

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ
TECHNOLÓGIE

VIRTUÁLNE LABORATÓRIUM

DIPLOMOVÁ PRÁCA

FCHPT-5414-28522

Študijný program: Automatizácia a informatizácia v chémii a v potravinárstve

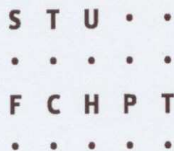
Číslo a názov študijného odboru: 5.2.14 Automatizácia

Školiace pracovisko: Oddelenie informatizácie a riadenie procesov

Vedúci záverečnej práce/školiťel': Ing. Ľuboš Čírka, PhD.

Bratislava 2010

Bc. Lenka Vlková



ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Lenka Vlková**
ID študenta: 28522
Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor: 5.2.14 automatizácia
Vedúci práce: Ing. Ľuboš Čirka, PhD.

Názov práce: **Virtuálne laboratórium**

Špecifikácia zadania:

Cieľom diplomovej práce je vybudovanie virtuálneho laboratória na monitorovanie procesov chemickej, biochemickej a potravinárskej technológie. Toto laboratórium bude slúžiť na výučbu dynamických vlastností procesov a experimentálnu identifikáciu procesov. Vybudovanie tohto virtuálneho laboratória umožní skvalitniť a zatriktívniť výučbu vybraných predmetov špecializácie. Študenti vo virtuálnom laboratóriu získajú potrebné vedomosti a zručnosti, ktoré následne využijú pri práci na reálnych laboratórnych zariadeniach. Výsledkom celého procesu bude jednoduchý prístup k študijným materiálom a laboratórnym experimentom pre pohybovo postihnutých a cestujúcich študentov ako aj širokú verejnosť. Využitím Internetu sa zvýši prístupnosť študijných materiálov a laboratórií až po hodnotu 24/7.

Použité technológie: CMS Moodle, PHP a Flash

Rozsah práce: 50

Riešenie zadania práce od: 15. 02. 2010

Dátum odovzdania práce: 22. 05. 2010



Bc. Lenka Vlková
študentka

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracovala sama. Všetky pramene, zdroje a literatúru, ktorú som použila, uvádzam v zozname použitej literatúry.

V Bratislave 22. 05. 2010

.....

podpis

POĎAKOVANIE

Ďakujem veľmi pekne vedúcemu mojej diplomovej práce Ing. Lubošovi Čirkovi, Phd. za všetky rady, pripomienky a vedenie, ktoré mi poskytol pri tejto práci a všetkým ostatným, ktorí ma podporovali.

SÚHRN

Diplomová práca sa zaoberá tvorbou virtuálneho laboratória na sledovanie modelov procesov chemickej technológie. Toto virtuálne laboratórium bolo vytvorené pomocou programu Macromedia Flash. Laboratórium obsahuje päť modelov. Medzi modely patrí 1 zásobník kvapaliny, 2 zásobníky kvapaliny s interakciou, 3 zásobníky kvapaliny bez interakcie, prietokový ohrievač a miešač s dvomi prúdmi s rôznou teplotou. Tieto animácie, ktoré som vytvorila, zobrazujú priebeh daného procesu. Animácie pomôžu študentom pri študovaní, poznávaní neznámych procesov a pomôžu im veľmi rýchlo pochopiť ako daný proces funguje. Študenti získajú potrebné vedomosti a zručnosti, ktoré neskôr môžu využiť pri riadení reálnych zariadení. Do animácie zadajú niekoľko vstupných hodnôt a za pár sekúnd uvidia ako proces beží, ako sa menia hodnoty výstupných veličín. Veľkou výhodou takýchto animácií je, že študenti nemusia tráviť čas s počítaním hodnôt a simulovaním v rôznych programoch. Takto to majú za niekoľko sekúnd všetko zobrazené. Stačí si zapnúť Internet a aj z pohodlia domova môžu nahliadnuť na spomínané procesy.

Kľúčové slová: virtuálne laboratórium, Macromedia Flash, ActionScript, modely procesov

ABSTRACT

A diploma thesis deals with creation virtual laboratory to tracking process of chemical technology. This virtual laboratory was created with programm Macromedia Flash. Laboratory contains five models. These models belongs one tank of liquid, two tanks of liquid with interaction, three tanks of liquid without the interaction, a flow heater and mixer with two currents with different temperature. These animations which I created, represent running of the process. Animations help to students for studying, recognition unknowns processes and help to understand them very quickly as that process functions. Students get needed knowledge and skills which can later use at control real gears. They enter some inputs values to animation and during a few seconds they see as process is running, as the value of input values change. A big advantage of these animations is that students do not have to spend time with counting of values a with simulation on different programmes. So they have during a few seconds all displays. Enough to turn the Internet on and also from comfort of the home can insight on said processes.

Keywords: virtual laboratory, Macromedia Flash, ActionScript, models of the processes

Obsah

Zoznam obrázkov	8
ÚVOD.....	10
1. Teoretická časť	12
1.1 Matematické modely	12
1.1.1 Zásobník kvapaliny	13
1.1.2 Dva zásobníky kvapaliny s interakciou	14
1.1.3 Tri zásobníky kvapaliny bez interakcie	14
1.1.4 Prietokový ohrievač	15
1.1.5 Miešanie dvoch prúdov s rôznou teplotou.....	16
1.2 Metóda Runge-Kutta	17
1.3 Virtuálne laboratória.....	19
1.4 Prínos použitia on-line výučby	20
1.5 Macromedia Flash	21
1.5.1 História Flash.....	24
1.5.2 ActionScript.....	25
1.5.3 História ActionScriptu	26
2. Praktická časť	28
2.1 Vlastnosti projektov.....	28
2.2 Typy modelov.....	28
2.3 Tvorba animácií.....	29
2.3.1 Grafika animácií	29
2.3.2 Vrstvy animácie	30
Záver	60
Zoznam použitej literatúry.....	61
Prílohy	62

Zoznam obrázkov

Obr. 1: Panel nástrojov	23
Obr. 2: Časová os – Timeline	24
Obr. 3: Vývojové prostredie Flash	25
Obr. 4: Panel ActionScript	26
Obr. 5: Vrstvy a snímky animácií.....	29
Obr. 6: Objekty na scéne animácie.....	31
Obr. 7: Zadávanie konštánt - 1 zásobník kvapaliny	32
Obr. 8: Zadávanie konštánt - 2 zásobníky kvapaliny	32
Obr. 9: Zadávanie konštánt - 3 zásobníky kvapaliny	33
Obr. 10: Zadávanie konštánt - prietokový ohrievač	33
Obr. 11: Zadávanie konštánt - proces miešania 2 prúdov s rôznou teplotou.....	34
Obr. 12: Výpočet rovnovážneho stavu - 1 zásobník kvapaliny	34
Obr. 13: Výpočet rovnovážneho stavu - 2 zásobníky kvapaliny	35
Obr. 14: Výpočet rovnovážneho stavu - 3 zásobníky kvapaliny	35
Obr. 15: Výpočet rovnovážneho stavu - prietokový ohrievač.....	36
Obr. 16: Výpočet rovnovážneho stavu - proces miešania 2 prúdov s rôznou	36
Obr. 17: Výsledná animácia jedného zásobníka kvapaliny	37
Obr. 18: Výsledná animácia dvoch zásobníkov kvapalín s interakciou	37
Obr. 19: Výsledná animácia troch zásobníkov kvapaliny bez interakcie.....	38
Obr. 20: Výsledná animácia prietokového ohrievača.....	38
Obr. 21: Výsledná animácia procesu miešania 2 prúdov s rôznou teplotou.....	39
Obr. 22: Zastavenie časovej osi, nastavenie kvality a rozlíšenia.....	40
Obr. 23: Výpočet rovnovážneho stavu pre 1 zásobník kvapaliny	40
Obr. 24: Výpočet rovnovážnych stavov pre 2 zásobníky kvapaliny	41
Obr. 25: Výpočet rovnovážnych stavov pre 3 zásobníky kvapaliny	41
Obr. 26: Výpočet rovnovážneho stavu pre ohrievač	42
Obr. 27: Výpočet rovnovážneho stavu pre miešač	42
Obr. 28: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 1 zásobník kvapaliny	43
Obr. 29: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 2 zásobníky kvapaliny	43

Obr. 30: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 3 zásobníky kvapaliny	44
Obr. 31: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre ohrievač	44
Obr. 32: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre miešač	45
Obr. 33: Runge – Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie	46
Obr. 34: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie	48
Obr. 35: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie	51
Obr. 36: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie	53
Obr. 37: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie teploty miešača s dvomi rôznymi teplotami	55
Obr. 38: Slučka na snímku 5	56
Obr. 39: Oznam, ktorý sa zobrazí pri nezadaní potrebných hodnôt	56
Obr. 40: Ukážka príkladu „on (release)“ a „gotoAndStop(3)“	58

ÚVOD

Vyspelé krajiny po celom svete sa chcú neustále zdokonaľovať a zabezpečiť si čo najlepšiu budúcnosť. Vedia, že k dosiahnutiu takýchto cieľov, patrí aj vzdelávanie mladých ľudí. Preto sa hľadajú neustále nové spôsoby a možnosti ako zlepšiť a zmodernizovať vzdelávanie na školách a vzdelávacích inštitúciách. Žijeme v dobe inovácií a informačno-komunikačných technológií. Tie zavládli na celom svete a uľahčujú život mnohým ľuďom. Internet je jeden z najznámejších objavov, ktorý otvoril nové cesty k poznávaniu, vzdelávaniu a to nielen na školách, ale aj v rôznych inštitúciách. Foriem vzdelávania je veľa. Jednou z možností ako zabezpečiť on-line vzdelávanie je aj vo forme virtuálnych laboratórií.

Každý jeden z nás sa určite na Internete stretol s množstvom rôznych programov, ktoré ponúkajú veľké možnosti práce s multimédiami. Ved' predsa v dnešnej dobe je to samozrejmosť, že tvorcovia Internetu sa snažia vytvoriť čo najlepšie a najzaujímavejšie prostredie pre užívateľov. Jedným takýmto programom je aj Flash. Stretávame sa s ním v rozličnej forme, či už vo forme animácií, videí, reklamných pútačov, hier alebo aj webových stránok. Foriem je množstvo. Jeden spôsob využitia Macromedia Flash je aj tvorba prezentácií.

Cieľom tejto práce bolo vytvorenie virtuálneho laboratória. Pod pojmom virtuálne laboratórium si môžeme predstaviť poznávanie neznámych procesov, získavanie vedomostí, zručností, ktoré môžu byť neskôr využité v praxi. Virtuálne laboratória slúžia v rôznych oblastiach ako napríklad lekárstvo, chémia, elektrotechnika, strojárstvo, informatika, matematika, psychológia.

Práca pozostáva z dvoch častí, z teoretickej a praktickej časti. Teoretická časť opisuje samotný program Flash a skriptovací jazyk ActionScript, on-line virtuálne laboratória a metódu Runge-Kutta, podľa ktorej sa v animáciách počítali výstupné veličiny. Praktická časť zahŕňala tvorbu samotného virtuálneho laboratória v programe Macromedia Flash, ktoré pozostávalo z 5 animácií modelov rôznych procesov. Medzi modely patrí 1 zásobník kvapaliny, 2 zásobníky kvapaliny

s interakciou, 3 zásobníky kvapaliny bez interakcie, prietokový ohrievač a miešač dvoch prúdov s rôznou teplotou.

Animácie budú študentom pomáhať získať potrebné vedomosti a zručnosti, ktoré budú môcť neskôršie využiť pri reálnych zariadeniach. Otvoria si animáciu, zadajú niekoľko vstupných a začiatočných hodnôt a behom pár sekúnd uvidia, ako proces beží. Stačí si zapnúť internet a aj z pohodlia domova môžu pracovať so spomínanými procesmi.

1. Teoretická časť

1.1 Matematické modely

[1]

„Za účelom kvantitatívneho vyjadrenia vlastností procesov sa využívajú matematické opisy. Tieto opisy sa nazývajú matematické modely. Matematický model procesu je matematickou abstrakciou reálneho procesu a poskytuje možnosť určiť správanie sa procesu, ak sú známe vstupy. Matematické modely procesov môžu byť použité pri riadení spojitých technologických celkov, pri skúmaní dynamických vlastností procesov, pri optimálnom návrhu procesov...„

Ak chceme skúmať ľubovoľný proces, tak musíme vedieť ako zostrojiť matematický model daného procesu. Vytvorený matematický model v sebe zahŕňa základné rovnice, veličiny, opis statiky a dynamiky.

[1]

„Pre riadenie je dôležitý dynamický model procesu. Pri vytváraní matematického modelu procesu sa vychádza z problému skúmania, pričom je dôležité dokonalé pochopenie skúmaných javov. Keď sa zostavuje model procesu za účelom simulácie riadenia procesu, potom sa okrem iného vyžaduje, aby zahŕňal prinajmenšom tú časť procesu, ktorá rozhodujúcim spôsobom ovplyvňuje dynamiku procesu.“

Primárnou úlohou pri zostavovaní modelu procesu je potrebné určiť známe konštanty alebo parametre, ktoré sú počas celého priebehu nemenné. Ich nemennosť môže vyplývať z rozmerov zariadenia, z konštantných fyzikálnych vlastností, atď. Ďalším krokom je nutné určiť veličiny, ktoré sa budú získavať riešením rovníc modelu. A poslednou fázou je treba určiť veličiny, ktorých časový priebeh je daný okolím procesu.

[2]

Matematické modely sa používajú v prírodných vedách, inžinierskych disciplínach, ale aj v sociálnych vedách. Matematické modely môžeme klasifikovať niekoľkými spôsobmi:

- lineárne a nelineárne
- deterministické a stochastické
- statické a dynamické
- so sústredenými a s rozloženými parametrami

1.1.1 Zásobník kvapaliny

Zásobníky kvapaliny patria medzi základné zariadenia v prevádzkach, ktoré slúžia na transport tekutín. Z matematického hľadiska zásobníky predstavujú nelineárny model, ktorý sa dá opísať rovnicou.

Nelineárny dynamický matematický model:

rovnica dynamiky

$$\frac{dh(t)}{dt} = \frac{1}{F} q_0(t) - \frac{k_{11}}{F} \sqrt{h(t)} \quad h(0) = h^s \quad (1)$$

rovnica výstupu

$$y(t) = h(t) \quad (2)$$

Matematický model rovnovážneho stavu:

$$0 = \frac{1}{F} q_0^s - \frac{k_{11}}{F} \sqrt{h^s} \quad (3)$$

1.1.2 Dva zásobníky kvapaliny s interakciou

Nelineárny dynamický matematický model (DMM):

rovnice dynamiky

$$\frac{dh_1(t)}{dt} = \frac{1}{F_1} q_0(t) - \frac{k_{11}}{F_1} \sqrt{h_1(t) - h_2(t)} \quad h_1(0) = h_1^s \quad (4)$$

$$\frac{dh_2(t)}{dt} = \frac{k_{11}}{F_2} \sqrt{h_1(t) - h_2(t)} - \frac{k_{22}}{F_2} \sqrt{h_2(t)} \quad h_2(0) = h_2^s \quad (5)$$

$$y_1(t) = h_1(t) \quad (6)$$

$$y_2(t) = h_2(t) \quad (7)$$

Matematický model rovnovážneho stavu:

$$0 = \frac{1}{F_1} q_0^s - \frac{k_{11}}{F_1} \sqrt{h_1^s - h_2^s} \quad (8)$$

$$0 = \frac{k_{11}}{F_2} \sqrt{h_1^s - h_2^s} - \frac{k_{22}}{F_2} \sqrt{h_2^s} \quad (9)$$

1.1.3 Tri zásobníky kvapaliny bez interakcie

Nelineárny dynamický matematický model (DMM):

rovnice dynamiky

$$\frac{dh_1(t)}{dt} = \frac{1}{F_1} q_0(t) - \frac{k_{11}}{F_1} \sqrt{h_1(t)} \quad h_1(0) = h_1^s \quad (10)$$

$$\frac{dh_2(t)}{dt} = \frac{k_{11}}{F_2} \sqrt{h_1(t)} - \frac{k_{22}}{F_2} \sqrt{h_2(t)} \quad h_2(0) = h_2^s \quad (11)$$

$$\frac{dh_3(t)}{dt} = \frac{k_{22}}{F_3} \sqrt{h_2(t)} - \frac{k_{33}}{F_3} \sqrt{h_3(t)} \quad h_3(0) = h_3^s \quad (12)$$

$$y_1(t) = h_1(t) \quad (13)$$

$$y_2(t) = h_2(t) \quad (14)$$

$$y_3(t) = h_3(t) \quad (15)$$

Matematický model rovnovážneho stavu:

$$0 = \frac{1}{F_1} q_0^s - \frac{k_{11}}{F_1} \sqrt{h_1^s} \quad (16)$$

$$0 = \frac{k_{11}}{F_2} \sqrt{h_1^s} - \frac{k_{22}}{F_2} \sqrt{h_2^s} \quad (17)$$

$$0 = \frac{k_{22}}{F_3} \sqrt{h_2^s} - \frac{k_{33}}{F_3} \sqrt{h_3^s} \quad (18)$$

1.1.4 Prietokový ohrievač

Prietokový ohrievač je zariadenie, využívané na ohrev vody v rôznych oblastiach. V technologických prevádzkach, ale samozrejme aj v domácnostiach. Takisto ako pri zásobníkoch, aj prietokový ohrievač sa dá opísať rovnicou, ktorá má nelineárny tvar.

Nelineárny dynamický matematický model (DMM):

rovnica dynamiky

$$\frac{d\vartheta(t)}{dt} = \frac{q}{V} \vartheta_0(t) + \frac{1}{\rho c_p V} \omega(t) - \frac{q}{V} \vartheta(t) \quad \vartheta(0) = \vartheta^s \quad (19)$$

rovnica výstupu

$$y_1(t) = \vartheta(t) \quad (20)$$

Matematický model rovnovážneho stavu:

$$0 = \frac{q}{V} \vartheta_0^s + \frac{1}{\rho c_p V} \omega^s - \frac{q}{V} \vartheta^s \quad (21)$$

1.1.5 Miešanie dvoch prúdov s rôznou teplotou

Nelineárny dynamický matematický model (DMM):

rovnica dynamiky

$$\frac{d\vartheta_1(t)}{dt} = \frac{1}{V} q_{01}(t) \vartheta_{01}(t) + \frac{1}{V} q_{02}(t) \vartheta_{02}(t) - \frac{1}{V} (q_{01}(t) + q_{02}(t)) \vartheta_1(t) \quad \vartheta_1(0) = \vartheta_1^s \quad (22)$$

rovnica výstupu

$$y(t) = \vartheta_1(t) \quad (23)$$

Matematický model rovnovážneho stavu:

$$0 = \frac{1}{V} q_{01}^s \vartheta_{01}^s + \frac{1}{V} q_{02}^s \vartheta_{02}^s - \frac{1}{V} (q_{01}^s + q_{02}^s) \vartheta_1^s \quad (24)$$

1.2 Metóda Runge-Kutta

[1]

„Metóda Runge-Kutta je založená na rozvoji funkcie do Taylorovho radu. Taylorov rad môže byť použitý na vyjadrenie riešenia diferenciálnej rovnice $x(t + \Delta t)$ v závislosti od $x(t)$ a jej derivácií nasledovne

$$x(t + \Delta t) = x(t) + \Delta t \dot{x}(t) + \frac{1}{2} (\Delta t)^2 \ddot{x}(t) \dots \quad (25)$$

Ak diferenciálna rovnica, ktorú chceme riešiť, má tvar

$$\frac{dx(t)}{dt} = f(t, x(t)) \quad (26)$$

potom

$$\dot{x}(t) = f(t, x(t)) \quad (27)$$

$$\ddot{x}(t) = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \frac{dx}{dt} = \frac{\partial f}{\partial x} f \quad (28)$$

a potom

$$x(t + \Delta t) = x(t) + \Delta t f + \frac{1}{2} (\Delta t)^2 (f_t + f_x f) + \dots \quad (29)$$

pričom

$$f \equiv f(t, x(t)), f_t \equiv \frac{\partial f}{\partial t}, f_x \equiv \frac{\partial f}{\partial x} \quad (30)$$

Riešenie $x(t + \Delta t)$ je závislé na znalosti derivácií funkcie f . Vyššie derivácie funkcie f sa ťažko určujú, a preto pri aplikácii vzťahu (29) sa pri riešení diferenciálnej rovnice uvažuje len niekoľko členov na pravej strane rovnice (29). Jedná sa o

aproximáciu Taylorovho radu, čo je podstatou metódy Runge-Kutta. Počet členov určuje rád metódy Runge-Kutta.

S uvažovaním integračného kroku h bude platiť

$$t_{k+1} = t_k + h \quad (31)$$

Metóda Runge-Kutta druhého rádu je založená na aplikácii diferenčnej rovnice

$$x(t_{k+1}) = x(t_k) + h\dot{x}(t_k) + \frac{1}{2}h^2\ddot{x}(t_k) \quad (32)$$

Rekurzívny vzťah vhodný na riešenie diferenciálnych rovníc vyplývajúci z metódy Runge-Kutta druhého rádu je nasledovný

$$x_{k+1} = x_k + \gamma_1 k_1 + \gamma_2 k_2 \quad (33)$$

γ_1, γ_2 - váhové koeficienty

Najznámejší rekurzívny vzťah vhodný na riešenie diferenciálnych rovníc vyplývajúci z metódy Runge-Kutta štvrtého rádu je

$$x_{k+1} = x_k + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \quad (34)$$

pričom

$$\begin{aligned} k_1 &= hf(t_k, x_k) \\ k_2 &= hf\left(t_k + \frac{1}{2}h, x_k + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(t_k + \frac{1}{2}h, x_k + \frac{1}{2}k_2\right) \\ k_4 &= hf(t_k + h, x_k + k_3) \end{aligned} \quad (35)$$

1.3 Virtuálne laboratória

Po celom svete sa stredné a vysoké školy snažia spestriť výučbu rôznymi spôsobmi. Využívanie informačno-komunikačných technológií sa stáva v poslednom čase veľmi dôležité a nevyhnutné pri modernizácii vyučovacieho procesu. Dištančná forma vzdelávania je teda taká forma, pri ktorej sa študent s vyučujúcim nemusí stretnúť na jednom mieste, ale komunikujú spolu on-line. Takúto formu využíva množstvo vzdelávacích inštitúcií po celom svete – USA, Kanada, Austrália, Nemecko a samozrejme aj Slovensko.

Medzi spestrenie výučby patria aj tzv. virtuálne laboratória, v ktorých si študenti cez Internet z rôznych kútov sveta uľahčujú výučbu. Medzi univerzity, ktoré takéto laboratória majú, patrí napríklad:

- Strojnícka fakulta STU v Bratislave
<http://suhajko.com/laboratory/index.php/lang-sk/virtlab>
- Fakulta Elektrotechniky a informatiky v Bratislave
<http://ural.elf.stuba.sk/vrlab/>
- Fakulta chemickej a potravinárskej technológie
<http://ka.chtf.stuba.sk/lcza/>
- Johns Hopkins University
<http://www.jhu.edu/virtlab/virtlab.html>
- The University of Utah
<http://learn.genetics.utah.edu/content/labs/gel/>

Nielen školy majú svoje virtuálne laboratória, ale takisto existujú stránky s podobným obsahom, ktoré slúžia ako virtuálne laboratória, napr.:

- <http://www.engin.umich.edu/group/ctm/>
- <http://www.softintegration.com/webservices/control/>
- <http://plc.cwru.edu/tutorial/enhanced/lab/lab.htm>
- <http://www.chemcollective.org/vlab/vlab.php>
- http://websites.kahoks.org/Richert_Gary/sciweb/applets.html

Tieto stránky obsahujú virtuálne laboratória, ktoré sa týkajú:

- Riadenia križovatky, dopravníka, modelu robota, návrhu PID pomocou GA, prekladania kontajnerov, matematických operácií polynómov, riešenia systému rovníc, grafov pólov a núl, prechodových a impulzných charakteristík, algebre prenosov, grafickej konvolúcie nepretržitého signálu na obrazovke, Fourierovho radu aproximácií, skúmaniu frekvencie, používaniu Matlabu, vykresľovaniu 2D a 3D grafov a množstvo ďalších návodov.

1.4 Prínos použitia on-line výučby

On-line výučba ponúka široké spektrum výhod oproti klasickému vyučovaniu. Tu je zopár príkladov prínosu on-line výučby na školách a vzdelávacích inštitúciách:

1. mobilita projektora a notebooku, diaľkové ovládanie, čo umožňuje vyučujúcim oslobodiť sa od státi pri projekčnom prístroji,
2. vysoká spoľahlivosť v porovnaní s klasickým projektorom,
3. možnosť pracovať aj on-line na Internete a spúšťať aplikácie, videá, obrázky...,
4. možnosť prenosu úloh, dát, referátov, prednášok pomocou kompaktných diskov, či USB kľúčov,
5. možnosť návratu ku ktorejkoľvek snímke, bez ohľadu na to, v ktorej časti prednášky sa vyučujúci nachádza,
6. široké spektrum rôznych odkazov, webových stránok, dokumentov,
7. možnosť kedykoľvek materiál doplniť alebo zmeniť,
8. rýchlejšia možnosť komunikácie medzi študentmi a vyučujúcimi,
9. lepšia dostupnosť študijných materiálov pre hendikepovaných študentov,
10. štúdium podľa vlastného tempa.

Avšak on-line výučba môže mať aj svoje nevýhody, napr.:

1. príprava multimedialných prezentácií je veľmi náročná – financie, čas, podklady...,
2. nevyspytateľnosť techniky – výpadok elektriny, pokazenie prístroja...,

3. náklady študentov – kúpa počítača a internetu.

Avšak takýchto nevýhod je len veľmi málo a samozrejme výhody prevyšujú tieto malé nedostatky a preto zavádzanie dištančnej formy vzdelávania je veľmi aktuálne.

1.5 Macromedia Flash

[3], [4]

Flash je multimediálny softvér, vytvorený v Amerike firmou Macromedia a v súčasnej dobe je vyvíjaný a šírený pomocou Adobe Systems. Táto firma sa už preslávila produktmi ako sú napríklad Macromedia Director, či Macromedia Dream.

Flash je ale jedným z najrozšírenejších programov na vytváranie atraktívnych internetových stránok, prezentácií, aplikácií, rôznych umeleckých, zábavných, či informatívnych projektov, ktoré spájajú fotografie, text, zvuk, animáciu, či zvuk. Je založený na vektorovej grafike. Vektorová grafika je jednoznačne najmenej náročná na dátový objem informácií. Zjednodušene povedané, je to animácia, čiže skupina obrázkov, ktoré sa medzi sebou vymieňajú. Niečo podobné ako animovaný gif. Avšak zatiaľ čo v gife sa striedajú rastrové obrazy, vo Flash sa jedná o vektorové obrazy. Rastrový obraz je tvorený v sieti bodov - pixelov, kde každý bod má presne určenú polohu a farbu. Rastrovanie vzniká snímaním predlohy, preto sú bitmapy dominantné v oblasti spracovania fotografií. Vektorové obrazy sú tvorené grafikou, ktorá je definovaná pomocou čiar a výplní. Vektorová grafika je menej náročná na dátový objem informácií o výzore obrázku. Krivka, ktorú takto nakreslíme je definovaná počiatočnými bodmi, vektorom, ktorý určuje smer a zakrivenie koncovým bodom. Program zakaždým vykreslí celý obrázok pomocou matematických vzorcov, pri ktorých nie je problém napríklad prepočítanie obrázku na inú veľkosť bez straty kvality. A najväčšie plus možno zbrať pri grafike nakreslenej priamo v programe, ale nerobí mu problém ani import grafických súborov. Aj keď za cenu zbytočnej veľkosti.

Flash pracuje s vrstvami, ktoré sa dajú veľmi ľahko ovládať. Tieto vrstvy sa najviac využívajú pri animáciách, pri ktorých sa využíva veľké množstvo prvkov.

Výhody, ktoré prináša vektorová grafika oproti rastrovej:

- vektorová grafika je dátovo omnoho menšia
- objekty môžu byť zväčšované bez straty kvality
- výplne a čiary môžu byť priesvitné alebo vyfarbené prechodovou výplňou s minimálnym zväčšením súboru

Nevýhody vektorovej grafiky, ktoré umožňujú vkladať do animácií aj rastrové obrázky:

- vektorová grafika sa nehodí na veľmi zložité obrazce (napr. fotky...)

Flash sa stal súčasťou moderného webu. Program bol navrhnutý na propagáciu vektorovej grafiky na Internete. Používa sa na vytvorenie animácií, reklám, videí, webových aplikácií, kreslených snímok, hier... V prostredí Flash je možné vytvárať nielen vektorovú grafiku pre web stránky, ale aj napríklad plnohodnotné formuláre, programy komunikujúce s databázou...[5]

Program obsahuje veľké množstvo nástrojov, potrebných pre tvorbu objektov. Paleta nástrojov ponúka veľké množstvo prvkov pre prácu s farbami, pohybom, veľkosťami, rôznymi efektmi. Program je veľmi prehľadný a ak by nastal nejaký problém, nájdeme v ňom pomocníka (HELP), ktorý je vo forme HTML.

Pracovný súbor, ktorý skompilujeme má koncovku FLA a výsledná animácia môže mať rôzny formát. Najpoužívanejší je súbor s koncovkou SWF, pretože ponúka veľké možnosti užívateľskej interaktivity vďaka jazyku ActionScript.

A čo si vieme predstaviť pod pojmom animácia? Pod animáciou si môžeme predstaviť určitý pohyb. Rovnako za animáciu môžeme považovať zmenu farby, tvaru, či veľkosti nejakého objektu. Animácie sú poskladané z jednotlivých snímok, ktoré sa zobrazujú jedna po druhej a vytvárajú tak dojem plynulého pohybu.

Pri práci vo Flash sa musíme najskôr rozhodnúť, čo chceme vytvárať. Môžeme sa rozhodnúť pre obyčajný Flash dokument alebo napríklad pre Flash prezentáciu. Pri práci s ktoroukoľvek formou Flash potrebujeme určité nástroje, ktoré nám zabezpečujú vznik animácie. Hlavný panel s názvom Tools (Obr. 1), ktorý kreslenie a editovanie objektov umožňuje, sa nachádza v ľavej časti obrazovky. Má 16 nástrojov, ktoré sa pri tvorbe a kreslení používajú. Ich ďalšie parametre, ktoré sa môžu nastaviť, ako je farba, či typ čiary, sa nachádzajú taktiež v tomto paneli, ale pod týmito nástrojmi.

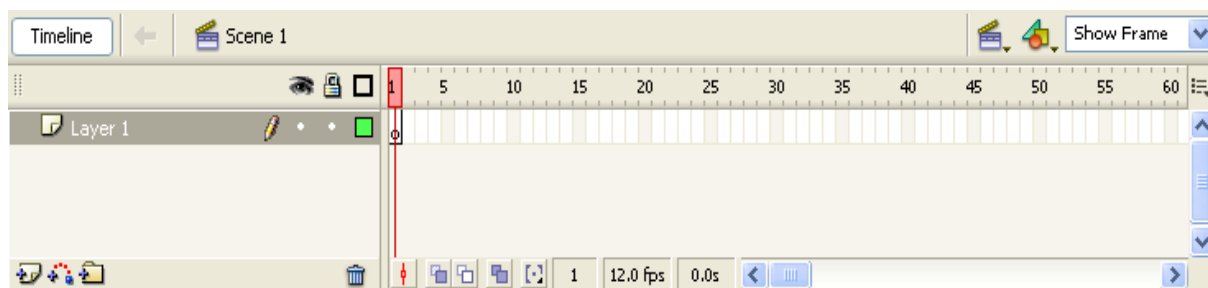


Obr. 1: Panel nástrojov

Jednou z najdôležitejších vecí pri vytváraní Flash animácií je Timeline, čiže časová os (Obr. 2). Slúži na ovládanie a organizovanie animácie v čase. Každá snímka je jedna kresba, kresby nasledujú v čase za sebou a tým vytvárajú určitý pohyb. [6]

Časovú os tvoria 2 základné časti. Na ľavej strane sú vrstvy (Layer), ktoré je možné premenovať, skryť, či zakázať editovanie... V jednotlivých vrstvách sa nachádza daná animácia, tlačidlo alebo nejaká statická grafika. Platí také pravidlo, že

v jednej vrstve môže prebiehať len animácia jedného objektu v ten istý časový okamžik, preto keď potrebujeme animovať viac objektov, používa sa viac vrstiev. [6] Druhú časť tvorí samotná časová os. Tu je podľa jednotlivých snímok vytvorená animácia alebo umiestnenie grafiky.



Obr. 2: Časová os – Timeline

1.5.1 História Flash

[7]

Začiatok Flash spadá asi do roku 1994. Nešlo o Flash ako ho poznáme dnes, ale volal sa SmartSketch a bol založený na Java. Táto aplikácia bola navrhnutá tak, aby vytváranie počítačovej grafiky bolo rovnako jednoduché ako kreslenie na papier.[5] Pomaly sa však od tohto smeru upustilo a Java ako programovací jazyk nevyhovoval nárokom na rýchlosť a spoľahlivosť. Keď niekedy okolo roku 1995 objavili prehliadač, ktorý podporoval zásuvné moduly typu PLUG-IN, bol SmartSketch premenovaný na FutureSplash Animator a bola úplne zmenená jeho podoba. Macromedia v dnešnej dobe pracuje so svojim projektom pod názvom Shockwave.

V roku 1996 Macromedia kúpila FutureSplash Animator a vznikol tak Macromedia Flash 1.0. Táto verzia neobsahovala ActionScript, ale usmerňovala vývoj webových animácií. Postupne vznikali ďalšie verzie, až po doterajšiu Adobe Flash CS4 Professional.

1.5.2 ActionScript

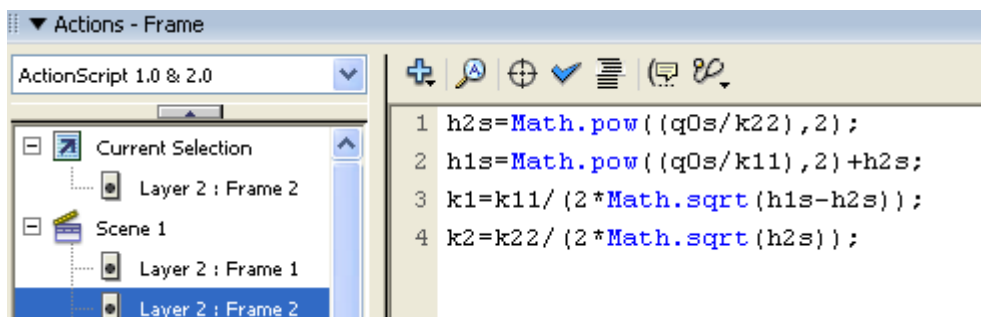
[8]

ActionScript je jednoduchý, objektovo orientovaný jazyk vytvorený pre písanie príkazov v programe Flash. Je hlavným nástrojom práce s objektmi Flash. Pomocou ActionScriptu môžeme ovládať jednotlivé objekty v scéne, ovládať prezentáciu ako celok, vytvárať formuláre, ale aj volať príkazy iných skriptovacích jazykov. ActionScript má niektoré spoločné vlastnosti s JavaScriptom.

Písanie skriptov je značne zjednodušené vďaka vývojovému prostrediu panelu Actions (Obr. 3) a píšú sa v panely ActionScript (Obr. 4). Príkazy sú prehľadne usporiadané, akciu môžeme priradiť objektu alebo rámčeku scény. Príkazy nemusíme ani poznať naspamäť, pretože priamo v menu vývojového prostredia sa nachádza prehľadný zoznam usporiadaný do niekoľkých kategórií. Pomocou ActionScriptu môžeme riadiť priebeh celého prehrávania.



Obr. 3: Vývojové prostredie Flash



Obr. 4: Panel ActionScript

1.5.3 História ActionScriptu

[9]

Prvá verzia ActionScriptu, označovaná ako ActionScript 1.0 bola navrhnutá, tak aby bola čo najjednoduchšia a mohli sa ju tak naučiť používať i začiatočníci bez väčších problémov. Obsahovala sadu intuitívnych príkazov a veľmi jednoduchú syntax. V tej dobe nemal ešte nikto tušenie, že sa Flash stane profesionálnym nástrojom pre vytváranie rozhraní a webových aplikácií. Ako náhle začali Flash používať skúsení programátori, došlo k vývoju oveľa sofistikovanejších druhov interaktivity, ktoré užívateľom umožnilo zadávať dáta alebo komunikovať so serverovými aplikáciami a posunuli tak hranice ActionScriptu. Pôvodne bol tento jazyk určený pre ovládanie jednoduchých 2D vektorových animácií, ale neskôršie verzie už umožňovali tvorbu Web-based hier a bohatých internetových aplikácií ako sú video alebo audio.

Vývojový tím ActionScriptu na túto skutočnosť zareagoval tak, že prvú verziu skriptovacieho jazyka zrevidoval a rozšíril do verzie ActionScript 2.0. Táto nová verzia priniesla skúseným programátorom celú radu potrebných nástrojov, zatiaľ čo zostala dostupná aj pre menej skúsených programátorov.

Keď bola pred niekoľkými rokmi vydaná súčasná verzia ActionScriptu 3.0, vzbudzovala medzi užívateľmi rozruch. Na rozdiel od predchádzajúcich verzií je totiž AS 3.0 mocným, objektovo orientovaným programovacím jazykom, ako sú napríklad C++ alebo Java.

Pozitívnu stránkou AS 3.0 je, že sa vďaka nemu Flash stal funkčne schopným nástrojom k vývoju sofistikovaných aplikácií nielen pre web, ale s pomocou Adobe AIR takisto pre desktopové aplikácie. Za negatívum ActionScript 3.0 je to, že sa značne vzdialil začiatočníkom. Pokiaľ nepoznajú princípy a techniky objektovo orientovaného programovania, nebudú toho môcť veľmi veľa robiť.

2. Praktická časť

Cieľom práce bolo vytvorenie piatich animácií modelov rôznych procesov. Animácie budú slúžiť študentom pri poznávaní neznámych procesov, boli vytvorené pomocou programu Macromedia Flash 8 a pomocou skriptovacieho jazyka ActionScript 1.0, 2.0. Animácie pomôžu študentom pri študovaní, poznávaní neznámych procesov a pomôžu im veľmi rýchlo pochopiť ako daný proces funguje. Študenti získajú potrebné vedomosti a zručnosti, ktoré neskôr môžu využiť pri riadení reálnych zariadení.

2.1 Vlastnosti projektov

verzia programu	Macromedia Flash 8
verzia ActionScript	1.0, 2.0
veľkosť klipu	800x600
typ súboru kompilácie	.fla
typ súboru publikácie	.swf
počet vrstiev	2
počet snímiek	6
počet súborov	5 – swf., 5 – fla.

2.2 Typy modelov

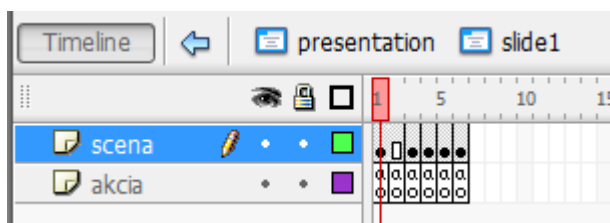
Modely, na ktoré som sa v tejto práci zamerala, boli nasledovné:

- Zásobník kvapaliny.
- Dva zásobníky kvapaliny s interakciou.
- Tri zásobníky kvapaliny bez interakcie.
- Prietokový ohrievač.
- Miešanie dvoch prúdov s rôznou teplotou.

2.3 Tvorba animácií

Prvou dôležitou vecou bolo navrhnutie výslednej podoby animácie. Dôležité je aj to, aby boli animácie zrozumiteľné aj pre iných používateľov, ktorí ich budú využívať. Bolo treba naštudovať správanie sa systému a na základe toho spraviť animácie.

Pri zostavovaní modelu procesu je potrebné najskôr určiť známe konštanty alebo parametre, ktoré sú nemenné. Táto nemennosť môže vyplývať napríklad z rozmerov zariadenia, z konštantných fyzikálnych vlastností, ako napríklad hustota, objem, objemový prietok... Druhým krokom je určenie začiatočných hodnôt, ktoré sa v ďalších krokoch budú získavať riešením rovníc modelu (teplota, výšky hladín) pomocou metódy Runge-Kutta. V poslednom kroku je možnosť vidieť výstupnú veličinu a jej grafický priebeh. Toto všetko bolo zhrnuté vo vrstvách a jednotlivých snímkoch animácií. Každá z týchto animácií obsahuje 2 vrstvy a 6 snímkov (Obr. 5).



Obr. 5: Vrstvy a snímky animácií

2.3.1 Grafika animácií

Všetky animácie sú vytvorené rovnakým štýlom. Pozostávajú z dvoch častí:

- Vrchná časť obsahuje zadávané hodnoty, alebo tie hodnoty, ktoré sa vypočítavajú.
- Spodná časť obsahuje vizuálne nahliadnutie na jednotlivé modely: ako sa mení ich výstupná veličina, jej grafický priebeh, náhľad do zariadení, sledovanie zmien hladín v zásobníkoch a zmien teplôt v miešač, či ohrievači.

- Jednotlivé fázy animácie:
 - a) načítanie nami zvolených hodnôt, ktoré si sami určíme a sú nemenné počas celého priebehu animácie.
 - b) výpočet rovnovážneho stavu modelov podľa príslušných vzorcov.
 - c) voľba jednotlivých parametrov: začiatočná podmienka pre riešenie diferenciálnych rovníc a čas priebehu.
 - d) priebeh animácie: graf, zmena v zariadení (výška hladiny, teplota)

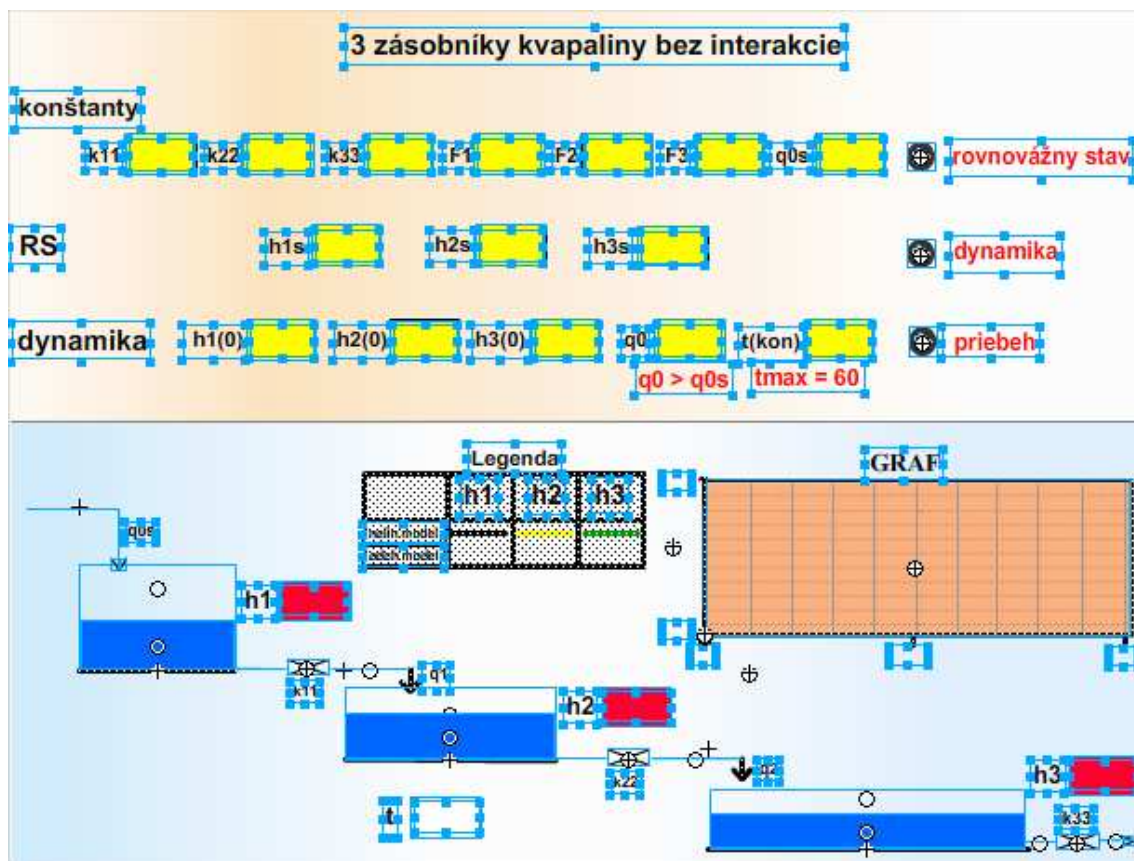
Grafické rozdelenie, 4 fázy animácie sú pre všetky modely rovnaké, rozdiel je len v konštantách a výpočtoch.

- Pri zásobníkoch kvapaliny sa po načítaní jednotlivých hodnôt (fáza a, b, c) a po stlačení tlačidla priebeh, zobrazí animácia zásobníkov. Tak ako sa budú meniť hodnoty výšky hladín, tak sa bude meniť aj hladina v zásobníkoch. Pri rastúcej hodnote výstupnej veličiny (hladiny), bude výška hladiny narastať a pri klesajúcej hodnote, bude výška hladiny klesať.
- Pri ohrievači a miešači bude animácia nasledovná. Ako pri zásobníkoch, tak aj pri týchto dvoch animáciách, sa po načítaní jednotlivých hodnôt (fáza a, b, c) a po stlačení tlačidla priebeh zobrazí animácia miešača, či ohrievača. Pri rastúcej hodnote výstupnej veličiny (teploty) sa pri zvyšujúcej teplote bude meniť farba zariadenia z modrej farby na červenú (ohrievanie) a pri znižujúcej teplote sa bude meniť farba zariadenia z červenej farby na modrú (chladenie).

2.3.2 Vrstvy animácie

Vrstvy animácie:

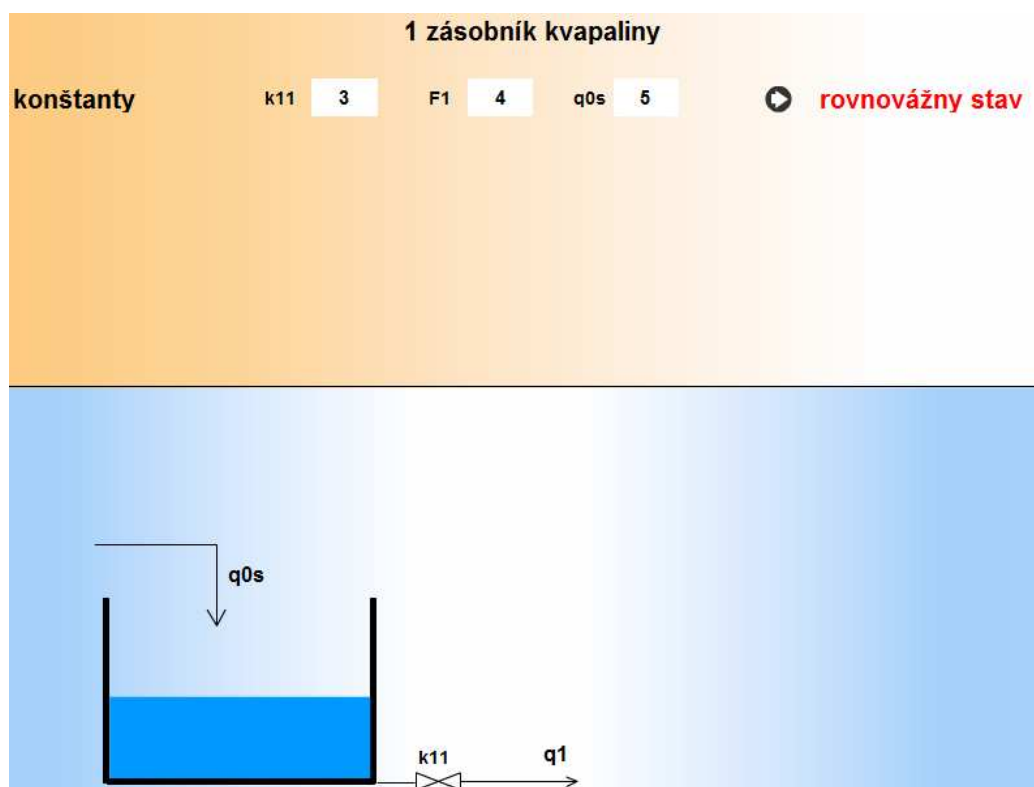
1. objekty v scéne (Obr. 6),
2. ActionScript.



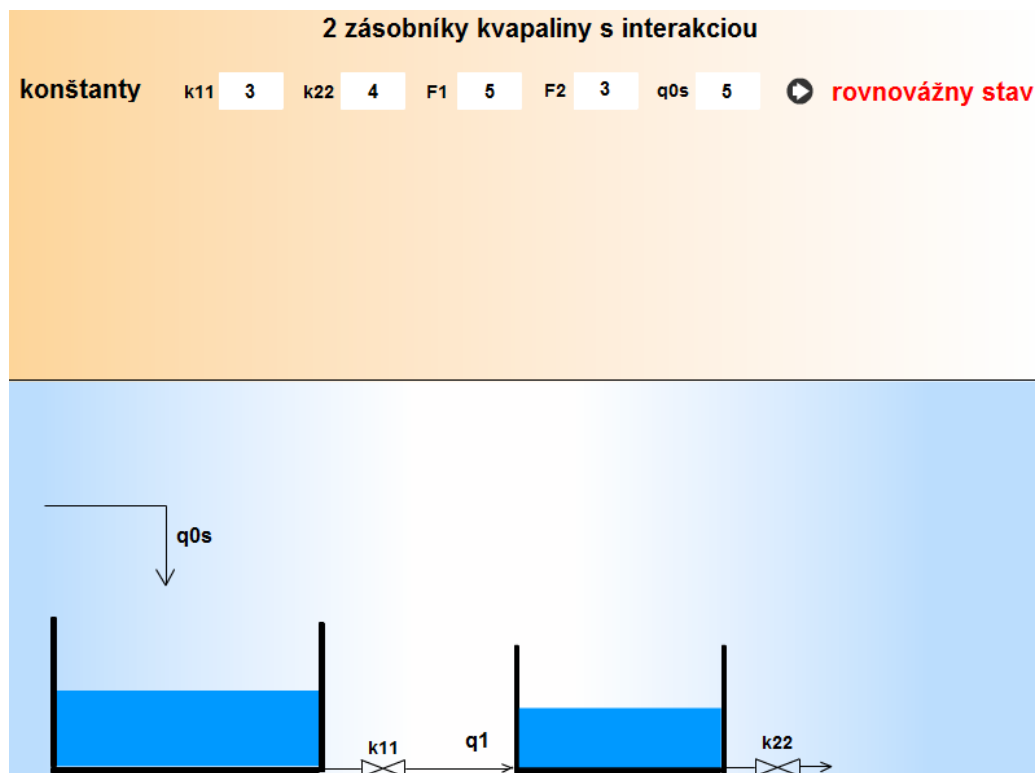
Obr. 6: Objekty na scéne animácie

1. vrstva animácií - objekty

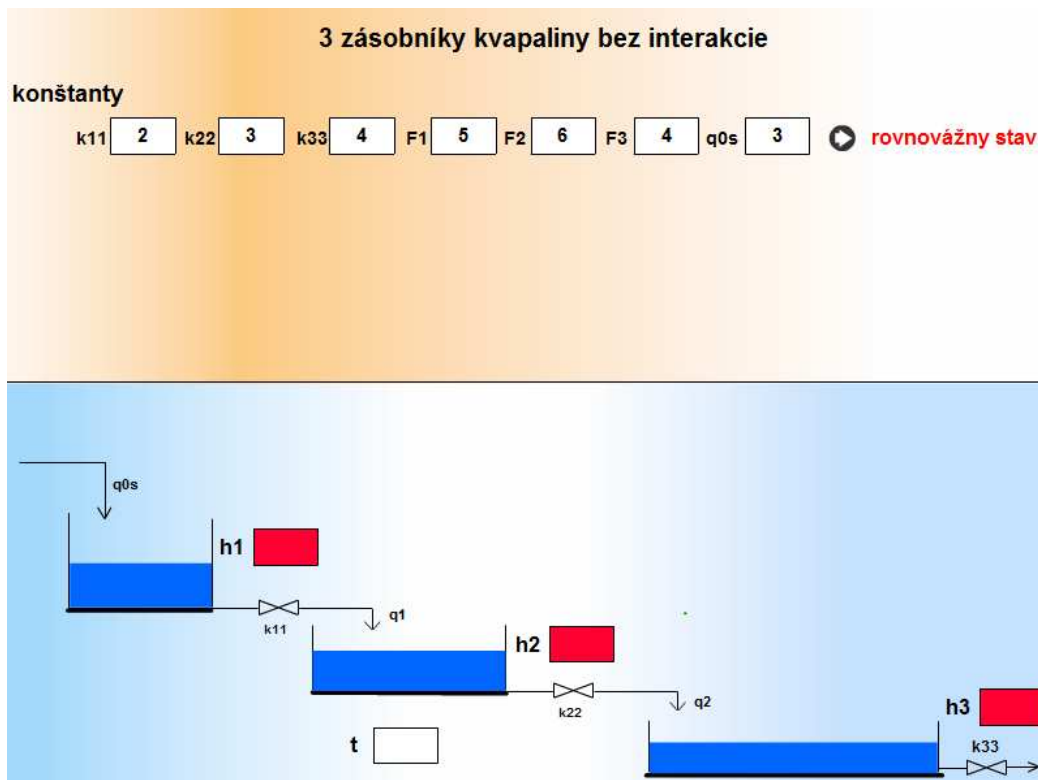
- Snímka 1,2 (Obr.7, 8, 9, 10, 11) – zadávanie príslušných konštánt.
- Snímka 3 (Obr.12, 13, 14, 15, 16) – výpočet rovnovážnych stavov.
- Snímka 4 – dynamika, zadanie počiatočného stavu veličiny a času priebehu animácie.
- Snímka 5 (Obr.17, 18, 19, 20, 21) – výpočet výstupnej veličiny, zobrazenie priebehu výstupnej veličiny na grafe.
- Snímka 6 – slučka na snímku 5.



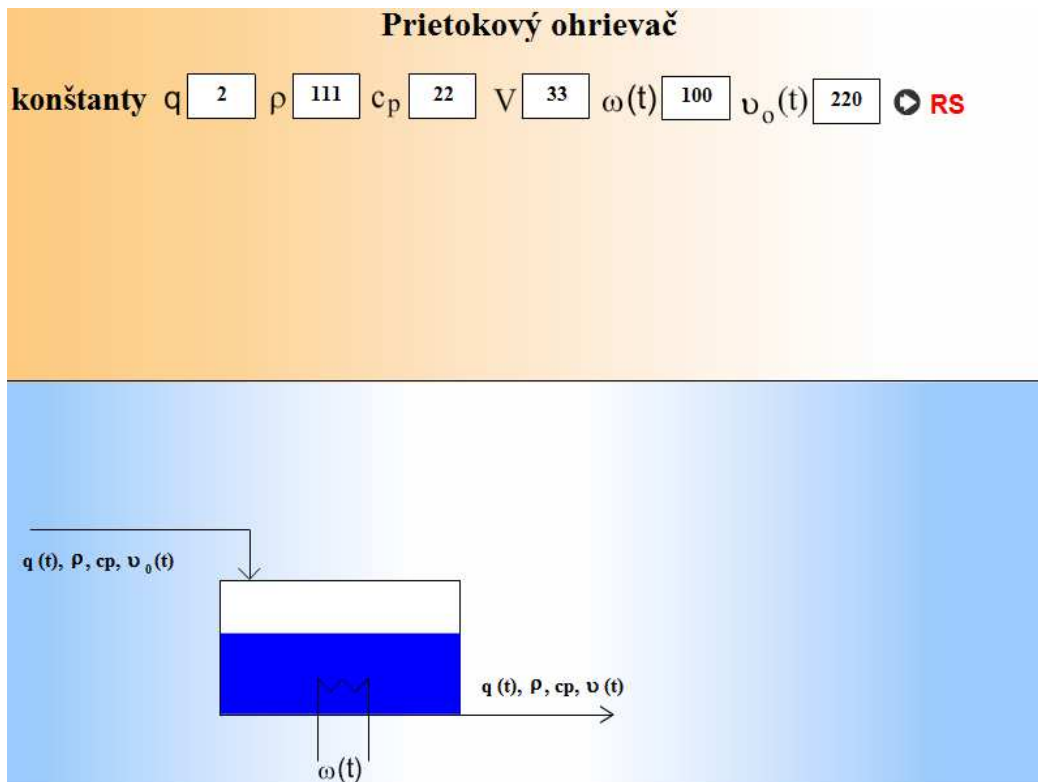
Obr. 7: Zadávanie konštánt - 1 zásobník kvapaliny



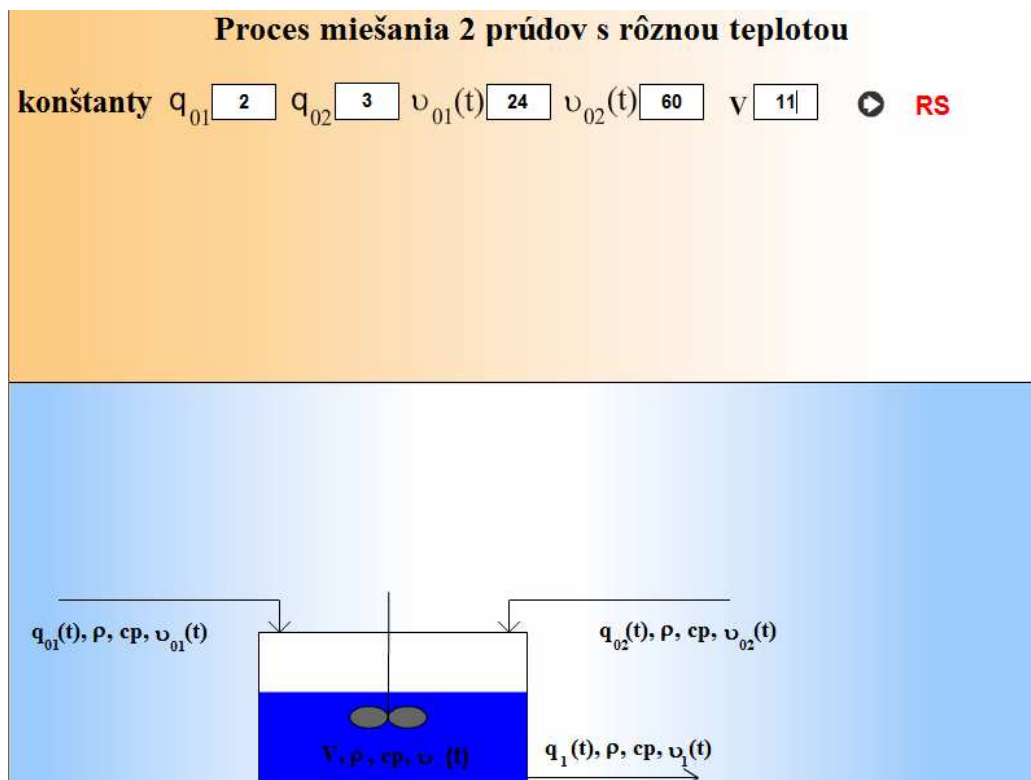
Obr. 8: Zadávanie konštánt - 2 zásobníky kvapaliny



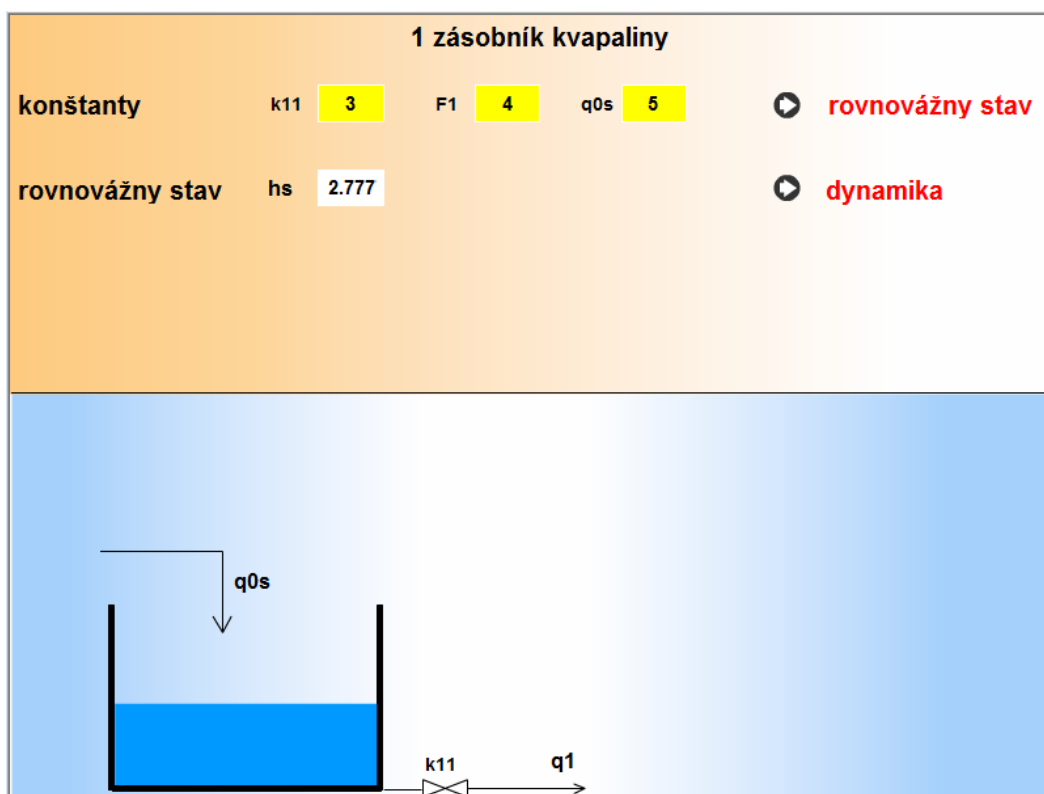
Obr. 9: Zadávanie konštánt - 3 zásobníky kvapaliny



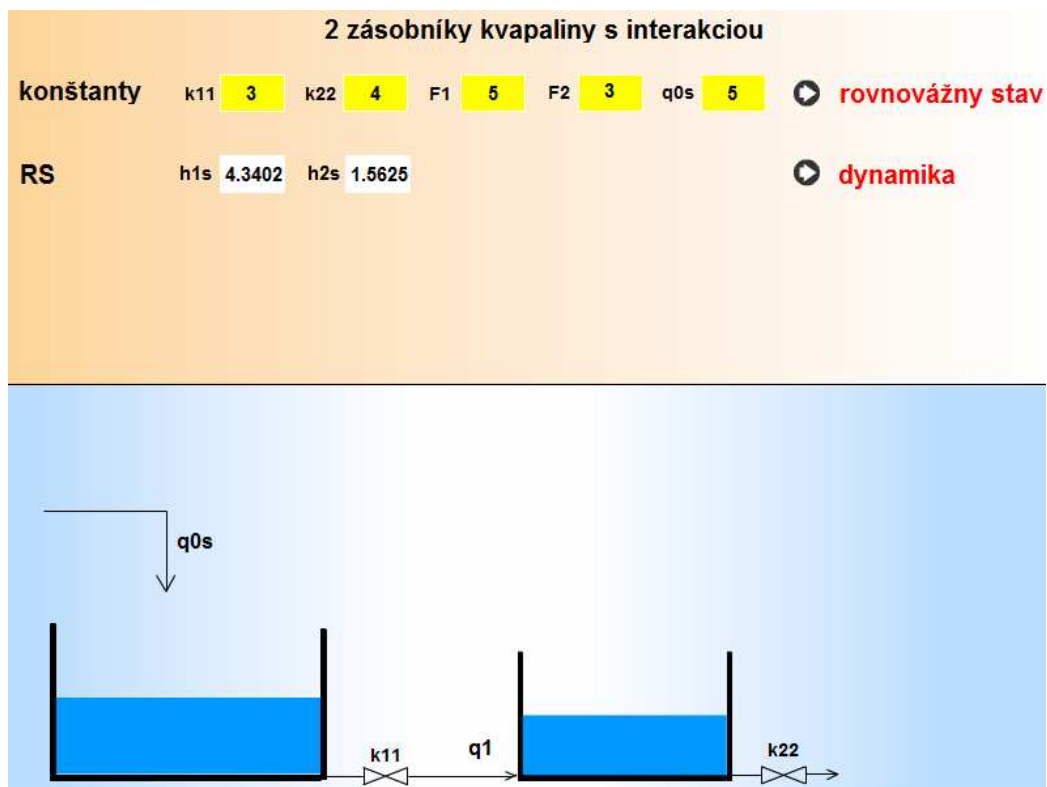
Obr. 10: Zadávanie konštánt - prietokový ohrievač



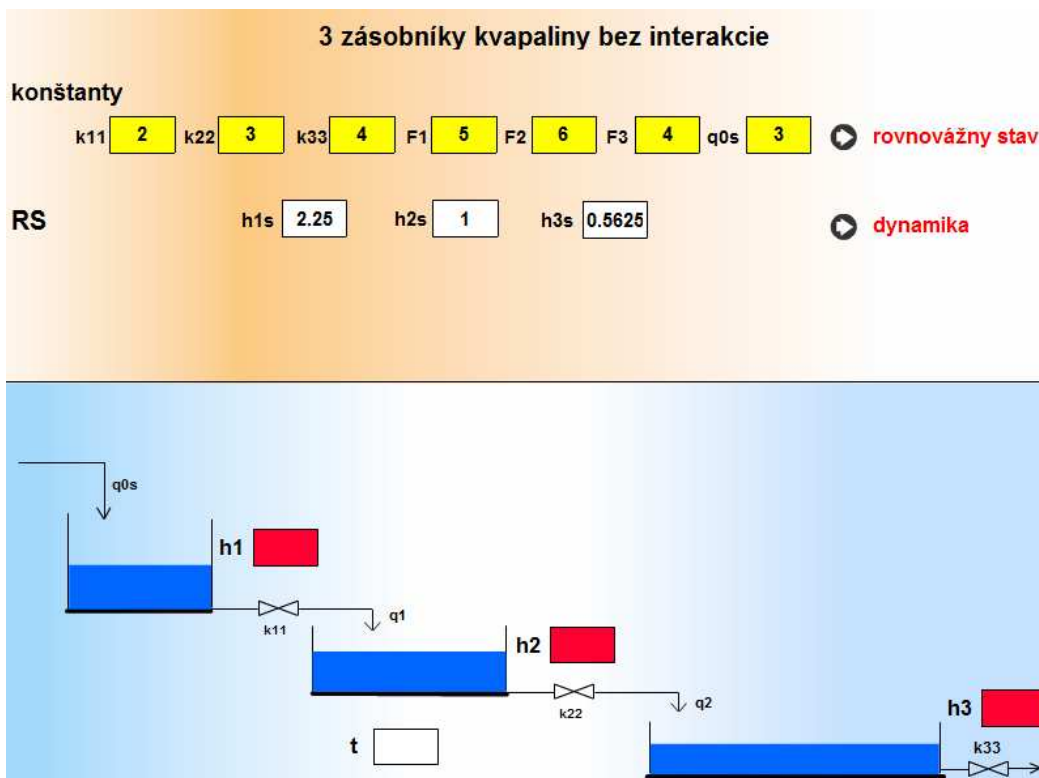
Obr. 11: Zadávanie konštánt - proces miešania 2 prúdov s rôznou teplotou



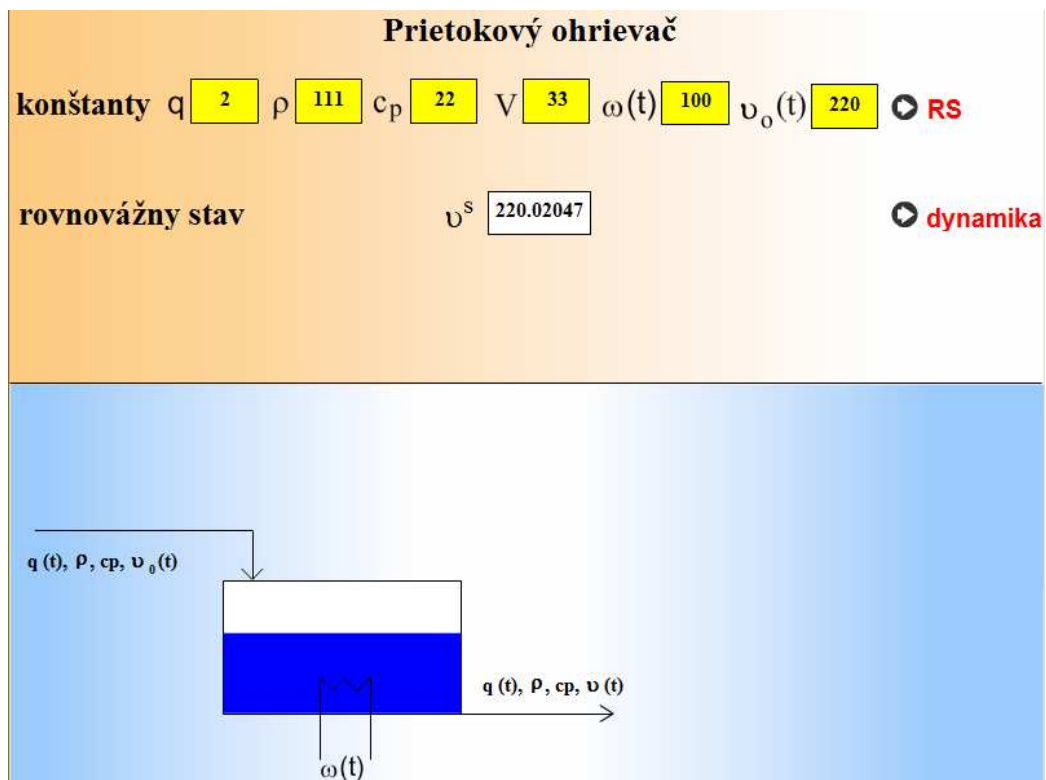
Obr. 12: Výpočet rovnovážneho stavu - 1 zásobník kvapaliny



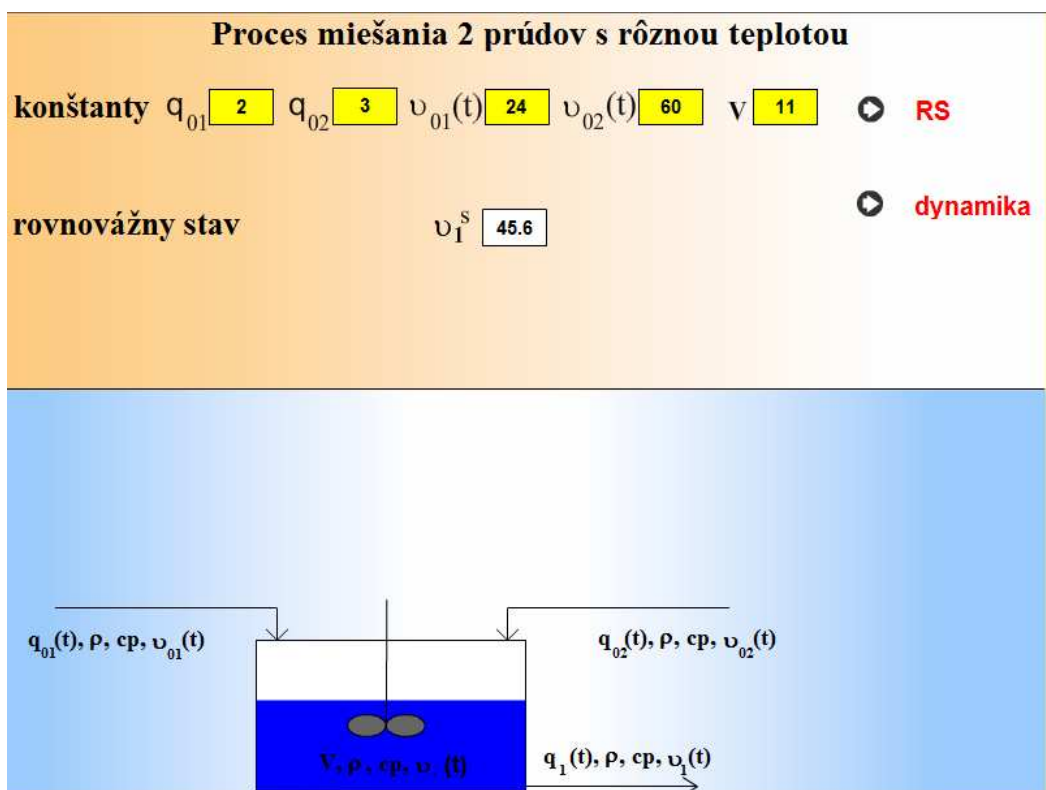
Obr. 13: Výpočet rovnovážneho stavu - 2 zásobníky kvapaliny



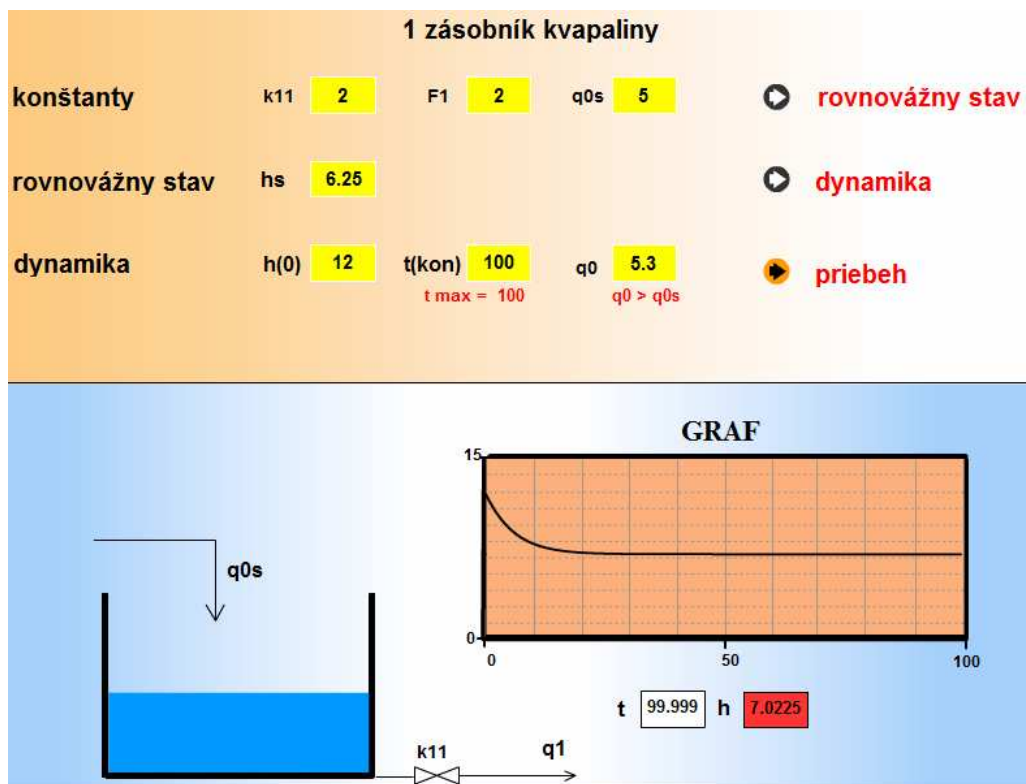
Obr. 14: Výpočet rovnovážneho stavu - 3 zásobníky kvapaliny



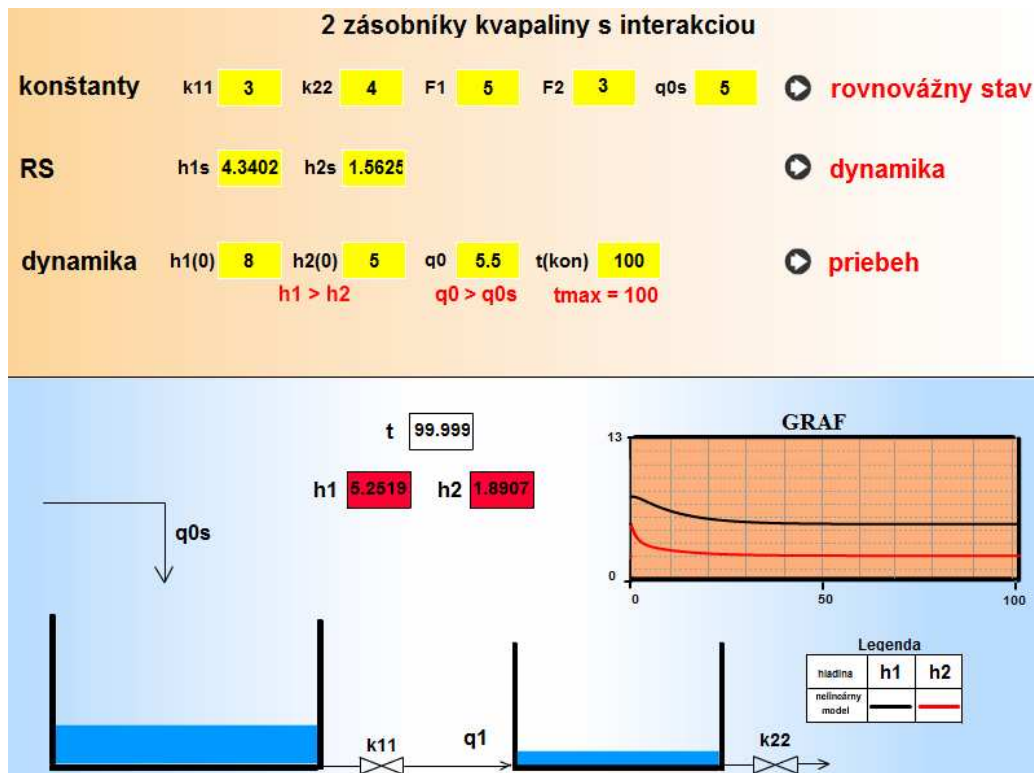
Obr. 15: Výpočet rovnovážneho stavu - prietokový ohrievač



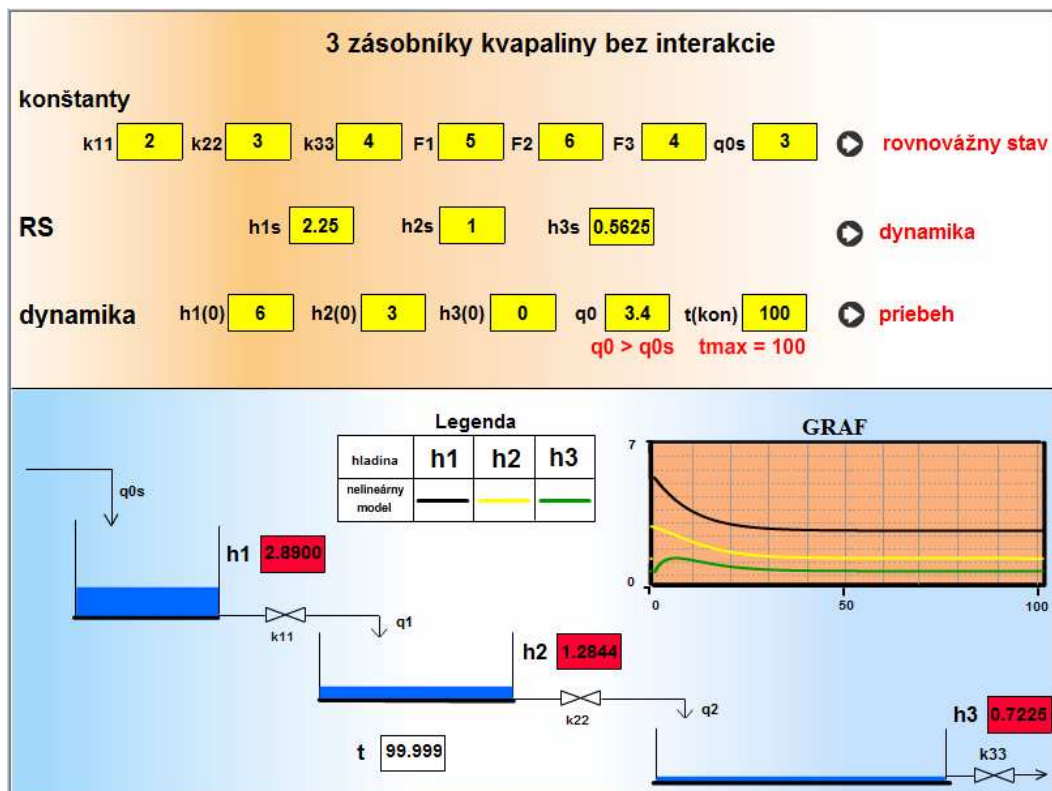
Obr. 16: Výpočet rovnovážneho stavu - proces miešania 2 prúdov s rôznou teplotou



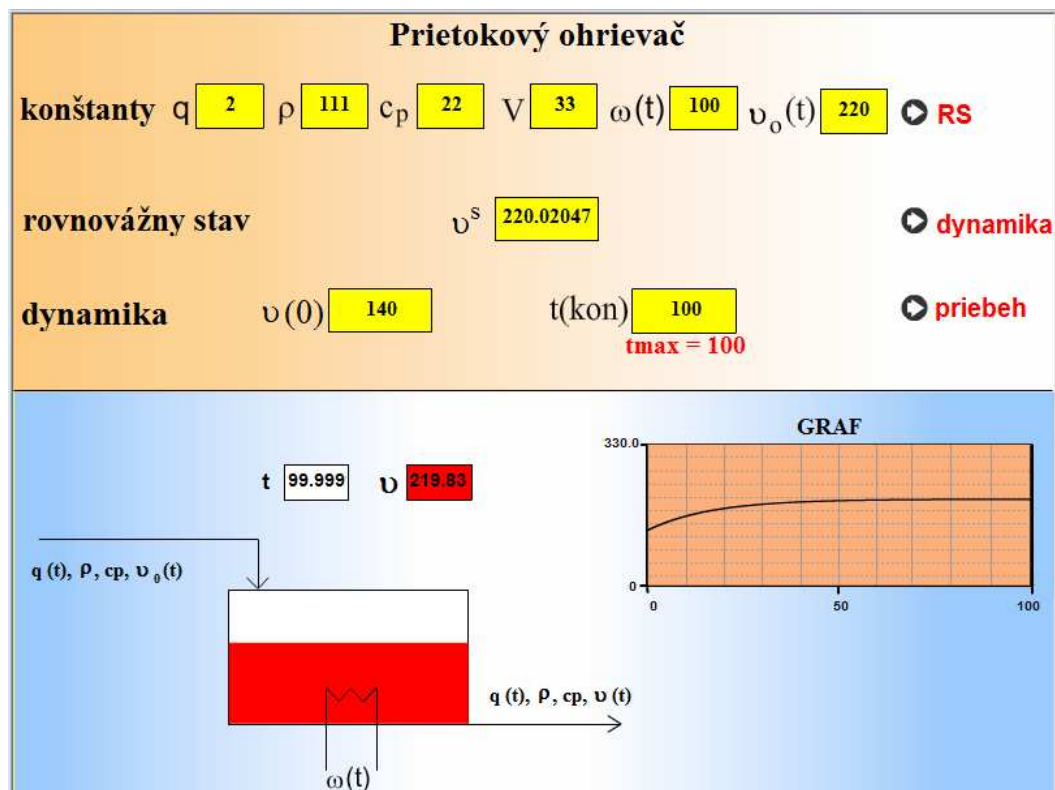
Obr. 17: Výsledná animácia jedného zásobníka kvapaliny



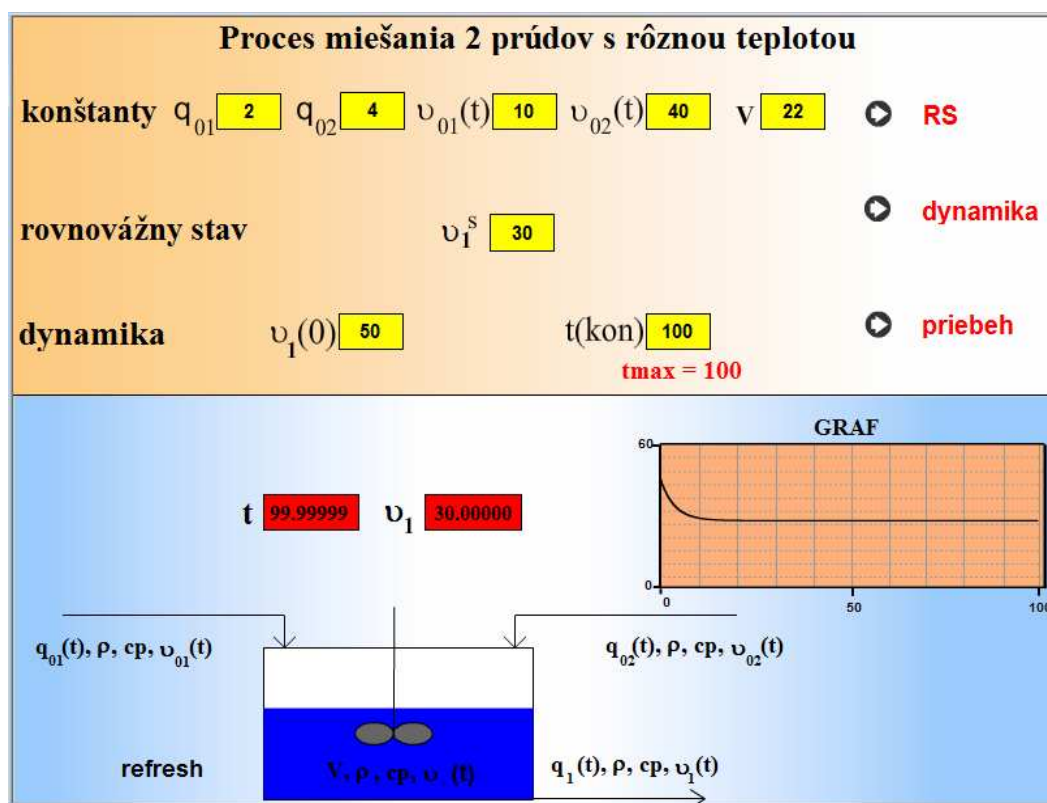
Obr. 18: Výsledná animácia dvoch zásobníkov kvapalín s interakciou



Obr. 19: Výsledná animácia troch zásobníkov kvapaliny bez interakcie



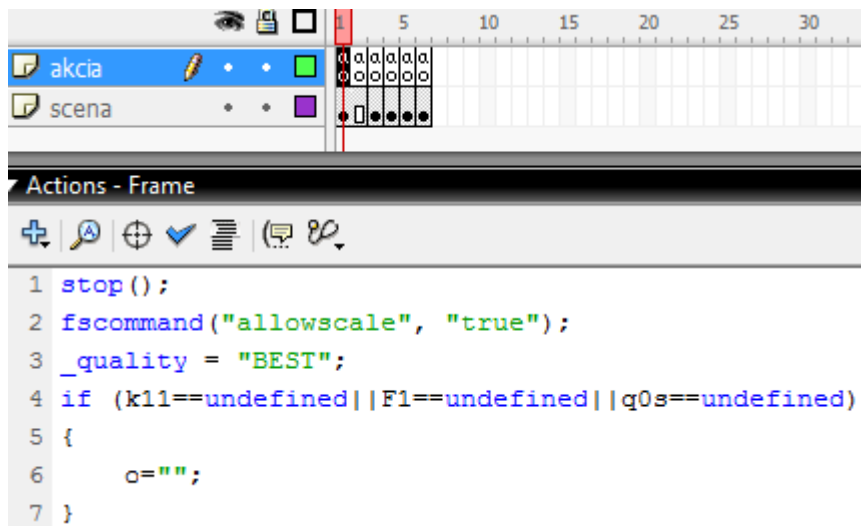
Obr. 20: Výsledná animácia prietokového ohrievača



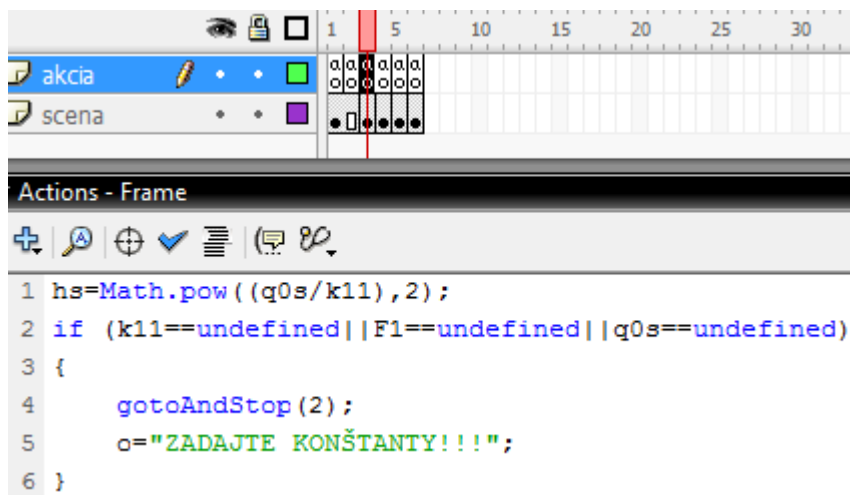
Obr. 21: Výsledná animácia procesu miešania 2 prúdov s rôznou teplotou

2. vrstva animácií - ActionScript

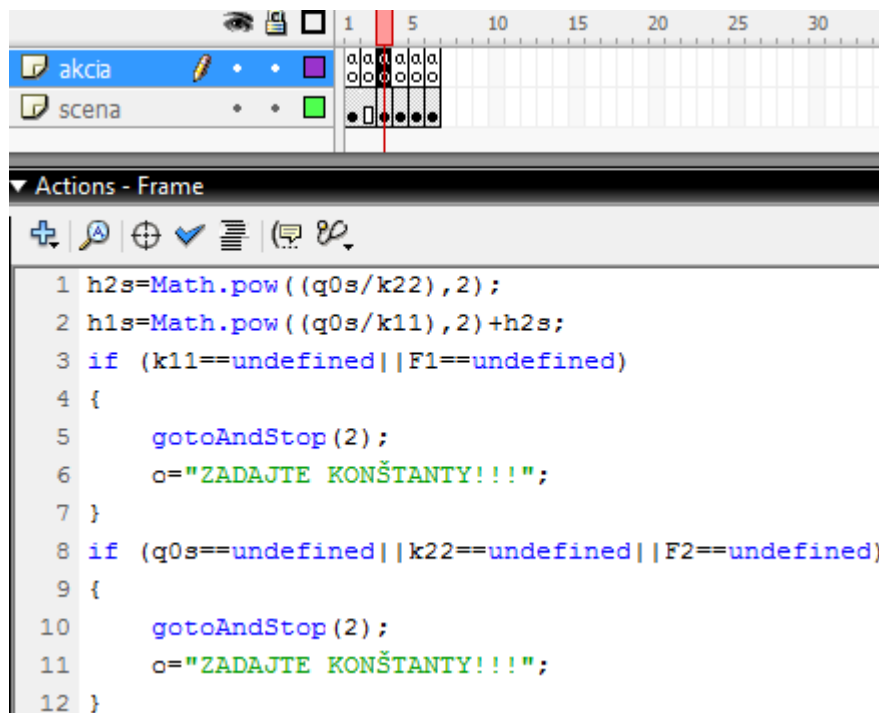
- Snímka 1, 2 (Obr. 22) – zastavenie časovej osi, nastavenie kvality a rozlíšenia.
- Snímka 3 (Obr. 23, 24, 25, 26, 27) – výpočet rovnovážnych stavov a zobrazenie oznamu, ak nie sú zadefinované konštanty.
- Snímka 4 (Obr. 28, 29, 30, 31, 32) – zadefinovanie:
 - konštanty 1, potrebnú pri výpočte metódy Runge-Kutta,
 - začiatočného času $t = 0$,
 - hodnôt a_1 , a_3 , a_4 , v , u , nula, bnx ($n = 1, 2, 3$) a maximálnych hodnôt veličín potrebných pre graf.
- Snímka 5 (Obr. 33, 34, 35, 36, 37) – metóda Runge-Kutta, výpočet a grafické zobrazenie výstupnej veličiny.
- Snímka 6 (Obr. 38) – slučka na snímku 5.



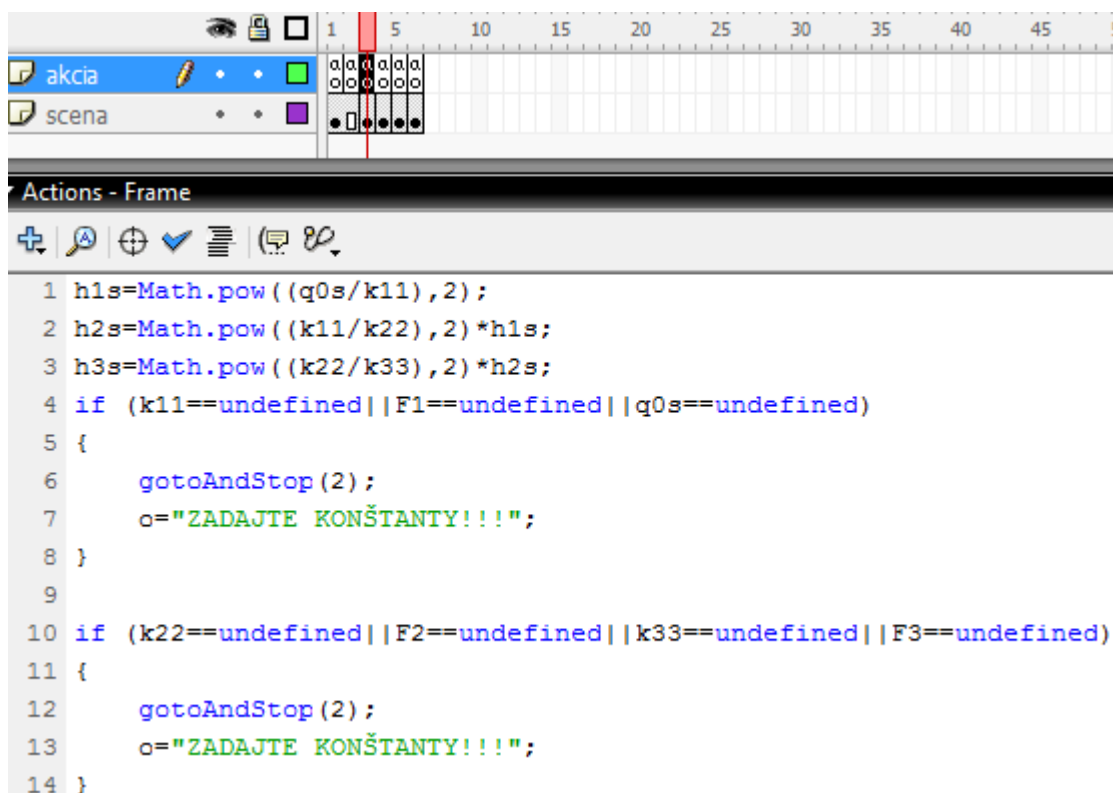
Obr. 22: Zastavenie časovej osi, nastavenie kvality a rozlíšenia



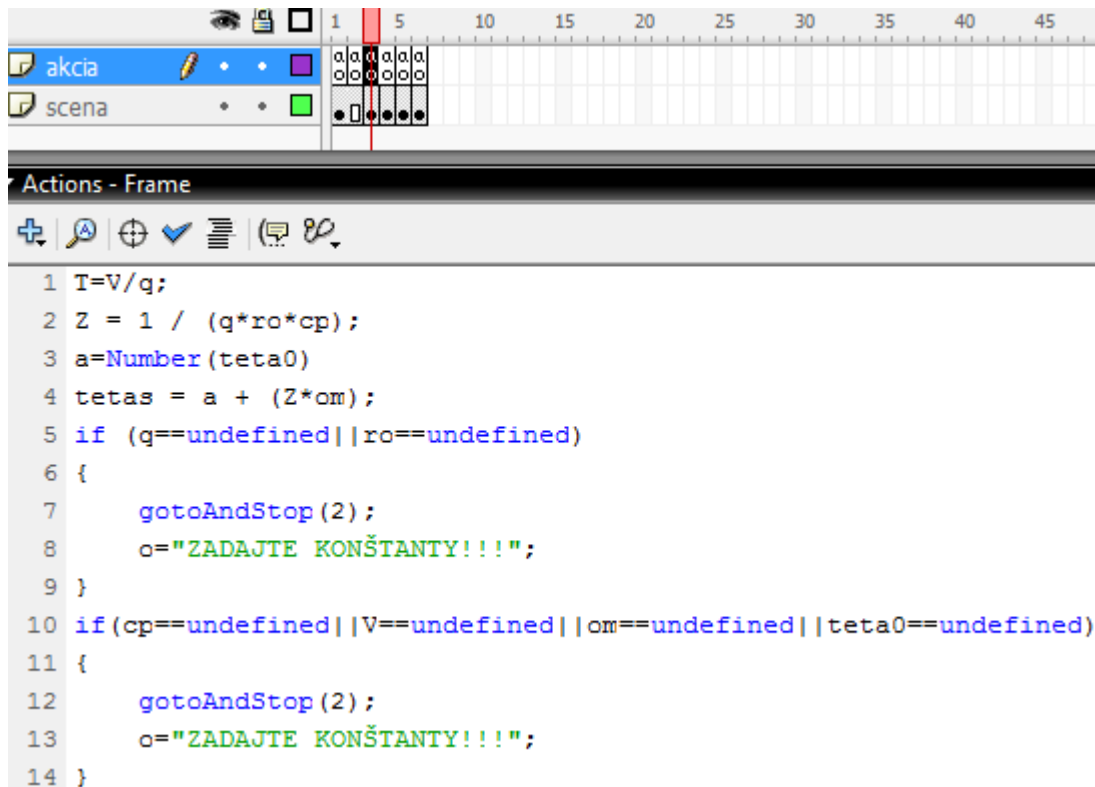
Obr. 23: Výpočet rovnovážneho stavu pre 1 zásobník kvapaliny



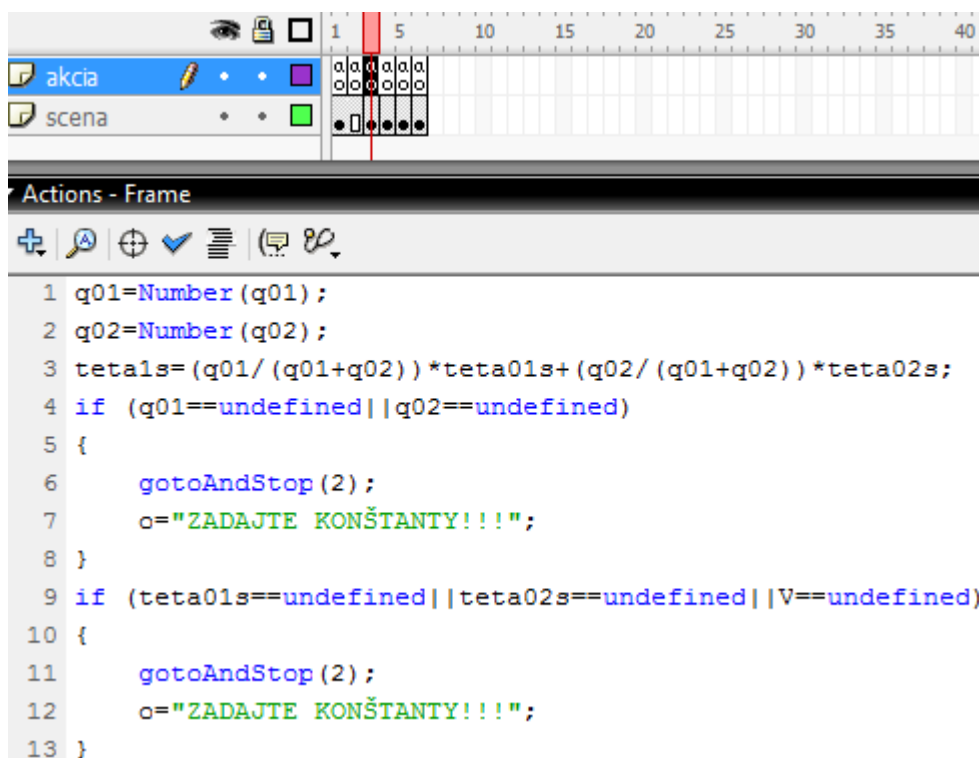
Obr. 24: Výpočet rovnovážných stavov pre 2 zásobníky kvapaliny



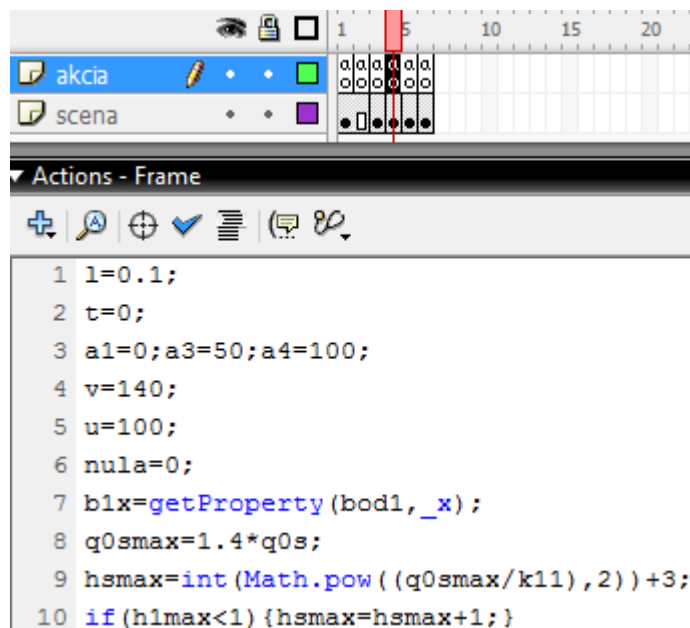
Obr. 25: Výpočet rovnovážných stavov pre 3 zásobníky kvapaliny



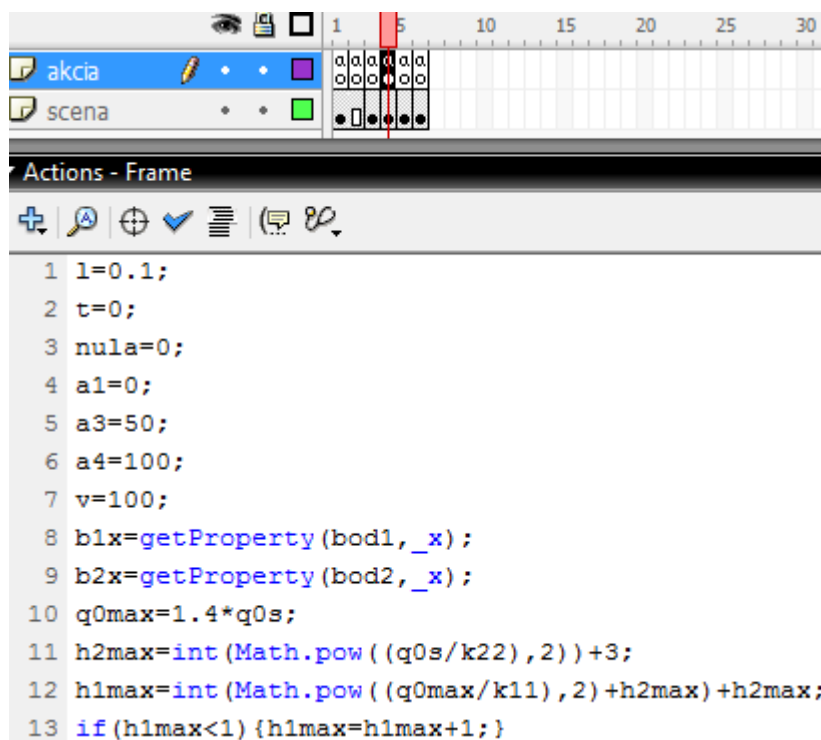
Obr. 26: Výpočet rovnovážneho stavu pre ohrievač



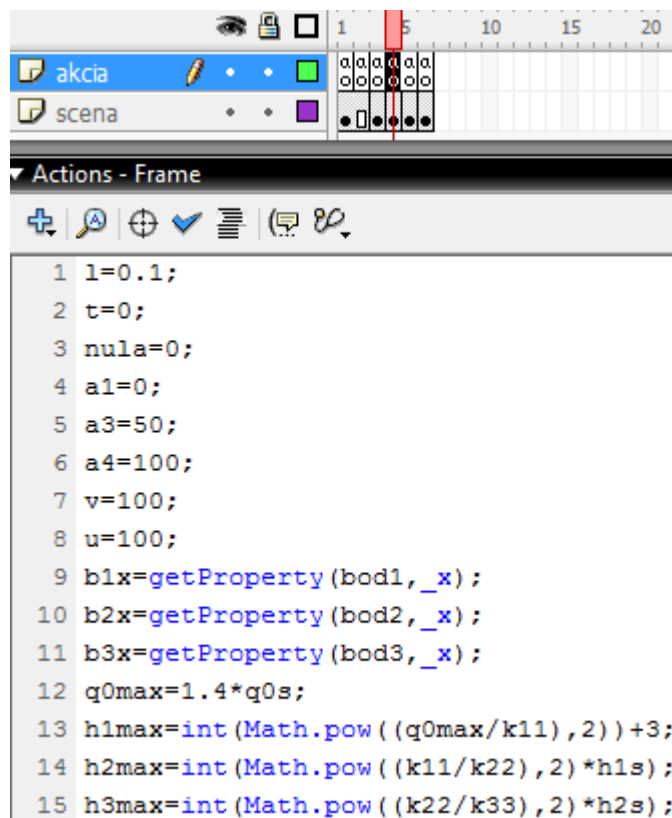
Obr. 27: Výpočet rovnovážneho stavu pre miešač



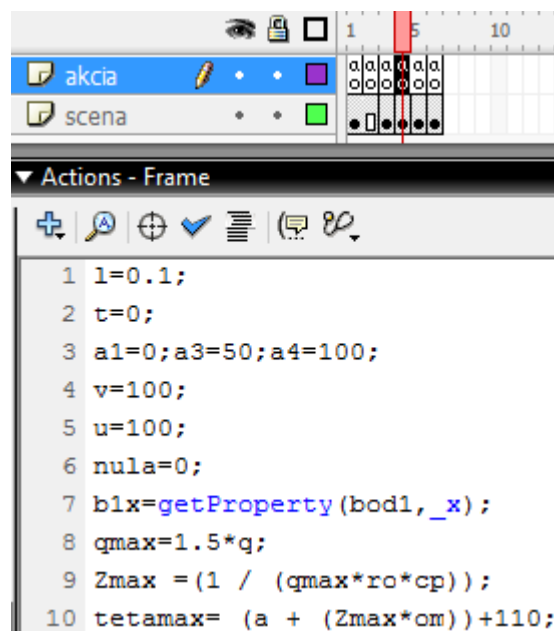
Obr. 28: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 1 zásobník kvapaliny



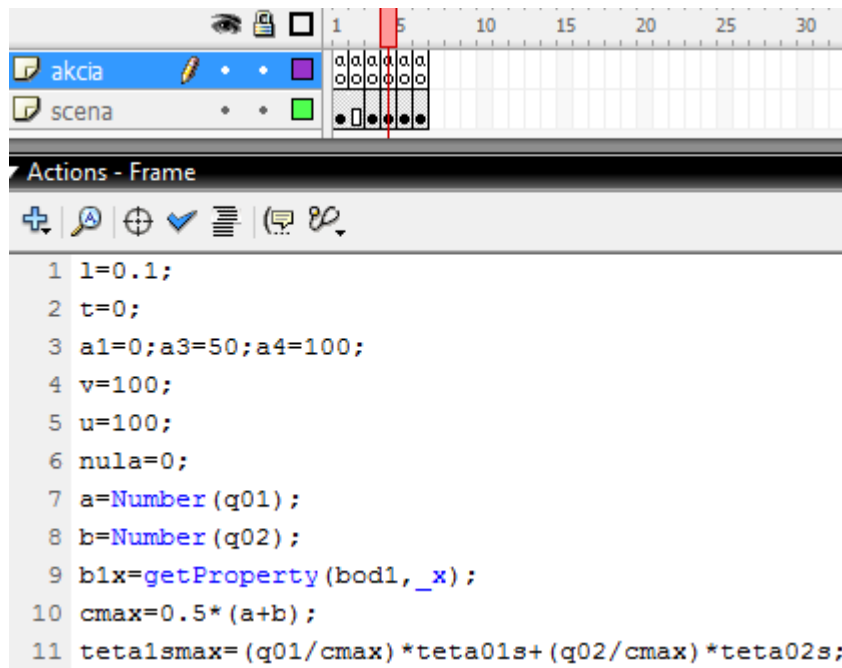
Obr. 29: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 2 zásobníky kvapaliny



Obr. 30: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre 3 zásobníky kvapaliny



Obr. 31: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre ohrievač



Obr. 32: Zadefinovanie konštánt potrebných pre výpočet Runge-Kutta a graf pre miešač

```

1  if (t == 0) {
2      x1 = Number(h0);
3  }
4  //zastavenie priebehu
5  if (t>=tkon-0.15) {
6      stop();
7  }
8  t = t+1;
9
10 //rungekutta 1.rad
11 x1_old = x1;
12 f=(q0/F1)-(k11/F1)*Math.sqrt(x1);
13 K1 = 1*f;
14 x1 = x1_old+0.5*K1;
15 f=(q0/F1)-(k11/F1)*Math.sqrt(x1);
16 K2 = 1*f;
17 x1 = x1_old+0.5*K2;
18 f=(q0/F1)-(k11/F1)*Math.sqrt(x1);
19 K3 = 1*f;
20 x1 = x1_old+K3;
21 f=(q0/F1)-(k11/F1)*Math.sqrt(x1);
22 K4 = 1*f;
23 x1 = x1_old+(K1+2*(K2+K3)+K4)/6;
24 h = x1;
25
26 hn=h/hsmax*v;
27
28 if(i==undefined)i=1;
29 setProperty(bod1,_y,-hn+185.7);
30 duplicateMovieClip("bod1","bod1"+i,2*i);
31 setProperty("bod1"+i,_y,bod1._y);
32 setProperty("bod1"+i,_x,bod1._x+t*3.65);
33 h1x=getProperty("bod1"+i,_x);
34 h1y=getProperty("bod1"+i,_y)+10;
35 setProperty(l1,_x,h1x);
36 setProperty(l1,_y,h1y);
37 i++;
38
39 this.hlad._yscale =800*h/hsmax;

```

Obr. 33: Runge – Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie hladiny 1 zásobníka kvapaliny

```

1  if(t==0)
2  {
3      x1=Number(h10);
4      x2=Number(h20);
5  }
6  //zastavenie priebehu
7  if (t>=tkon-0.15)
8  {
9      stop();
10 }
11 t=t+0.1;
12
13 //rungekutta 1
14 x1_old = x1;
15 f1 = (q0 - k11* Math.sqrt(x1-x2))/F1;
16 K1 = 1*f1;
17
18 x1=x1_old + 0.5*K1;
19 f1 = (q0 - k11* Math.sqrt(x1-x2))/F1;
20 K2 = 1*f1;
21
22 x1 = x1_old + 0.5*K2;
23 f1 = (q0 - k11* Math.sqrt(x1-x2))/F1;
24 K3 = 1*f1;
25
26 x1 = x1_old + K3;
27 f1 = (q0 - k11* Math.sqrt(x1-x2))/F1;
28 K4 = 1*f1;
29
30 x1 = x1_old + (K1+2*(K2+K3)+K4)/6;
31 h1=x1;
32
33 //rungekutta 2
34 x2_old = x2;
35 f2 = (k11*Math.sqrt(x1-x2)-(k22*Math.sqrt(x2)))/F2;
36
37 K11 = 1*f2;
38
39 x2=x2_old + 0.5*K11;
40 f2 = (k11*Math.sqrt(x1-x2)-(k22*Math.sqrt(x2)))/F2;
41 K22 = 1*f2;
42
43 x2 = x2_old + 0.5*K22;
44 f2 = (k11*Math.sqrt(x1-x2)-(k22*Math.sqrt(x2)))/F2;
45 K33 = 1*f2;
46
47 x2 = x2_old + K33;
48 f2 = (k11*Math.sqrt(x1-x2)-(k22*Math.sqrt(x2)))/F2;
49 K44 = 1*f2;

```

```

50
51 x2 = x2_old + (K11+2*(K22+K33)+K44)/6;
52
53 h2=x2;
54
55 hn1=h1/h1max*v;
56 hn2=h2/h1max*v;
57
58 //-----GRAF-----
59 if(i==undefined)i=1;
60 setProperty(bod1,_y,-hn1+143);
61 duplicateMovieClip("bod1","bod1"+i,2*i);
62 setProperty("bod1"+i,_y,bod1._y);
63 setProperty("bod1"+i,_x,bod1._x+t*3);
64 h1x=getProperty("bod1"+i,_x);
65 h1y=getProperty("bod1"+i,_y)+10;
66 setProperty(l1,_x,h1x);
67 setProperty(l1,_y,h1y);
68
69 if(i==undefined)i=1;
70 setProperty(bod2,_y,-hn2+80);
71 duplicateMovieClip("bod2","bod2"+i,2*i+1);
72 setProperty("bod2"+i,_y,bod2._y);
73 setProperty("bod2"+i,_x,bod2._x+t*3);
74 h2x=getProperty("bod2"+i,_x);
75 h2y=getProperty("bod2"+i,_y)+10;
76 setProperty(l2,_x,h2x);
77 setProperty(l2,_y,h2y);
78 i++;
79
80 this.hlاد._yscale=1000*h1/h1max;
81 this.hlاد1._yscale=1000*h2/h1max;
82
83 if (h10 < h20)
84 {
85     oznam="h1 musí byt > ako h2";
86     stop();
87 }
88 if (h10 >= h20)
89 {
90     oznam="";
91 }

```

**Obr. 34: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie
hladín 2 zásobníkov kvapalín**


```

1  if (t==0)
2  {
3      x1=Number(h10);
4      x2=Number(h20);
5      x3=Number(h30);
6  }
7
8  //zastavenie priebehu
9  if (t>=tkon-0.15)
10 {
11     stop();
12 }
13
14 t=t+0.1;
15
16 //rungekutta 1
17 x1_old = x1;
18 f1 = (q0 - k11* Math.sqrt(x1))/F1;
19 K1 = 1*f1;
20
21 x1=x1_old + 0.5*K1;
22 f1 = (q0 - k11* Math.sqrt(x1))/F1;
23 K2 = 1*f1;
24
25 x1 = x1_old + 0.5*K2;
26 f1 = (q0 - k11* Math.sqrt(x1))/F1;
27 K3 = 1*f1;
28
29 x1 = x1_old + K3;
30 f1 = (q0 - k11* Math.sqrt(x1))/F1;
31 K4 = 1*f1;
32
33 x1 = x1_old + (K1+2*(K2+K3)+K4)/6;
34 h1=x1;
35
36 //rungekutta 2
37
38 x2_old = x2;
39 f2 = (k11*Math.sqrt(x1)-k22*Math.sqrt(x2))/F2;
40
41 L1 = 1*f2;
42
43 x2=x2_old + 0.5*L1;
44 f2 = (k11*Math.sqrt(x1)-k22*Math.sqrt(x2))/F2;
45
46 L2 = 1*f2;
47
48 x2 = x2_old + 0.5*L2;f2 = (k11*Math.sqrt(x1)-k22*Math.sqrt(x2))/F2;

```

```

49 L3 = 1*f2;
50
51 x2 = x2_old + L3;
52 f2 = (k11*Math.sqrt(x1)-k22*Math.sqrt(x2))/F2;
53 L4 = 1*f2;
54
55 x2 = x2_old + (L1+2*(L2+L3)+L4)/6;
56
57 h2=x2;
58
59 //rungekutta 3
60
61 x3_old = x3;
62 f3 = (k22*Math.sqrt(x2)-k33*Math.sqrt(x3))/F3;
63 M1 = 1*f3;
64
65 x3=x3_old + 0.5*M1;
66 f3 = (k22*Math.sqrt(x2)-k33*Math.sqrt(x3))/F3;
67 M2 = 1*f3;
68
69 x3 = x3_old + 0.5*M2;
70 f3 = (k22*Math.sqrt(x2)-k33*Math.sqrt(x3))/F3;
71 M3 = 1*f3;
72
73 x3 = x3_old + M3;
74 f3 = (k22*Math.sqrt(x2)-k33*Math.sqrt(x3))/F3;
75 M4 = 1*f3;
76
77 x3 = x3_old + (M1+2*(M2+M3)+M4)/6;
78
79 h3=x3;
80
81 hn1=h1/h1max*v;
82 hn2=h2/h1max*v;
83 hn3=h3/h1max*v;
84
85 //-----GRAF-----
86 if(i==undefined)i=1;
87 setProperty(bod1,_y,-hn1+137.3);
88 duplicateMovieClip("bod1","bod1"+i,120*i+1);
89 setProperty("bod1"+i,_y,bod1._y);
90 setProperty("bod1"+i,_x,bod1._x+t*3);
91 h1x=getProperty("bod1"+i,_x);
92 h1y=getProperty("bod1"+i,_y)+1;
93 setProperty(l1,_x,h1x);
94 setProperty(l1,_y,h1y);
95

```

```

96 if(i==undefined) i=1;
97 setProperty(bod2, _y, -hn2+74);
98 duplicateMovieClip("bod2", "bod2"+i, i+1);
99 setProperty("bod2"+i, _y, bod2._y);
100 setProperty("bod2"+i, _x, bod2._x+t*3);
101 h2x=getProperty("bod2"+i, _x);
102 h2y=getProperty("bod2"+i, _y)+1;
103 setProperty(l2, _x, h2x);
104 setProperty(l2, _y, h2y);
105
106 if(i==undefined) i=1;
107 setProperty(bod3, _y, -hn3+137.3);
108 duplicateMovieClip("bod3", "bod3"+i, 110*i);
109 setProperty("bod3"+i, _y, bod3._y);
110 setProperty("bod3"+i, _x, bod3._x+t*3);
111 h3x=getProperty("bod3"+i, _x);
112 h3y=getProperty("bod3"+i, _y)+1;
113 setProperty(l3, _x, h3x);
114 setProperty(l3, _y, h3y);
115
116 i++;
117
118 this.hlad1._yscale=1000*h1/h1max;
119 this.hlad2._yscale=1000*h2/h1max;
120 this.hlad3._yscale=1000*h3/h1max;

```

Obr. 35: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie

hladín 3 zásobníkov kvapalín

```

1  if (t==0)
2  {
3      x1=Number (tetad);
4  }
5
6  if (t>=tkon-0.15)
7  {
8      stop();
9  }
10 t=t+0.1;
11
12 //rungekutta 1
13 x1_old = x1;
14 f1 = (1/T)*teta0 + (Z/T)*om - (1/T)*x1;
15 K1 = 1*f1;
16
17 x1 = x1_old + 0.5*K1;
18 f1 = (1/T)*teta0 + (Z/T)*om - (1/T)*x1;
19 K2 = 1*f1;
20
21 x1 = x1_old + 0.5*K2;
22 f1 = (1/T)*teta0 + (Z/T)*om - (1/T)*x1;
23 K3 = 1*f1;
24
25 x1 = x1_old + K3;
26 f1 = (1/T)*teta0 + (Z/T)*om - (1/T)*x1;
27 K4 = 1*f1;
28 x1 = x1_old + (K1+2*(K2+K3)+K4)/6;
29
30 tetak=x1;
31
32 tetan=tetak/tetamax*v;
33
34 if (i==undefined) i=1;
35 setProperty (bod1,_y,-tetan+138.8);
36 duplicateMovieClip ("bod1","bod1"+i,15*i+1);
37 setProperty ("bod1"+i,_y,bod1._y);
38 setProperty ("bod1"+i,_x,bod1._x+t*2.95);
39 h1x=getProperty ("bod1"+i,_x);
40 h1y=getProperty ("bod1"+i,_y)+10;
41 setProperty (l1,_x,h1x);
42 setProperty (l1,_y,h1y);
43 i++;
44
45 function zmenaFarbyObjektu (hladina,tetak,tetas,tetad,farba,filter)
46 {
47     farba = new Color(hladina);
48

```

```

49  if(tetad>tetas)
50  {
51    st=Number(tetas);
52    ht=Number(tetad);
53  }
54  else
55  {  st=Number(tetad);
56    ht=Number(tetas);
57  }
58  if(filter==1)
59  {
60    r=255;
61    g=(1-(tetak-st)/(ht-st))*255;
62    b=0;
63  }
64  if(filter==2)
65  {
66    r=(tetak-st)/(ht-st)*255;
67    g=0;
68    b=(1-(tetak-st)/(ht-st))*255;
69  }
70  farba.setRGB(rgb2hex(r, g, b));
71  function rgb2hex(r, g, b):Number
72  {
73    return(r<<16 | g<<8 | b);
74  }
75  }
76  zmenaFarbyObjektu (hladina,tetak,tetas,tetad,m,2);

```

**Obr. 36: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie
teploty prietokového ohrievača**

```

1  if(t==0)
2  {
3      x1=Number(teta10);
4  }
5
6  if (t>=tkon-0.15)
7  {
8      stop();
9  }
10
11 t=t+0.1;
12 a=Number(q01);
13 b=Number(q02);
14 c=(a+b);
15
16
17 //rungekutta 1
18 x1_old = x1;
19 f1 = (q01*teta01s)/V + (q02*teta02s)/V - (c/V)*x1;
20 K1 = 1*f1;
21
22 x1=x1_old + 0.5*K1;
23 f1 = (q01*teta01s)/V + (q02*teta02s)/V - (c/V)*x1;
24 K2 = 1*f1;
25
26 x1 = x1_old + 0.5*K2;
27 f1 = (q01*teta01s)/V + (q02*teta02s)/V - (c/V)*x1;
28 K3 = 1*f1;
29
30 x1 = x1_old + K3;
31 f1 = (q01*teta01s)/V + (q02*teta02s)/V - (c/V)*x1;
32
33 K4 = 1*f1;
34 x1 = x1_old + (K1+2*(K2+K3)+K4)/6;
35
36 teta1=x1;
37
38 tetan=teta1/teta1smax*v;
39 if(i==undefined)i=1;
40 setProperty(bod1,_y,-tetan+133.5);
41 duplicateMovieClip("bod1","bod1"+i,10*i+1);
42 setProperty("bod1"+i,_y,bod1._y);
43 setProperty("bod1"+i,_x,bod1._x+t*2.9);
44 h1x=getProperty("bod1"+i,_x);
45 h1y=getProperty("bod1"+i,_y)+10;
46 setProperty(l1,_x,h1x);
47 setProperty(l1,_y,h1y);
48 i++;

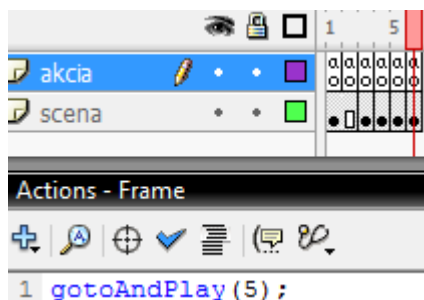
```

```

48 i++;
49
50 function zmena (hladina,tetal,tetals,teta01,farba,filter)
51 {
52     farba = new Color(hladina);
53
54     if(tetal>tetals)
55     {
56         st=Number(tetals);
57         ht=Number(tetal);
58     }
59     else
60     {
61         st=Number(tetal);
62         ht=Number(tetals);
63     }
64     if(filter==1)
65     {
66         r=255;
67         g=(1-(tetal-st)/(ht-st))*255;
68         b=0;
69     }
70     if(filter==2)
71     {
72         r=(tetal-st)/(ht-st)*255;
73         g=0;
74         b=(1-(tetal-st)/(ht-st))*255;
75     }
76
77     farba.setRGB(rgb2hex(r, g, b));
78     function rgb2hex(r, g, b):Number
79     {
80         return(r<<16 | g<<8 | b);
81     }
82 }
83
84 zmena (hladina,tetal,tetals,teta01,farba,2);

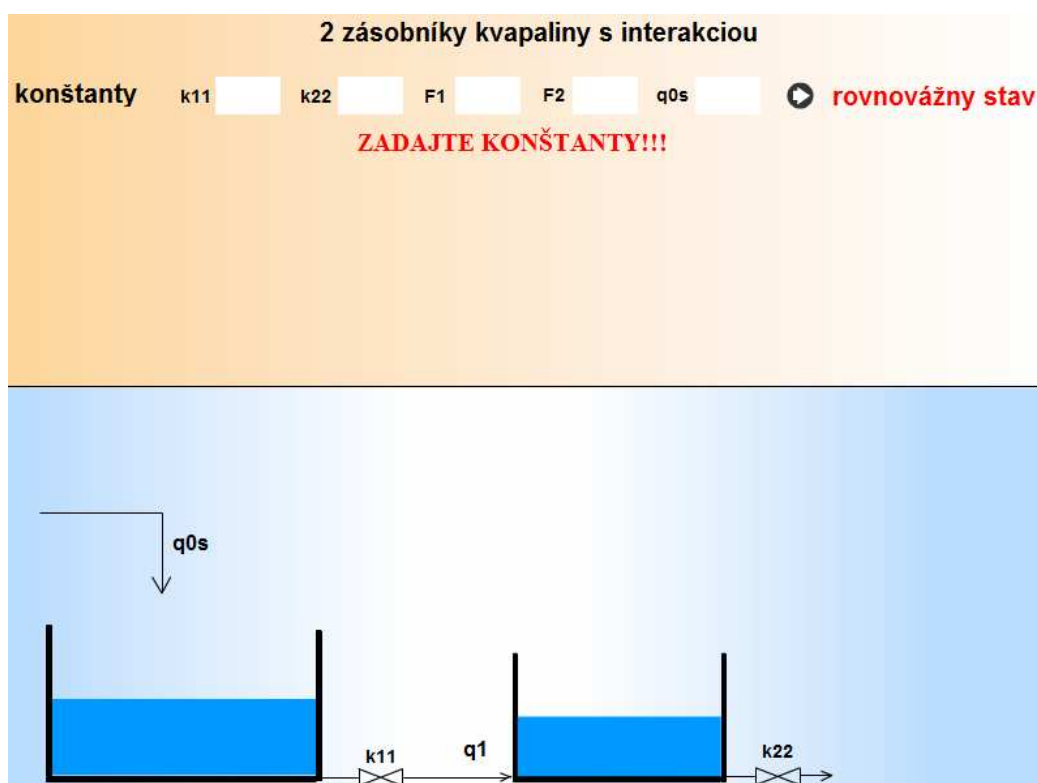
```

Obr. 37: Runge-Kutta, grafické zobrazenie výstupnej veličiny a vykresľovanie teploty miešača s dvomi rôznymi teplotami



Obr. 38: Slučka na snímku 5

Po tom ako spustíme animáciu, sa časový ukazovateľ bude posúvať po časovej osi od prvej snímky a postupne sa budú načítavať údaje až po poslednú snímku. Hodnoty, ktoré zadávame, si program postupne zapamätá a vypočíta ďalšie potrebné hodnoty. Ak by sa náhodou stalo, že používateľ zabudne zadať tieto vstupné hodnoty, tak sa nám na obrazovke zobrazí oznam „ZADAJTE KONŠTANTY!!!“ a pokiaľ ich nezadá, animácia ďalej prebiehať nebude (Obr. 39)



Obr. 39: Oznam, ktorý sa zobrazí pri nezadaní potrebných hodnôt

Všetky animácie a výpočty, ktoré sa v animácii nachádzajú, sú vytvorené pomocou jazyka ActionScript.

Vysvetlenie použitých príkazov ActionScript v animáciách:

1. **stop()**

Funkcia, ktorá zastaví prehrávanie aktuálnej animácie. Najčastejšie sa táto akcia používa pri ovládaní prehrávania klipov pomocou tlačidiel.

2. **this.hlad._yscale**

this – označuje určitý objekt, v tomto prípade sa odkazuje na objekt hladina s názvom hlad

(.) – zápis, ktorý sa používa pre priradenie určitej vlastnosti.

yscale – príkaz pre zmenu hodnoty y

3. **fscommand("allowscale", "true")**

Tento príkaz nám umožní zmeniť veľkosť animácie spolu so zmenou veľkosti okna. Animácia rastie alebo sa zmenšuje spolu s oknom. Pri zmene hodnoty druhého argumentu na *false* zostane veľkosť animácie rovnaká a to aj pri akejkolvek zmene veľkosti okna.

4. **_quality = "HIGH"**

Kvalita dokumentu.

Príkaz `_quality` môže nadobúdať 4 hodnoty: LOW – nízka kvalita, MEDIUM – stredná kvalita, HIGH – vysoká kvalita (používa sa pri prehrávaní súboru swf), BEST – najlepšie dosiahnuteľná kvalita.

5. **on (release)**

Priradenie akcie k tlačidlu. Príklad: po stlačení tlačidla myšou sa presunieme na snímku 3 (Obr. 40).

```

1 on (release)
2 {
3     gotoAndStop(3);
4 }

```

Obr. 40: Ukážka príkladu „on (release)“ a „gotoAndStop(3)“

6. gotoAndStop(3)

Každá snímka časovej osi má svoje číslo, určené poradím snímku na časovej osi. Tento príkaz nám umožní posunúť sa na snímku 3 a tam animáciu zastaviť.

7. duplicateMovieClip("bod1","bod1"+i,120*i+1)

Funkcia, ktorá vytvorí novú inštanciu movie klipu ako jeho kópiu počas prehrávania originálneho movie klipu.

Prehrávanie v duplikovaných movie klipoch vždy začína na prvom snímku, bez ohľadu aktuálnej polohy prehrávaného movie klipu.

Ak je originálny movie klip vymazaný automaticky, vymaže sa aj jeho duplikácia.

Príkaz má 3 parametre:

1. Prvý parameter: meno klipu.
2. Druhý parameter: meno novo vytvoreného klipu.
3. Tretí parameter: vrstva, do ktorej sa má klip vložiť.

8. a=Number(q01)

Transformácia pola na dátový typ číslo.

9. Math.sqrt ()

Vypočíta odmocninu čísla.

Príklad: $f = (q0/F1) - (k11/F1) * \text{Math.sqrt}(x1);$

10. Math.pow (x,y)

Vracia x umocnené na y

Príklad: $hs = \text{Math.pow}((q0s/k11),2);$

Trieda Math je trieda najvyššej úrovne. Obsahuje množstvo príkazov, napr.: Math.abs(), Math.acos(), Math.log(), Math.random(), Math.max(), Math.Min()...

Metódy a vlastnosti tejto triedy sa využívajú pre manipuláciu s matematickými konštantami a funkciami.

11. setProperty(„meno klipu“,vlastnosť, hodnota);

Príkaz, ktorý nám pomôže priradiť nejakú akciu klipu v scéne.

12. getProperty(„meno klipu“,vlastnosť, hodnota);

Funkcia, ktorá vracia hodnotu určených vlastností pre vybranú inštanciu movie klipu.

Záver

Dôležitou súčasťou takmer každej webovej stránky je nejaká animácia. Animácia môže byť v rôznej podobe, či už vo forme prezentácie, videa, hry, či reklamného pútača. Ako iste všetci vieme, webové stránky patria k nášmu každodennému životu. Flash sa stal veľkou rýchlosťou populárnym u tvorcov, ale tak isto aj u návštevníkov a užívateľov internetových stránok. Tvorcovia Internetu sa snažia o to, aby vytvorili čo najlepšie a najzaujímavejšie prostredie pre užívateľov. Flash ponúka neobmedzené možnosti. Dodá obsahu stránky zaujímavejšiu podobu.

Jednou možnosťou ako Flash využiť je tvorba on-line výučby. On-line výučba sa vo svete stala veľmi obľúbenou formou vzdelávania. Nie je sa čomu čudovať, veď žijeme vo svete inovácií, informačno-komunikačných technológií. Internet vládne po celom svete, tak je nevyhnutné sa tomuto faktu prispôbovať, a to rôznymi spôsobmi. Jedným takýmto spôsobom je virtuálne laboratórium. Virtuálne laboratórium sa stáva vo svete novinkou vo forme vzdelávania. Školy a vzdelávacie inštitúcie sa snažia modernizovať a zlepšovať výučbu aj vďaka rôznym takýmto virtuálnym laboratóriám. Odvetví, v ktorom sú takéto laboratóriá vyhotovené je množstvo. Či už je to oblasť lekárstva, chémie, strojárstva, elektrotechniky, informatiky alebo matematiky.

Cieľom mojej práce bolo vytvorenie takéhoto virtuálneho laboratória, ktoré poslúži na získanie nových vedomostí a zručností o neznámych procesov. Virtuálne laboratórium pozostáva z piatich modelov. Tri modely sa týkali zásobníkov kvapalín a ďalšie dva miešača a prietokového ohrievača. Užívatelia si otvoria animáciu, zadajú do nej niekoľko vstupných dát a behom pár sekúnd sa animácia spustí. Uvidia ako sa mení výstupná veličina, jej hodnota a aj grafické zobrazenie výstupnej veličiny.

Zoznam použitej literatúry

- [1] J. Mikleš, M. Fikar,: Modelovanie, identifikácia a riadenie procesov I., Modely a dynamické charakteristiky spojitých procesov, STU Bratislava 1999
- [2] Matematické modely. Dostupné na:
<http://sk.wikipedia.org/wiki/Matematick%C3%BD_model>, online [01. 05. 2010]
- [3] Macromedia Flash. Dostupné na: <<http://www.flashland.sk/flash/flash.php>>, online [30. 04. 2010].
- [4] J. Fotr, Z. Schneider,: Flash 5, Pro grafiky a tvůrce webů [Pre grafikov a tvorcov webov], Computer Press Praha 2000
- [5] Macromedia Flash. Dostupné na:
<http://en.wikipedia.org/wiki/Adobe_Flash>, online [01. 04. 2010].
- [6] Macromedia Flash. Dostupné na:
www.gjar-po.sk/~dzurenko3c/files/inf/prirucka.k.macromedia.flash.mx.pdf,
online [01. 05. 2010].
- [7] História Flash. Dostupné na:
<http://www.tahaky-referaty.sk/Tvorba-vzdelavacieho-obsahu-v-prostredi-Macromedia-Flash/2991/>, online [12. 05. 2010]
- [8] ActionScript. Dostupné na:
<http://lst.science.upjs.sk/2006/postupy/grafika/gunisovci/flash/actionscrip.htm>
online [01. 05. 2010].
- [9] M. Schaeffer,: Adobe Flash CS4 Professional, 100 nejlepších postupů [100 najlepších postupov], Computer Press a.s. Brno 2009

Prílohy

Príloha A:

CD s elektronickou verziou diplomovej práce a so súbormi:

zasobnik.swf

zasobnik.flu

dva_zasobniky.swf

dva_zasobniky.flu

tri_zasobniky.swf

tri_zasobniky.flu

miesac.swf

miesac.flu

ohrievac.swf

ohrievac.flu