



Univerzita Karlova
První lékařská fakulta

Autoreferát disertační práce

Modelica in physiological modelling

*Models with spatially distributed parameters,
Authoring educational simulators*

Jan Šilar

Praha 2019

Doktorské studijní programy v biomedicíně

*Univerzita Karlova v Praze
a Akademie věd České republiky*

Obor: Biomedicínská informatika

Předseda oborové rady: Prof. MUDr. Štěpán Svačina, DrSc., MBA

Školící pracoviště: Ústav patologické fyziologie

Autor: Ing. Jan Šilar

Školitel: doc. MUDr. Jiří Kofránek, CSc.

Oponenti:

prof. RNDr. MUDr. Petr Maršálek, Ph.D.

prof. Ing. Karel Roubík, Ph.D.

Autoreferát byl rozeslán dne:

Obhajoba se koná

Dne: 2.9.2019 v 11:15

V knihovně 3. interní kliniky, U nemocnice 1, Praha 2

S disertační prací je možné se seznámit na děkanátě První lékařské fakulty Univerzity Karlovy v Praze.

Obsah

| | |
|--|-----------|
| Souhrn | 6 |
| Summary | 7 |
| Obsah práce | 8 |
| Úvod | 8 |
| Modely ve fyziologii | 8 |
| Jazyky a nástroje pro modelování | 8 |
| Cíle práce | 12 |
| Parciální diferenciální rovnice v Modelica | 13 |
| Současné možnosti propojení PDR a Modelica | 13 |
| Jazykové rozšíření PDEModelica1 | 14 |
| Implementace v OM | 16 |
| Aplikace PDEModelica1 | 17 |
| Protiporudová teplená výměna | 17 |
| Dýchání do sněhu | 18 |
| Tvorba výukových simulátorů | 21 |
| Bodylight.js | 23 |
| Použité technologie | 23 |
| Nástroje a pracovní postup | 23 |
| Nové simulátory | 25 |
| Závěr | 29 |
| Reference | 29 |
| Seznam publikací | 33 |
| S IF, součástí práce | 33 |
| Bez IF, součástí práce | 33 |
| S IF, další | 34 |
| Bez IF, další | 34 |

Souhrn

Tato práce pojednává o využití jazyka Modelica pro fyziologické modelování. Jsou zde popsány nově vyvinuté metody pro implementaci modelů a tvorbu simulátorů.

Práce se zabývá konkrétně dvěma základními tématy: 1) modelováním systémů popsaných pomocí parciálních diferenciálních rovnic v Modelica (což není standardem jazyka přímo podporováno) a 2) tvorbou webových e-learningových simulátorů založených na modelech implementovaných v Modelica.

Bylo navrženo jazykové rozšíření *PDEModelica1* pro 1-dimenzionální parciální diferenciální rovnice. Modelovací nástroj OpenModelica byl rozšířen o podporu PDEModelica1. Bylo vytvořeno několik modelů, které toto rozšíření využívají. Jedním z nich je model advekce a difuze CO_2 a O_2 ve sněhu, který doplňuje experiment týkající se dýchání osoby zasypané lavinou. Prokázala se použitelnost nového jazykového rozšíření v praxi.

Byla vytvořena sada nástrojů Bodylight.js pro tvorbu webových klientských simulátorů využívající JavaScript technologie. Model je implementován v Modelica, animace jsou vytvářeny v nástroji Adobe Animate. Nově vytvořený nástroj Bodylight.js Composer slouží k sestavení simulátoru z interaktivních animací, grafů, dalších vstupních a výstupních komponent a modelu běžícího na pozadí. Série simulátorů vysvětlujících regulaci sodíku a vody v nefronu je zatím nejkompexnější aplikací vytvořenou pomocí Bodylight.js. Nová technologie se ukázala být velice užitečná. Tvorba podobných simulačních aplikací by byla bez tohoto nástroje extrémně náročná.

Summary

This thesis deals with modelling in physiology using the *Modelica* language. New methods for model implementation and simulator production were developed.

The thesis focuses on two main topics: 1) modelling of systems described by partial differential equations in Modelica (which is currently not supported by the language standard) 2) production of web-based e-learning simulators driven by models implemented in Modelica.

A Modelica language extension called PDEModelica1 for 1-dimensional partial differential equations was designed. The OpenModelica modelling tool was extended to support PDEModelica1. Several models using PDEModelica1 were implemented. One of them is the model of advection and diffusion of CO₂ and O₂ in snow, which aimed to complement experiment on breathing of an individual buried in an avalanche. PDEModelica1 was proved useful for the purpose.

A framework termed Bodylight.js for production of web-based client-side e-learning simulators based on JavaScript technologies was implemented. Models are implemented in Modelica. Interactive animations are created using Adobe Animate. A new tool called Bodylight.js Composer allows to produce the final simulator comprising the interactive animations, plots, various input and output widgets and the model running in background. A simulator series on sodium and water maintenance in the nephron are the most complex application implemented using Bodylight.js so far. The novel technology was found convenient for its purpose as implementation of such simulators would be extremely laborious without it.

Obsah práce

Úvod

Modely ve fyziologii

Matematické modely jsou ve fyziologii využívány pro vědecké účely: umožňují formulovat a ověřovat (srovnáním výsledku simulace s experimentálními daty) hypotézy. Začínají se také objevovat v klinické praxi: je možné je využívat pro predikci budoucího vývoje i pro odhad neměřitelných veličin (inverzní úlohy). Modely se využívají také při výuce.

Výukové modely nemusí být vždy úplně detailní. Často stačí studenty seznámit s obecnými trendy v systému a nejde o přesné hodnoty. V tom případě stačí jen jednodušší modely popisující pouze studovaný systém. Další související systémy mohou být často zanedbány (princip *Ceteris paribus*) [1].

Pokud je potřeba, aby simulace dávala přesné výsledky, je nutné vycházet z rozsáhlých integrativních modelů zahrnujících všechny související systémy. Některé fyziologické fenomény jsou ze své podstaty komplexní a není možné je popsat jednoduchým modelem. Příkladem je modelování hypertenze. Zabývali jsme se mimo jiné testováním komplexních integrativních modelů pro výpočet hypertenze indukované zvýšeným příjmem soli [2].

V této práci se zabýváme především modely popsanými pomocí algebraických, obyčejných diferenciálních (ODR) a partiálních diferenciálních (PDR) rovnic.

Jazyky a nástroje pro modelování

Pro implementaci modelů je možné využívat obecné programovací jazyky jako je Fortran, C/C++, Python, C# a mnoho dalších. Tento přístup

je ale pracný. Současně s modelem je třeba implementovat i řešič a další podpůrné funkce. Takto zapsaný model je navíc dost nepřehledný – rovnice modelu často mizí mezi dalším kódem.

Lepší je využít některý doménově specifický jazyk pro modelování. Některé jazyky po modelování využívají **blokově orientovaný** přístup (např. Simulink, LabView). Modely jsou zde skládány z funkčních bloků (součet, násobek, integrace, ...), které mají definované vstupy, výstupy a funkci mapující vstupní hodnotu na výstupní. Vstupní hodnoty modelu procházejí sítí vzájemně propojených bloků a tak jsou počítány výstupy. Kauzalita modelu (tzn. které proměnné jsou vstupy a které výstupy) musí být dána předem. Vnitřní kauzalita výpočtu modelu musí být odvozena vývojářem při implementaci modelu. To je pracné a navíc to znemožňuje použít model v jiném kontextu s odlišnou kauzalitou.

V **rovnícovém** (akauzálním) přístupu jsou modely popsány rovnicemi, které definují vztahy mezi proměnnými. V modelu není určeno, která proměnná se má počítat z které rovnice. Kauzalitu výpočtu (t.j. označení výstupních proměnných pro jednotlivé rovnice, a určení pořadí, ve kterém se budou rovnice vyhodnocovat) odvodí automaticky simulační nástroj až při simulaci. Modely jsou často sestavovány ze submodelů z různých knihoven. Díky nezávislosti na kauzalitě může být každý submodel v knihovně jen jedenkrát, přestože bude používán v kontextu s různou kauzalitou.

Modelica¹ je rovnícový, objektově orientovaný jazyk pro modelování komplexních fyzikálních systémů. Umožňuje zapsat modely pomocí algebro-diferenciálních a diskrétních rovnic. Parciální diferenciální rovnice (PDR) zatím nepodporuje. Nejvýznamnější nástroje pro modelování v Modelice jsou komerční Dymola a volně šiřitelná OpenModelica. Existuje mnoho knihoven (komerčních i otevřených) pro modelování v různých zejména fyzikálních a průmyslových doménách. Základní knihovnou, ze které mnoho dalších knihoven vychází, je *Modelica Standard Library* (MSL). Podporuje modelování elektrických, magnetických, mechanických, fluidních, termálních a obecně matematických systémů. *PhysioLibrary* je knihovna pro modelování

¹ modelica.org

fyziologických systémů implementovaná Markem Matejákem [3] při reimplementaci modelu *HumMod* od Roberta Hestera [4].

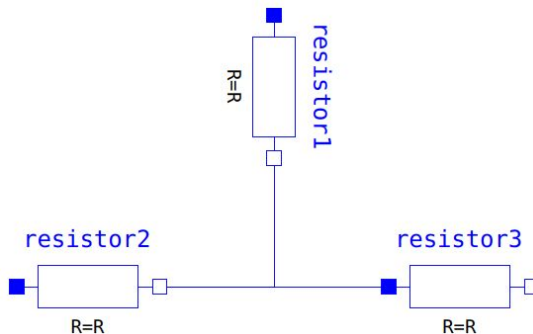
V Modelica jsou elementární modely implementovány textově pomocí rovnic. Komplexnější modely je možné skládat z komponent (submodelů). Komponenty mají konektory, jejichž pomocí mohou být vzájemně propojeny. Konektory obsahují dva druhy proměnných – tokové a potenciální. Propojení dvou nebo více komponent v jednom bodě (*connection set*) je při překladu modelu nahrazeno rovnicemi mezi proměnnými propojených konektorů podle zobecněného Kirchhoffova zákona:

- Všechny potenciální proměnné jsou si rovny
- Součet všech tokových proměnných je roven nule

Platí konvence, že tok je kladný ve směru *dovnitř* komponenty. Například spojení tří rezistorů (obrázek 1) je při překladu nahrazeno rovnicemi

$$\begin{aligned} \text{resistor1.n.v} &= \text{resistor2.n.v}; \\ \text{resistor1.n.v} &= \text{resistor3.p.v}; \\ \text{resistor2.n.i} + \text{resistor3.p.i} \\ &+ \text{resistor1.n.i} = 0.0; \end{aligned}$$

kde n značí negativní (bílý) konektor, p kladný modrý konektor.



Obrázek 1: Napětí na propojených konektorech jsou si rovna a součet proudů propojenými konektory je roven nule podle Kirchhoffových zákonů.

Zobecněné Kirchhoffovy zákony platí i v dalších fyzikálních doménách. V tabulce 1 jsou vypsány potenciální a tokové proměnné z různých domén MSL.

| Doména | Tok | Potenciál |
|--------------------|-----------------------------|------------------------------|
| Elektrická | Proud – i | Napětí – v |
| Magnetická | Magnetický tok – Φ | Magnetický potenciál – V_m |
| Mechanická (posun) | Síla – f | Poloha – s |
| Mechanická rotační | Sílový moment – τ | Úhel – ϕ |
| Fludní | Hmotnostní tok – m_{flow} | Tlak – p |
| Termální | Tepelný tok – Q_{flow} | Teplota – T |

Tabulka 1: *potenciální a tokové proměnné v různých doménách MSL.*

Modelica podporuje dědičnost. V případě komponent, které se shodují v některých rovnicích, umožňuje dědičnost tyto rovnice zapsat jen jednou ve společné komponentě, kterou ostatní komponenty dědí.

Překladače jazyka Modelica mívají dvě části. Tak zvaný *front-end* a *back-end*. Front-end konvertuje hierarchickou objektovou strukturu modelu do jednoho systému rovnic (tzv. *flat model*). Jsou odstraněny všechny třídy a komplexní typy a vyřešeny dědičnosti.

Back-end transformuje model tak, aby mohl být numericky řešen. Probíhá zde odvození kauzality: je stanoveno pořadí, v jakém budou rovnice vyhodnocovány a pro jednotlivé rovnice jsou určeny proměnné, jejichž hodnota se vyhodnocením rovnice má spočítat. Dále jsou určeny subsystemy rovnic, které není možné spočítat postupným vyhodnocováním a musí být spočítány dohromady jako soustava (tzv. *strong component*).

Nakonec překladač vygeneruje funkce reprezentující model v některém standardním programovacím jazyce (většinou C). Tyto funkce se přeloží standardním překladačem (třeba gcc) společně s knihovnou běhového prostředí (*runtime*), která implementuje numerické metody pro řešení modelu. Tím vznikne spustitelná aplikace, která počítá řešení modelu.

*Functional Mock-up Interface*² (FMI) je standardizované rozhraní pro sdílení modelů. Toto rozhraní je podporováno mnoha simulačními nástroji. *Functional Mock-up Unit* (FMU) je realizace FMI pro jeden konkrétní model. FMI je zip archiv, který obsahuje mimo jiné binární soubor s přeloženým modelem a rozhraním podle standardu FMI. Verze FMU pro kosimulaci obsahuje navíc solver pro řešení modelu. FMI může také obsahovat zdrojové soubory modelu i solveru.

Cíle práce

Modelica ale nepodporuje parciální diferenciální rovnice, které jsou potřeba pro popis některých fyziologických systémů. Prvním cílem této práce je navrhnout **rozšíření jazyka Modelica pro parciální diferenciální rovnice**, implementovat toto rozšíření do simulačního nástroje OpenModelica a využít ho k simulaci skutečných fyziologických problémů.

Výstupem z většiny simulačních nástrojů jsou modely přeložené jako nativní binární aplikace, které nemohou být spuštěny v prohlížeči. Webové aplikace jsou platformě nezávislé a neinstalují se. Aby byly výukové simulátory maximálně přístupné studentům, je vhodné je vytvářet formou webových aplikací. Druhým cílem této práce bylo vytvoření sady **nástrojů pro tvorbu výukových webových simulátorů**, běžících kompletně **na klientském zařízení** a založených na **modelech v Modelica**.

² fmi-standard.org

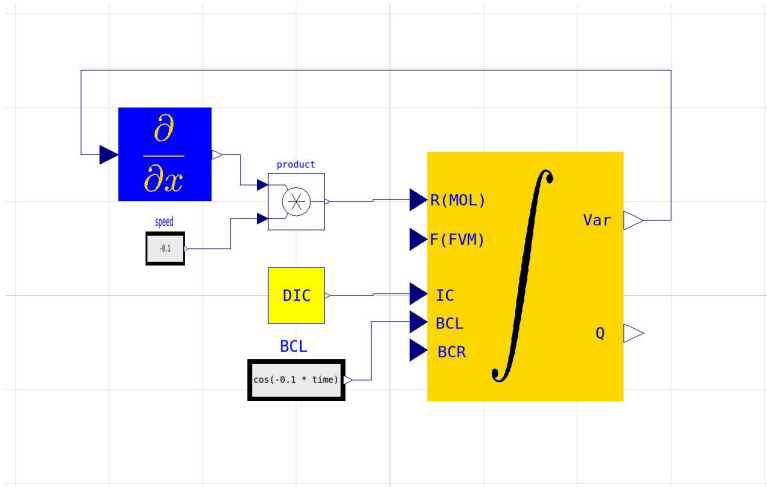
Parciální diferenciální rovnice v Modelica

Současné možnosti propojení PDR a Modelica

V současné době se nabízí několik možností, jak propojit modely obsahující parciální diferenciální rovnice s modely v jazyce Modelica.

Model s PDR může být implementován v nějakém **specializovaném nástroji pro PDR** jako je například HiFlow3³ a propojen s modelem v Modelica pomocí **FMI** [5].

PDELlib [6] je **knihovna** v Modelica, která umožňuje poskládat parciální diferenciální rovnici z komponent jako je např. prostorová derivace a časový integrál (obrázek 2). Pro řešení PDR využívá PDELlib metodu přímk (MOL) [7] nebo metodu konečných objemů [8]. Tato knihovna je bohužel neudržovaná a v současné době nefunkční.



Obrázek 2: Advekční rovnice implementovaná v PDELlib

Další variantou je transformovat PDR na systém ODE pomocí metody přímk (MOL). Prostorově závislá proměnná je nahrazena

³ hiflow3.org

posloupností, prostorové derivace jsou nahrazeny diferencemi. Tím je převedena PDR na soustavu ODR, kterou je možné Modelica standardně zapsat a řešit.

Operátor `spatialDistribution()` je součástí standardu Modelica pro řešení jednorozměrné advekční rovnice. Další typy rovnic tímto způsobem podporovány nejsou.

Přímá podpora pro PDR v Modelica pomocí jazykového rozšíření je ideální, ale doposud nedostupné řešení. Levon Saldamli navrhl v rámci své disertační práce [9] jazykové rozšíření *PDEModelica* zavádějící jazykové konstrukty pro PDR a částečně implementoval podporu pro toto rozšíření v OpenModelica. Implementace nebyla nikdy zcela dokončena a kvůli dalšímu vývoji v OpenModelica byla z kódu nakonec vypuštěna.

Jazykové rozšíření PDEModelica1

Navrhli jsme jazykové rozšíření *PDEModelica1* pro PDR v 1D, které vychází ze Saldamliho rozšíření, ale přináší některé změny a vylepšení. Nové jazykové prvky rozšíření ukážeme na jednoduchém příkladu advekční rovnice

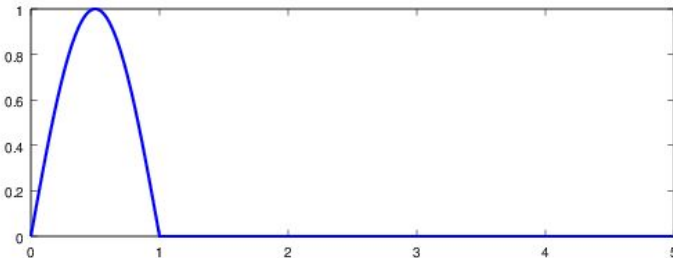
$$\frac{\partial u}{\partial t} + 2 \frac{\partial u}{\partial x} = 0 \quad t \in [0, T], x \in [0, 5]$$

s okrajovou podmínkou

$$u(t, 0) = 0$$

a počáteční podmínkou

$$u(0, x) = \text{if } x < 1 \text{ then } \sin(\pi x) \text{ else } 0.$$



Obrázek 3: Počáteční podmínka příkladu s advekční rovnicí.

Počáteční podmínka je vykreslena na obrázku 3. Tento model je možné zapsat pomocí PDEModelica1 jako:

```

model advection "advection"
  parameter DomainLineSegment1D omega(L = 5, N=200);
  field Real u(domain = omega);
initial equation
  u = if omega.x < 1 then sin(3.14*omega.x) else 0 indomain omega;
equation
  der(u) + 2*pder(u,x) = 0 indomain omega;
  u = 0 indomain omega.left;
  u = extrapolateField() indomain omega.right;
end advection;

```

- DomainLineSegment1D – nová třída (record) pro definiční doménu. Je definovaná jako

```

record DomainLineSegment1D "1-dimensional domain where a partial differ
  record Region
  end Region;
  parameter Real x0(unit="m")=0 "x value at left boundary";
  parameter Real L(unit="m")=1 "length of the domain";
  constant Integer N(unit="")=10 "number of grid nodes";
  parameter Real dx = L / (N-1) "grid space step";
  parameter Real[N] x(each unit="m") = array(x0 + i*dx for i in 0:N-1)
  Region left, right, interior "regions representing boundaries and the
end DomainLineSegment1D;

```

- field – nové klíčové slovo pro deklaraci pole (proměnné v prostoru). Povinným atributem v definici pole je domain, kterým deklarujeme definiční doménu.
- indomain – je klíčové slovo pro deklaraci oboru platnosti rovnice, používá se v
 - PDR a počátečních podmínkách (indomain.omega)
 - okrajových podmínkách (indomain.omega.left/right)
- Konstruktor literálu pole – v čase konstantní pole můžeme definovat pomocí prostorové souřadnice omega.x. Za výrazem musí následovat indomain.omega (viz počáteční podmínky)
- pder() – parciální derivaci zapisujeme pomocí tohoto operátoru .
 - 1. derivace pder(u,x)
 - 2. derivace pder(u,x,x)
- extrapolateField() – Pokud pro krajní hodnotu pole derivovaného podle x není zadána okrajové podmínka, je tuto hodnotu potřeba dopočítat extrapolací. Automatická extrapolace zatím není

implementovaná a použití operátoru `extrapolateField()` je dočasné řešení.

Oproti původnímu rozšíření od Saldamliho je změněna syntaxe

- definice domény
- operátoru parciální derivace
- operátoru `indomain`
- konstruktoru literálu pole

Implementace v OM

Rozšířili jsme Front-end překladače OpenModelica Compiler (OMC) o podporu pro PDEModelica1. Řešení modelu je založeno na metodě přímek (MOL). Implementovali jsme do překladače nové jazykové konstrukty. Dále jsme přidali diskretizační modul, který v modelu nahradí pole (proměnné definované pomocí `field`), posloupnostmi (`array`) a prostorové derivace diferencemi. Pro první i druhou prostorovou derivaci se používají centrální diference

$$\frac{\partial u}{\partial x} \rightarrow \frac{u_{i+1} - u_{i-1}}{2\Delta x}$$
$$\frac{\partial^2 u}{\partial x^2} \rightarrow \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}.$$

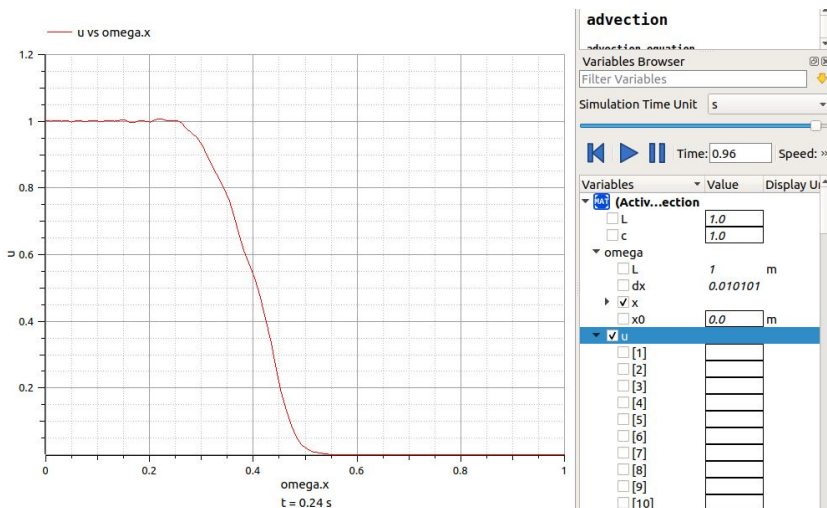
Tím vznikne soustava obyčejných diferenciálních rovnic, a další části překladače a simulačního runtimeu ji umí standardně řešit. OpenModelica nabízí několik numerických metod pro časovou integraci. Je vhodné použít některou z implicitních metod. Výpočet pomocí explicitních metod je pro mnoho PDR nestabilní.

Protože PDEModelica1 není součástí Modelica standardu, je potřeba její podporu v překladači zapnout přepínačem

```
--grammar=PDEModelica
```

(v `tools-options-simulation-OMCFlags`).

Rozšířili jsme vizualizační nástroje v OpenModelica o dvě nové funkce pro vykreslování polí. Funkce `array plot` vykresluje hodnotu prvku pole na ose y proti indexu prvku pole na ose x. Druhá funkce `array parametric plot` vykresluje hodnoty prvků jednoho pole na ose x proti hodnotám prvků druhého pole na ose y (obrázek 4). Čas se v obou případech ovládá posuvníkem, případně je možné jej zadat číselně.



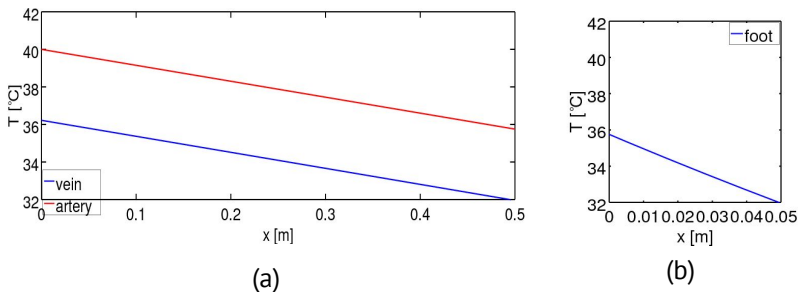
Obrázek 4: Funkce array plot parametric – proměnná u na ose y versus souřadnice $\omega \cdot x$ na ose x . Čas je ovládán posuvníkem.

Aplikace PDEModelica1

Protiproudová tepelná výměna

Čápi a další ptáci vydrží stát na ledu nebo ve studené vodě dlouhou dobu aniž by docházelo k významnému podchlazení. Pomáhá jim k tomu tepelná výměna, která probíhá mezi tepnou a žilou v noze [10]. Arteriální krev se v noze ochlazuje od žilní krve. Tím se sníží tepelný rozdíl mezi krví a ledem v místě kontaktu, a v důsledku toho i celková tepelná ztráta.

Pro otestování PDEModelica1 byl implementován model této protiproudové výměny popsany v [11]. Jedná se o soustavu tří advektivních rovnic (tepna, cévy chodidla, žila) rozšířených o členy přestupu tepla. PDEModelica1 se osvědčila. Řešení modelu v rovnovážném stavu je na obrázku 5.



Obrázek 5: teplota krve (a) v tepně a žíle, (b) v chodidle

Dýchání do sněhu

Podíleli jsme se na projektu, který se zabývá přežitím v lavině. Proudění vzduchu je ve sněhu výrazně omezeno. Zасыpaný nadechuje dříve vydechnutý vzduch. Koncentrace O_2 klesá, koncentrace CO_2 roste a zасыpaný se začne brzy dusit. Složení vzduchu se částečně obnovuje difuzí plynů ve sněhu a rozpouštěním CO_2 ve vodě ve sněhu obsažené.

Případná vzduchová dutina kolem úst zасыpaného výrazně zvyšuje šanci na přežití, což je známý fakt. Čím je objem dutiny větší, tím je delší čas, který oběť pravděpodobně ve sněhu přežije. Roli zde hrají dva faktory: 1) v dutině se míchá vydechovaný vzduch s čerstvým. Čím je dutina větší, tím více se vzduch v dutině občerství. 2) Při dýchání musí vzduch projít sněhovými vrstvami, které obklopují dutinu. S rostoucím povrchem dutiny klesá odpor vůči proudění vzduchu a v důsledku toho klesá dechová práce, spotřeba O_2 a produkce CO_2 .

Roubík a Sieger provedli experiment, při kterém dobrovolníci dýchali (systémem dvou trubek pro eliminaci mrtvého prostoru) do dutiny ve sněhu [12]. Měřili koncentrace O_2 , CO_2 v dutině i v náustku, objemový tok a další veličiny. Jednou z otázek, na kterou hledali odpověď bylo, který z mechanismů účinku dutiny je dominantní.

Implementovali jsme model advekce a difuze O_2 a CO_2 a rozpouštění CO_2 v kapalně složce sněhu (podle Henryho zákona). Cílem modelu bylo doplnit experiment a pomoci zodpovědět otázku ohledně dutiny. Advektivně-difuzní rovnici jsme převedli do sférických souřadnic a díky

symetrii položili všechny derivace vyjma v radiálním směru rovny nule, čímž se snížil počet dimenzí úlohy na 1. Dostali jsme

$$\frac{\partial c}{\partial t} + \left(u - \frac{2D}{r}\right) \frac{\partial c}{\partial r} - D \frac{\partial^2 c}{\partial r^2} = 0$$

$$u = \frac{q}{4\pi r^2}$$

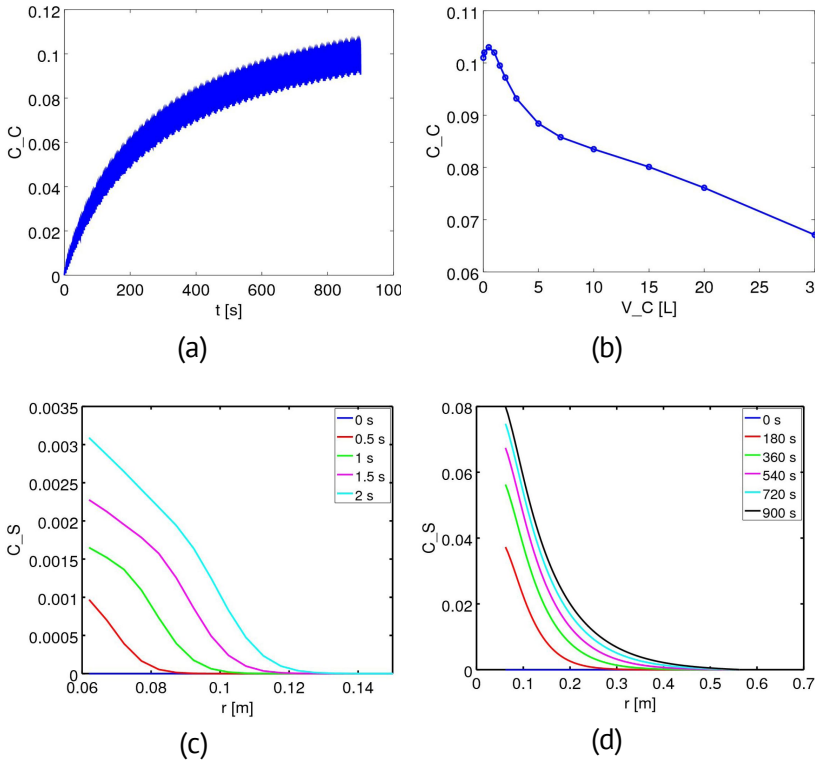
kde c je koncentrace, u rychlost advekce, D difuzní koeficient, r poloměr a q objemový tok vzduchu. Následující výpis kódu ukazuje část modelu s definicí domény ω , definicí pole C_S , okrajovou podmínkou, extrapolací na druhém kraji a advekčně-difuzní rovnicí:

```

DomainLineSegment1D omega (L = 0.5, N =
100, x0 = R_C) "x is actually r, center on
the left";
field Real C_S(domain = omega)
"concentration of CO2 in snow";
(...)
//Left BC during exhalation, extrapolation
during inhalation
C_S = if exhale then C_CS else
extrapolateField(C_S) indomain omega.left;
//The advection-diffusion equation
der(C_S) + (q / (4 * pi * omega.x ^ 2)
- 2 * D_S / omega.x) * pder(C_S, x)
- D_S * pder(C_S, x, x) = 0
indomain omega;
end sb1m;

```

Zatím byl použit jednoduchý model plic s konstantním periodickým dechovým vzorem a konstantní spotřebou O_2 a produkcí CO_2 . Některé parametry modelu byly jen odhadnuty. Model byl popsán v práci [13]. Řešení je na obrázku 6.



Obrázek 6: (a) – koncentrace CO₂ v dutině (volume 1L) během 15 minut. (b) – průměrná koncentrace CO₂ v dutině po 15 minutách v závislosti na objemu dutiny. (c) – koncentrační profil CO₂ ve sněhu v závislosti na poloměru během prvního výdechu. (d) – koncentrační profil CO₂ ve sněhu na konci výdechu v různých časech.

Dalším cílem bylo implementovat přesnější model metabolické spotřeby O₂ a produkce CO₂ beroucí v úvahu dechovou práci. Dále jsme chtěli odhadnout neznámé parametry (difuzní koeficient, poměr pevné a kapalné složky sněhu, fyziologické parametry ...) a verifikovat model pomocí fitování experimentálních dat. Ukázalo se, že experimentální data toků byla při měření poškozena. Nenašli jsme vhodný přístroj, se kterým by bylo možné experiment zopakovat a toky změřit správně.

Proto zatím verifikace modelu neproběhla. Nicméně jsme prokázali, že pomocí PDEModelica1 je možné řešit skutečný vědecký problém.

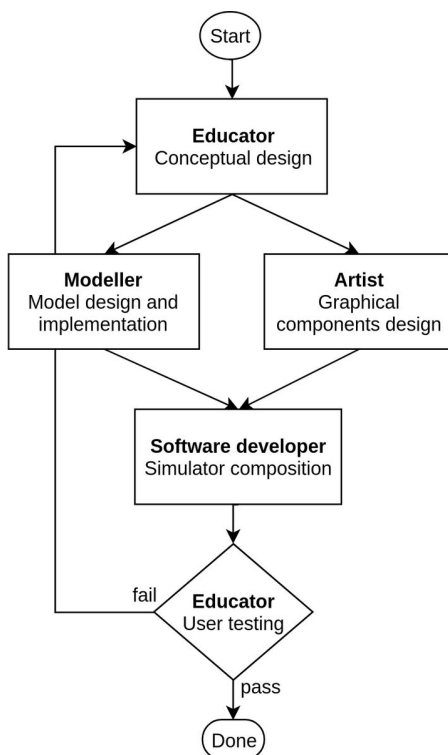
Tvorba výukových simulátorů

Druhá část práce se zabývá tvorbou webových simulátorů pro výuku fyziologie. Simulátor je aplikace kombinující model zkoumaného systému s grafickým uživatelským rozhraním (GUI), které tento systém vizualizuje a umožňuje interakci s modelem. Součástí GUI jsou vstupní prvky (textové a číselné vstupy, přepínače, tlačítka a posuvníky), které umožňují ovládat model a kontrolovat hodnoty jeho parametrů. Dále GUI obsahuje výstupní prvky (grafy a interaktivní animace), které zobrazují řešení modelu.

Simulátory jsou skvělým rozšířením tradičních forem výuky. Umožňují zábavnou formou aktivně zapojit studenty. Pomocí simulátorů mohou studenti testovat svoje předpoklady o chování systému a tak si ověřit a prohloubit správné pochopení přednášené látky.

Tvorba simulátorů je náročný proces, který vyžaduje multioborový tým (obrázek 7):

- Zkušený učitel simulátor konceptuálně navrhne (stanoví výukové cíle, požadavky na model a požadavky na GUI)
- Systémový fyziolog navrhne a implementuje model
- Grafik navrhne GUI po vzhledové stránce a vytvoří pro něj grafické interaktivní prvky
- Programátor pak dá vše dohromady



Obrázek 7: Vývoj simulátoru

Simulátory jsou ideální ve formě webových aplikací, běžících kompletně na klientském zařízení. Webové aplikace jsou platformě nezávislé a neinstalují se. Čistě klientské řešení pak řeší problém škálování a nutnosti neustálého rychlého připojení k internetu. V případě aplikací typu klient-server je nutné zajistit relativně drahé serverové služby (cloud service), které jsou schopné nárazově poskytnout dostatečný výpočetní výkon v případě, že simulátor chce používat zároveň například celá posluchárna studentů. Tenhle problém v případě klientského řešení odpadá.

Bodylight.js

Bodylight.js je nově vytvořená sada nástrojů pro tvorbu webových klientských simulátorů založených na modelech v jazyce Modelica [14]. Webová stránka projektu je na adrese bodylight.physiome.cz. Je zde dokumentace a tutoriály, které ukazují, jak postavit některé simulační aplikace.

Použité technologie

Bodylight.js využívá následující technologie:

- **Modelica, FMI** – popsáno dříve
- **JavaScript** je objektově orientovaný jazyk podporovaný naprostou většinou současných webových prohlížečů. Používá se běžně na internetových stránkách.
- **WebAssembly** je binární instrukční formát úzce spjatý s JavaScriptem. Prohlížeči je také široce podporován. Běh kódu ve WebAssembly se rychlostí blíží nativnímu kódu.
- **Emscripten** umožňuje překlad kódu z mnoha různých jazyků do WebAssembly [15].
- JavaScript knihovny
 - **GrapesJS** a **React** – pro tvorbu uživatelského rozhraní v HTML a JavaScript
 - **EaseUS** pro interaktivní grafiku
 - **Plotly.js** pro grafy
- **Adobe Animate** proprietární nástroj pro vytváření interaktivních grafických komponent. Podporuje export do EaseUS.

Nástroje a pracovní postup

Model je implementován v simulačních nástrojích **Dymola** nebo **OpenModelica** a exportován do FMU pro kosimulaci tak, že FMU obsahuje i zdrojový kód modelu a solveru. Tyto zdrojové kódy FMU jsou přeloženy překladačem Emscripten do WebAssembly. Pro tento krok byl vytvořen tzv. **Docker container**, který obsahuje správně nakonfigurovaný Emscripten a provede překlad do WebAssembly automaticky.

Interaktivní grafické prvky se vytvářejí v **Adobe Animate** a exportují se do JavaScriptu pro knihovnu EaseJS.

Bodylight.js Composer (obrázek 8) je nová webová aplikace pro vytvoření finálního simulátoru. Je dostupná na adrese bodylight.physiome.cz/composer. Umožňuje vložení přeloženého modelu a interaktivních animací, a složení GUI z komponent nabízených v paletách.

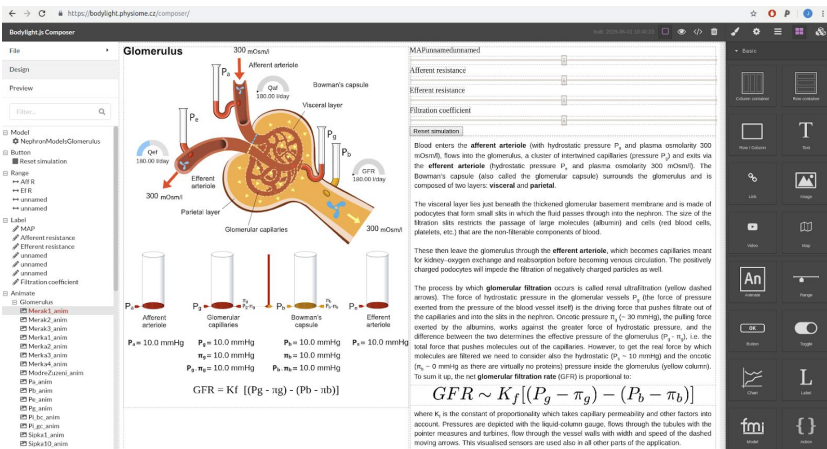
Dostupné komponenty pro ovládání modelu a simulace jsou: posuvník, přepínač, tlačítko, textový vstup. Tyto komponenty mohou být propojeny s parametry modelu, aby mohl uživatel ovládat jejich hodnotu.

Komponenty pro vizualizaci jsou: graf, textový výstup, kontejner pro animaci. Tyto komponenty jsou naopak propojeny s proměnnými modelu a jsou tak modelem řízeny. Než je hodnota proměnné podána do výstupní komponenty, je na ni možné aplikovat jakoukoli programátorem definovanou funkci.

Dále jsou dostupné komponenty pro rozmístění dalších prvků, kontejner modelu a jiné. Vzhled a chování použitých komponent může být nastaveno pomocí konfiguračních panelů.

Projekt simulátoru může být kdykoli vyexportován do souboru (*.bjp), uložen na disk a později znovu otevřen. Hotový simulátor je nakonec vyexportován jako webová stránka v HTML, JavaScript a WebAssembly.

Dále je možné komponentám přiřadit akce (hotové nebo definovat nové, např. pro restartování modelu, smazání grafu atp.).



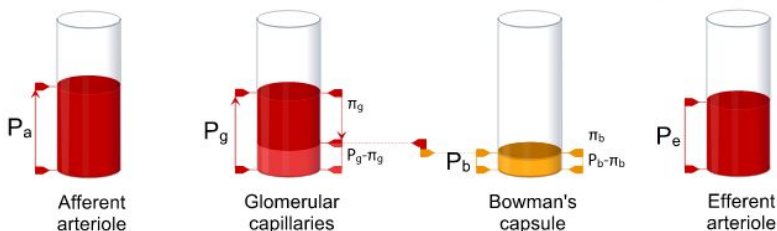
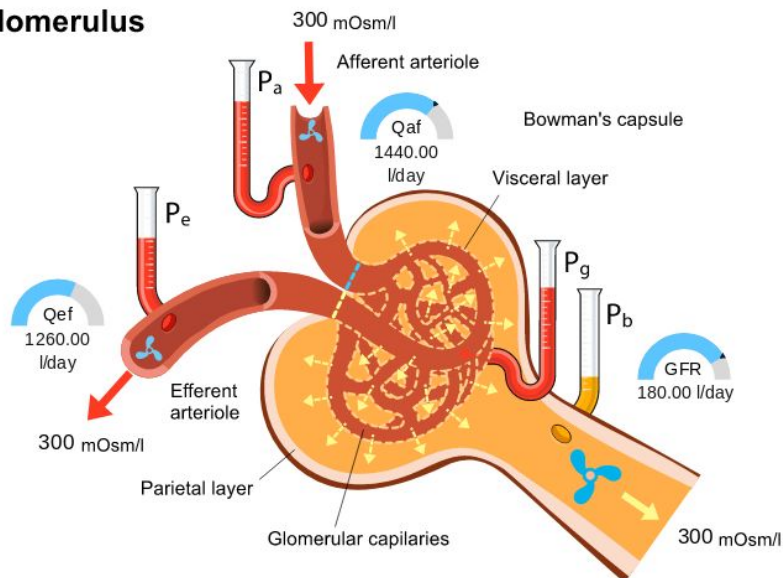
Obrázek 8: Bodylight.js Composer

Nové simulátory

Pomocí Bodylight.js jsme již vytvořili několik výukových aplikací. Simulátor nefronu [16] je zatím nejkomplexnější. Je dostupný na adrese physiome.cz/apps/Nephron. Vysvětluje tvorbu moči – exkreci a resorpci vody a sodíku v různých částech nefronu. Další soluty nejsou zahrnuty. Skládá se z několika nezávislých simulačních obrazovek: glomerulus, proximální tubulus, Henleova klička, distální tubulus a sběrný kanálek, celý nefron. Zde si ukážeme jen první a poslední obrazovku.

Simulační obrazovka glomerulu (obrázek 9) ukazuje, jak je glomerulární filtrace (GFR) regulována variabilními odpory aferentní a eferentní cévy. Uživatel může ovládat tyto odpory stejně jako arteriální tlak a filtrační koeficient. Model počítá tlaky a průtoky v systému. “Skleněné” sloupce v animaci glomerulu ukazují hydraulický tlak. Válce jsou znázorněny modrými půlkruhovými indikátory a rychlostí otáčení vrtulek. Velikost GFR navíc ukazuje šířka žlutých čárkovaných šipek. Rozsah posuvníků pro ovládání odporu je zvolen tak, aby jejich pomocí bylo možné regulovat GFR na požadovaných 180 l/d jen v rozsahu MAP 80-160 mmHg, což odpovídá rozsahu uváděnému v literatuře.

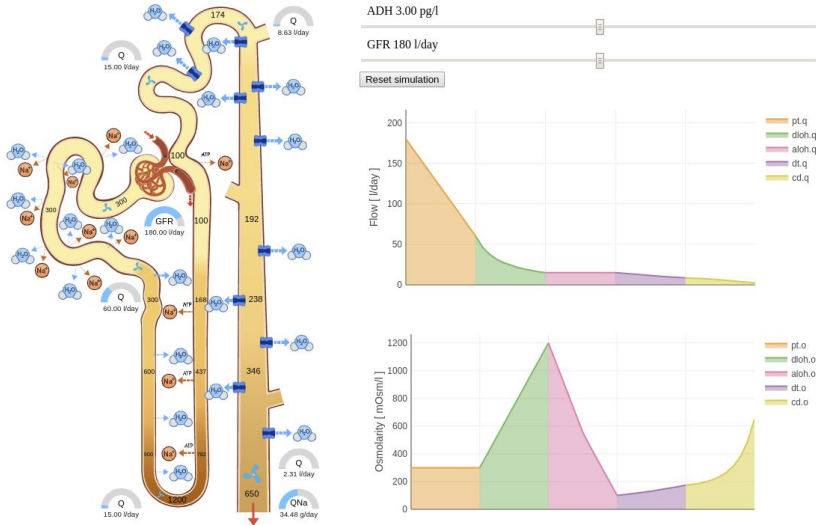
Glomerulus



Obrázek 9: Simulační obrazovka glomerulu

Poslední obrazovka simulátoru – kompletní nefron – je na obrázku 10. V proximálním tubulu se isoosmoticky resorbuje voda a sodík. V sestupné části klíčky z tubulu díky osmotickému gradientu pasivně odchází voda. Sodík zůstává, moč se koncentruje a osmolarita uvnitř tubulu se vyrovnává s okolní. Ve vzestupné části klíčky dochází k aktivní resorpci sodíku, takže osmolarita klesá. V distálním tubulu a sběrném kanálku odchází voda pasivně po koncentračním gradientu do dřeně. Propustnost stěn těchto tubulů je řízena hladinou ADH. Při minimálním ADH jsou tubuly nepropustné a koncentrace filtrátu se nemění. Naopak

při maximálním ADH se resorbuje tolik vody, že se osmolarita v tubulu a okolní dřeni téměř vyrovnává.



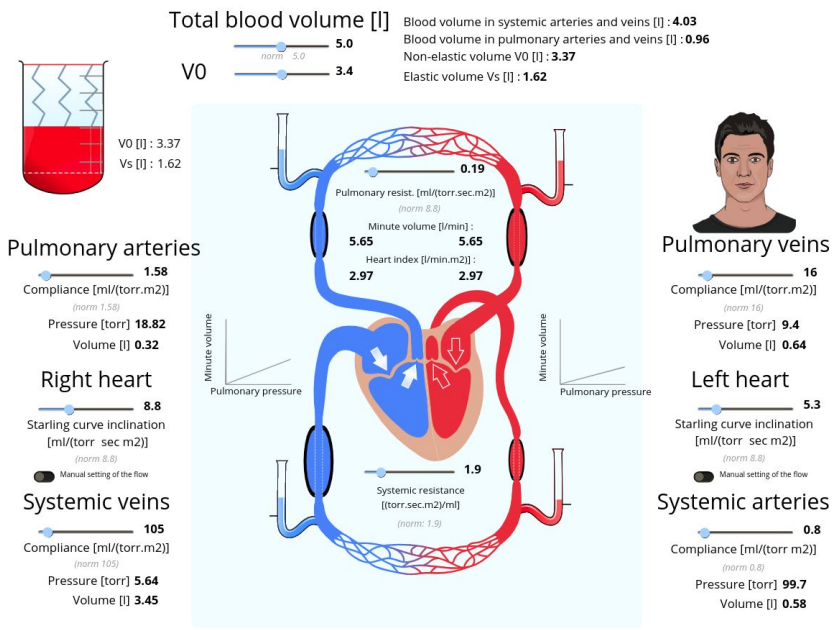
Obrázek 10: Simulační obrazovka celého nefronu

Uživatel může ovládat koncentraci ADH a GFR. Osmolarita v tubulech je znázorněna barevným odstínem i číselně. Půlkruhové indikátory ukazují průtok vody a hmotnostní tok vylučovaného sodíku. Prostupnost distálního tubulu a sběrného kanálku pro vodu je indikována šířkou modrých kanálek ve stěně tubulů. Profil osmolarity a průtoku filtrátu podél celého systému tubulů je vykreslen v grafech. Jednotlivé části nefronu jsou barevně odlišeny.

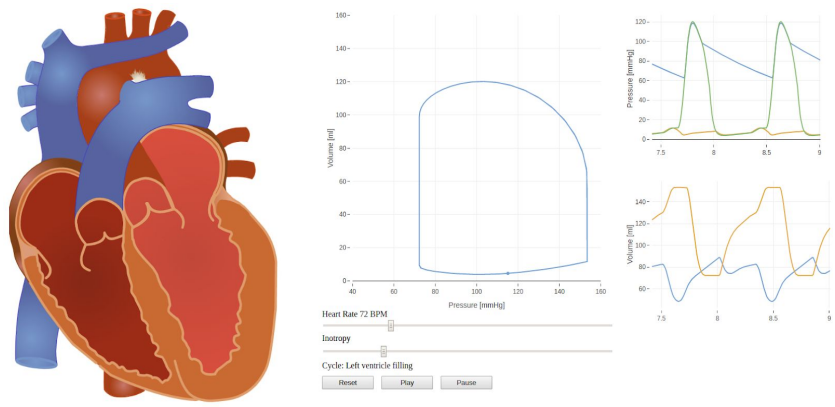
Tento simulátor byl použit při výuce patofyziologie na našem ústavu. Studenti i vyučující vyplňovali na závěr hodiny dotazník. Simulátor považovali za přínosný a obecně ho hodnotili velice kladně.

Další simulátory vytvořená s pomocí Bodylight.js jsou na obrázcích 11 a 12 a dostupné na physiome.cz/apps.

Bodylight.js se při tvorbě simulátorů velice osvědčil. Vytvoření podobných simulačních aplikací by bylo bez použití nástroje jako je Bodylight.js extrémně pracné, ne-li nemožné.



Obrázek 11: Simulátor cirkulace



Obrázek 12: Simulátor levého srdce

Závěr

Práce řeší dvě témata. První část popisuje nové jazykové rozšíření jazyka Modelica pro modelování pomocí parciálních diferenciálních rovnic zvané PDEModelica1. Podpora pro toto jazykové rozšíření založená na metodě přímek byla implementována v simulačním nástroji OpenModelica. Rozšíření bylo využito ve dvou popsanych modelech. První je model protiproudové tepelné výměny mezi tepnou a žílou v noze ptáka. Druhý je model dýchání osoby zasypané ve sněhové lavině. Pomocí PDEModelica1 se zde modeluje advekce a difuze O_2 a CO_2 ve sněhu. Jazykové rozšíření se pro modelování a simulaci obou problémů osvědčilo.

Druhá část práce se zabývá tvorbou výukových simulátorů. Je zde popsána nová sada nástrojů Bodylight.js pro tvorbu webových klientských simulátorů založených na modelech v jazyce Modelica. Hlavní důraz je kladen zejména na nově vyvinutý nástroj Bodylight.js Composer, který umožňuje dát dohromady předpřipravený model a animace, vytvořit grafické uživatelské rozhraní, vše dohromady funkčně propojit a vyexportovat hotový simulátor.

Reference

1. Kofránek J, Vu LDA, Snaselova H, Kerekes R, Velan T. GOLEM-multimedia simulator for medical education. Stud Health Technol Inform. IOS Press; 1999; 2001; 1042–1046.
2. Kurtz TW, DiCarlo SE, Pravenec M, Ježek F, Šilar J, Kofránek J, et al. Testing Computer Models Predicting Human Responses to a High-Salt Diet. Hypertension. Lippincott Williams & WilkinsHagerstown, MD; 2018; doi:10.1161/HYPERTENSIONAHA.118.11552
3. Mateják M, Kulhánek T, Šilar J, Privitzer P, Ježek F, Kofránek J.

Physiolibrary - Modelica library for Physiology. Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden. Linköping University Electronic Press; 2014. pp. 499–505.

4. Hester RL, Brown AJ, Husband L, Iliescu R, Pruett D, Summers R, et al. HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology. *Front Physiol.* 2011;2: 12.
5. Stavåker K, Fritzson P, Song C, Wlotzka M, Heuveline V. PDE Modeling with Modelica via FMI Import of HiFlow3 C Components with Parallel Multi-core Simulations [Internet]. *SNE Simulation Notes Europe.* 2015. doi:10.11128/sne.25.tn.10316
6. Dshabarow F, Cellier FE, Zimmer D, Dshabarow F, Com C, Ch IE. Support for Dymola in the modeling and simulation of physical systems with distributed parameters. Proceedings of the 6th International Modelica Conference. 2007. pp. 683–690.
7. Schiesser WE. *The Numerical Method of Lines: Integration of Partial Differential Equations.* Elsevier; 2012.
8. Eymard R, Gallouët T, Herbin R. Finite volume methods. *Handbook of Numerical Analysis.* Elsevier; 2000. pp. 713–1018.
9. Saldamli L. *PDEModelica--A High-Level Language for Modeling with Partial Differential Equations.* Institutionen för datavetenskap. 2006.
10. Mitchell JW, Myers GE. An analytical model of the counter-current heat exchange phenomena. *Biophys J.* 1968;8: 897–911.
11. Šilar J, Ježek F, Kofránek J. PDEModelica1: a Modelica language extension for partial differential equations implemented in OpenModelica [Internet]. *International Journal of Modelling and Simulation.* 2018. pp. 128–137. doi:10.1080/02286203.2017.1404417
12. Roubík K, Sieger L, Sykora K. Work of breathing into snow in the presence versus absence of an artificial air pocket affects hypoxia and hypercapnia of a victim covered with avalanche snow: a randomized double blind crossover study. *PLoS One. Public Library*

of Science; 2015;10: e0144332.

13. Šilar J, Ježek F, Kofránek J. PDEModelica and Breathing in an Avalanche. Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017. Linköping University Electronic Press; 2017. pp. 367–372.
14. Šilar J, Polák D, Mládek A, Ježek F, Kurtz TW, DiCarlo SE, et al. Development of In-Browser Simulators: Introduction of a Novel Software Toolchain (Preprint) [Internet]. doi:10.2196/preprints.14160
15. Zakai A. Emscripten: An LLVM-to-JavaScript Compiler. Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion. New York, NY, USA: ACM; 2011. pp. 301–312.
16. Šilar J, Ježek F, Mládek A, Polák D, Kofránek J. Model visualization for e-learning, Kidney simulator for medical students. Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4--6, 2019. Linköping University Electronic Press; 2019. Available:
<https://www.ep.liu.se/ecp/article.asp?issue=157&article=040&volume=>

Vlastní přínos autora

- Návrh jazykového rozšíření PDEModelica, které proti původnímu rozšíření zavádí několik změn a vylepšení
- Implementace podpory pro PDEModelica1 v nástroji OpenModelica
- Vývoj modelu protiproudové výměny tepla mezi tepnou a žílou
- Vývoj modelu advekce a difuze CO₂ a O₂ ve sněhu
- Implementace simulačního běhového prostředí pro technologii Bodylight(.NET)
- Účast na vývoji technologie Bodylight.js
- Návrh a implementace modelů a simulátoru nefronu
- Reimplementace modelu Guyton 72 a vytvoření nových simulačních protokolů

Seznam publikací

S IF, součástí práce

- Kurtz TW, DiCarlo SE, Pravenec M, Ježek F, Šilar J, Kofránek J, et al. Testing Computer Models Predicting Human Responses to a High-Salt Diet. Hypertension. Lippincott Williams & WilkinsHagerstown, MD; 2018; doi:10.1161/HYPERTENSIONAHA.118.11552
IF 6.823
- Šilar J, Polák D, Mládek A, Ježek F, Kurtz TW, DiCarlo SE, Živný J, Kofranek J. Development of In-Browser Simulators: Introduction of a Novel Software Toolchain. Journal of Medical Internet Research. (forthcoming/in press) DOI: 10.2196/14160
IF 4.945

Bez IF, součástí práce

- Šilar, J., Ježek, F., Mládek, A., Polák, D., Kofránek, J. Model visualization for e-learning, Kidney simulator for medical students. In Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019 (No. 157). Linköping University Electronic Press.
- Šilar, Jan, Filip Ježek a Jiří Kofránek. PDEModelica1: a Modelica language extension for partial differential equations implemented in OpenModelica. International Journal of Modelling and Simulation [online]. 2017, 38(2), 128-137 [cit. 2018-10-18]. DOI: 10.1080/02286203.2017.1404417. ISSN 0228-6203.
- Šilar, Jan, Filip Ježek a Jiří Kofránek. PDEModelica and Breathing in an Avalanche. Proceedings of the 12th International Modelica

Conference, Prague, Czech Republic, May 15-17, 2017, Linköping University Electronic Press, Linköpings universitet, str. 367-372, ISBN 978-91-7685-575-1, Linköping Electronic Conference Proceedings, ISSN (print) 1650-3686, ISSN (online) 1650-3740, DOI 10.3384/ecp17132367.

- Šilar, Jan, Kofránek, Jiří: Physiological Model Creation and Sharing. European Journal for Biomedical Informatics (EJBI), ISSN 1801-5603, volume 7 (2011), str. 55-58.

S IF, další

- Heřmanová Z, Kvaček J, Halamski AT, Zahájská P, Šilar J. Reinterpretation of fossil reproductive structures *Zlivifructus microtriasseris* (Normapolles complex, Fagales) from the Czech and Polish Late Cretaceous. Review of Palaeobotany and Palynology (in press)
IF 1.674

Bez IF, další

- Ježek, F., Tribula, M., Kulhánek, T., Mateják, M., Privitzer, P., Šilar, J., Kofránek, J. (2015), Surviving Sepsis - a 3D integrative educational simulator, Proceedings of 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, Milano (It), pp. 3679-3682. ISBN: 978-1-4244-9270- 1, ISSN: 1557- 170X.
- T. Kulhanek, M. Matejak, J. Silar, and J. Kofranek. Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud. In Biomedical and Health Informatics (BHI), 2014 IEEE EMBS International Conference on, pages 712–715, June 2014.
- Marek Mateják, Tomáš Kulhánek, Jan Šilar, Pavol Privitzer, Filip Ježek, Jiří Kofránek: Physiobrary - Modelica library for Physiology, In

Conference Proceeding, 10th International Modelica Conference 2014, March 12, 2014

- Tomáš Kulhánek, Marek Mateják, Jan Šilar, Jiří Kofránek: Identifikace fyziologických systémů, sborník příspěvků MEDSOFT 2014, ISSN 1803-8115, 148-153
- Šilar J., Staváker K., Mateják M., Privitzer P., Nagy J.: Modeling with Partial Differential Equations - Modelica Language Extension Proposal. OpenModelica Annual Workshop February 3, 2014
- Kofránek J., Privitzer P., Tribula M., Šilar J., Kulhánek T., Mateják M., Ježek F.: Užitený vzor: „Simulátor virtuálního pacienta“, Úřad průmyslového vlastnictví, číslo přihlášky 2014-30329, číslo zápisu: 27613, datum zápisu 11.12. 2014.
- Kulhánek T., Mateják M., Šilar J., Privitzer P., Tribula M., Ježek F., Kofránek J.: Hybridní architektura pro webové simulátory. MEDSOFT 2013, str. 115-121, ISSN 1803-8115
- Ježek F., Tribula M., Kofránek J., Kolman J., Privitzer P., Šilar J.: Sada výukových simulátorů - výsledky vývoje frameworku BodyLight. MEDSOFT 2013, str. 38-48, ISSN 1803-8115.
- Šilar J., Kofránek J., Kulhánek T., Forgáčová K., Nečas E.: Modelování obnovy kmenových buněk. MEDSOFT 2013, str. 185-191, ISSN 1803-8115.
- Privitzer P., Šilar J., Kulhánek T., Mateják M., Kofránek J.: Simulation Applications in Medical Education. EFMI STC 2013 Prague 17-19 April 2013. WS1 workshop.
- Kulhánek T., Mateják M., Šilar J., Privitzer P., Tribula M., Ježek F., Kofránek J.: Hybrid architecture for web simulators of pathological physiology. EFMI STC 2013 Prague 17-19 April 2013. WS1 workshop.
- Kulhanek T., Matejak M., Silar J., Privitzer P., Tribula M., Jezek F., Kofranek J.: RESTful web service to build loosely coupled web based

simulation of human physiology: IEEE EMBC 2013, Osaka, Japan 3-7 July 2013, published in Transactions of Japanese Society for Medical and Biological Engineering Vol. 51 (2013) No. Supplement p. R-32, ONLINE ISSN: 1881-4379 DOI 10.11239/jmsbe.51.R-32

- Kofránek, J., Mateják, M., Privitzer, P., Tribula, M., Kulhánek, T., Šilar, J., Pecinovský, R., HumMod-Golem Edition: large scale model of integrative physiology for virtual patient simulators. Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13), 182-188
- Šilar, Jan, Vavrek, Martin, Kulhánek, Tomáš, Privitzer, Pavol, Kofránek, Jiří, Kroček, Tomáš, Tribula, Martin: Výpočty na grafických procesorech, řešení parciálních diferenciálních rovnic. In MEDSOFT 2012, (Milena Ziethamlová Ed.) Praha: Agentura Action M, Praha, str. 252-259, ISSN 1803-8115.
- Kofránek, Jiří, Mateják, Marek, Ježek, Filip, Privitzer, Pavol, Šilar, Jan: Výukový webový simulátor krevního oběhu. In MEDSOFT 2011(Milena Ziethamlová Ed.) Praha: Agentura Action M, Praha, str. 106-121, ISSN 1803-8115.
- Privitzer, Pavol, Šilar, Jan, Tribula, Martin, Kofránek, Jiří: Od modelu k simulátoru v internetovém prohlížeči. In MEDSOFT 2010 (Milena Ziethamlová Ed.) Praha: Agentura Action M, Praha, str. 149-169, ISSN 1803-8115.
- Tomáš Kulhánek, Jan Šilar, Marek Mateják, Pavol Privitzer: Od výukového modelu k identifikaci fyziologického systému. In Sborník konference MEFANET 2011, (Daniel Schwarz, Martin Komenda, Jaroslav Majerník, Stanislav Štípek, Vladimír Mihál, Ladislav Dušek, Eds.), ISBN978-80-7392-141-5, Masarykova Univerzita, Brno CD ROM
- Tomáš Kulhánek, Jan Šilar, Marek Mateják, Pavol Privitzer, Jiří Kofránek, Martin Tribula: Distributed computation and parameter estimation in identification of physiological systems, proceeding of

VPH 2010, extended abstract, 30th Sep - 1st Oct 2010 Brussels,
Belgium