+]\) from

1. partial pressure CO2 in plasma \((pCO_2)\),
2. total concentration \([\text{Buf}_w]\), weak (partially dissociated) acids \([\text{HBuf}]\) and their base \([\text{Buf}]\), where: \([\text{Buf}_w\text{Buf}]=\text{Buf}+[\text{HBuf}]\)
3. and from the difference between the concentration of fully dissociated cations and fully dissociated anions in \(SID\) (strong ion difference):
The problem of this approach is that the precision of acid-base calculations in the model depends on the precision of the SID calculation, that is the difference between the concentration of fully dissociated cations (that is mainly sodium and potassium) and fully dissociated anions (mostly chlorides). Imprecision that is created during the modeling of sodium, potassium and chlorides intake and excretion are transferred and reflected by the imprecision in the modeling process of the acid-base status.

Even though Coleman et al. [7], significantly improved the modeling of reception and excretion of sodium, potassium and chlorides in kidneys in his Hummod model, if we model a long-term status (when nothing is happening with the virtual patient), the virtual patient (in the current model version) has a tendency to fall into slight and steady metabolic acidosis after one month of the simulated time.

Our evaluative approach towards the modeling and evaluation of breakdowns in acid-basic balance [14, 18, 21] is based on the modeling and evaluation of two flows—the creation and excretion of CO₂ and the creation and excretion of strong acids, connected through the purification systems of each part of the bodily fluids. This approach, according to our opinion, better explains the physiological causality of acid-base regulations, rather than direct modeling of acid-base breakdowns through the balancing of accompanying electrolytes. Besides, the fidelity and truthfulness of the modeling process is getting better; mainly in mixed (acid-base and electrolyte) breakdowns in inner environments.

Another important modification of the Hummod, is the fact that the model was extended by adding the dependency of the potassium flow on the intake of glucose as a result of insulin, which enables us to model (besides other things), the influence of potassium solution infusions with insulin and gluoses, which are distributed in acute medicine for treating potassium depletions.

We have been using this “balancing and evaluation” approach [18] towards the modeling of acid-base balance in our old “Golem” simulator [15]. The extended Hummod model serves as the base for the educational simulator „eGolem“, used in medical tutoring in clinical physiology of urgent statuses which is being currently developed.

On the webpage [http://physiome.cz/hummod] you may find the updated and current structure of our implementation of the Hummod model („Hummod-Golem edition“). In collaboration with M. Tiller we are preparing a detailed description of this model with extensive descriptions of the various physiological regulatory circuits.

6 From a model to the simulator

A simulation model, implemented in the most sophisticated development environment, cannot be used as an education aid alone. It is the implementation of the formalized description of the modeled reality that enables testing of the behavior of the mathematic model during various input values and the search for model quotations and parameters, which within the
Figure 14: The original solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is a high-quality script, created by an experienced pedagogue. The creation of animated pictures is done by artists who create interactive animations in Adobe Flash. The core of simulators is the simulation model, created with special development tools, designed for the creation of simulation models. For a long time, we have been using Matlab/Simulink made by Mathworks for the development of models. The simulator development process is a demanding programming work. To make this task easier, we have developed special programmes that simplify the automatic transfer process of simulation models created in Matlab/Simulink over to ControlWeb or over to the Microsoft .NET environment.

established precision range, can ensure the sufficient compatibility of the behavior of the model with the modeled system (model identification).

Even after this goal is reached, there is still a long road ahead from the identified model to the educational or tutorial simulator. It is a very demanding development work, which requires the combination of ideas and experiences of teachers who create the script of the tutorial application, the creativity of art designers who create the multimedia components interconnected with the simulation model in the background, as well as the efforts of programmers who finally “sew up” the final masterpiece into its final shape. We have used a special web simulator creation technology for

Figure 15: Our new solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is still a high-quality script, created by an experienced pedagogue. The creation of animated pictures is done by artists who create interactive animations in Expression Blend. To create and test animations that will be controlled by the simulation model, art designers use the AnimoTester software tool, developed by us. The core of simulators is the simulation model, created in the Modelica simulation language environment. Within the project Open Modelica Source Consortium, we are in the process of creating a tool which will be able to generate the source text from Modelica in the C# language. This will enable us to generate a component from .NET used in the final application on the Silverlight platform, which will enable us to distribute the simulator as a web application, running in the internet browser (even on computers with various operating systems).
creation of educational simulators [22].
To automate the model debugging transfer from the simulation development environment (previously using Simulink and nowadays using Modelica) into the development environment where the development application is programmed, specialized software tools (developed by us) are used. We have been creating tutorial simulators in Microsoft .NET and Adobe Flash environments (Figure 14). Recently, we began using the Microsoft Silverlight platform (Figure 15), which enables distribution of simulators over the internet and may be executed directly into the internet browser environment (even on computers running various operating systems).

7 Conclusions
Nowadays, the old Comenius motto – “schola ludus,” or “playful school” [6], has found a modern use in interactive educational programs that use simulation games. Connection of the multimedia environment, serving as an audio-visual user interface, with simulation models, gives the studied problem a much more tangible feeling. A simulation game offers the possibility to test, without any risk, the simulated object’s behavior. The behavior of individual physiological subsystems can be appreciated in a simulation game, both under normal conditions and in the presence of a disorder.

Complex integrative simulators of human physiology can be of large importance when teaching pathophysiology or studying pathogenesis of varied medical conditions and syndromes using virtual patients. Such simulators include models of not only individual physiological subsystems but also of their mutual connection into more complex units. Modelica is a very convenient developing tool for design of those complex hierarchical models.

References


Acknowledgement

Work on the development of medical simulators has been supported by the grants MSM 206031 („eGolem”), MSM 0021620806 and by Creative Connections s.r.o.