

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

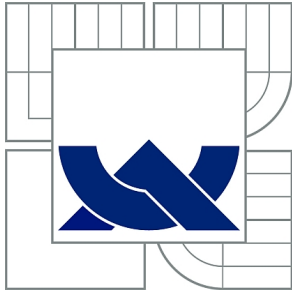
SIMULACE FARMAKOKINETICKÝCH MODELŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

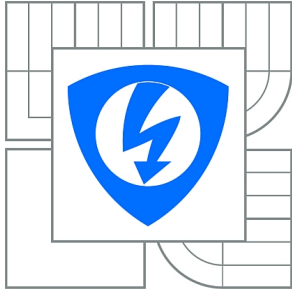
AUTOR PRÁCE
AUTHOR

JAKUB HEJČ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

SIMULACE FARMAKOKINETICKÝCH MODELŮ

SIMULATION OF PHARMACOKINETIC MODELS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

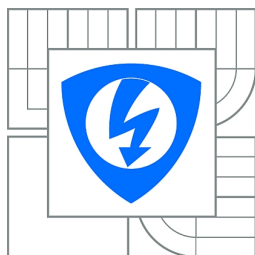
AUTOR PRÁCE
AUTHOR

JAKUB HEJČ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MARTIN MÉZL

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor

Biomedicínská technika a bioinformatika

Student: Jakub Hejč

ID: 106152

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Simulace farmakokinetických modelů

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte základní principy a metody používané při modelování farmakokinetických dějů. Zaměřte se na modely používané v perfúzním zobrazení. V programovém prostředí Matlab realizujte simulaci vybraných modelů. Pomocí vhodné optimalizační metody navrhnete algoritmus pro proložení experimentálních dat vhodným modelem.

Vytvořte grafické rozhraní umožňující simulaci vybraných modelů a automatické proložení naměřených dat vhodným modelem a výpočet základních farmakokinetických parametrů. Zhodnoťte vlastnosti jednotlivých modelů pro použití v perfúzním zobrazení.

DOPORUČENÁ LITERATURA:

[1] JIŘÍK, R.: Modely v biologii a epidemiologii. Elektronická skripta. Brno: VUT, 2006.

[2] HOLČÍK, J.: Modelování biologických systémů. Elektronické texty. Brno: VUT, 2001

[3] MISCHI, Massimo, DEL PRETE, Zaccaria, KORSTEN, Hendrikus H. M. Biomechanical System Technology : Cardiovascular Systems. Cornelius T. Leondes. Singapore : World Scientific Publishing, 2007. ISBN 981-270-981-9. Indicator Dilution Techniques in Cardiovascular Quantification, s. 89-156.

[4] FLETCHER, R. Practical Methods of Optimization. 2nd edition. New York : Wiley, 2000. 450 s. ISBN 978-0-471-49463-8.

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Martin Mézl

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Teoretická část projektu se zabývá rozbořem farmakokinetických dějů a základními vlastnostmi matematických modelů užívaných ve farmakokinetice. Tento popis je zaměřen především na modely využívané pro perfúzní zobrazovací metody. Cílem projektu je vytvořit algoritmus pro simulaci vybraných modelů na základě zadaných parametrů a algoritmus sloužící k proložení experimentálně měřených dat vybraným modelem s výpočtem základních farmakokinetických parametrů. Dalším krokem řešení je realizace grafického rozhraní, které umožní plně využívat vytvořené algoritmy v uživatelsky přístupnějším prostředí. Výstupem práce je program, který lze využít k získání parametrů reálných dat jako názorná ukázka vlivu těchto parametrů na průběh zvolených funkcí.

KLÍČOVÁ SLOVA

matematické modely, farmakokinetické modely, perfúzní zobrazení, gamma funkce, log-normální funkce, LDRW model, simulace modelů, proložení dat, optimalizace funkce

ABSTRACT

The theoretical part of this project is occupied with analysis of pharmacokinetic actions and also basic attributes of mathematical models used in pharmacokinetics. This description is mainly focused on models used for perfusion imaging methods. The aim of this project is to create an algorithm that simulates chosen models based on assigned parameters and also an algorithm that serves to fit experimentally measured data with a chosen model with a calculation of basic pharmacokinetic parameters. The next step of this solution is graphic interface realization which enables a full use of created algorithm in more accessible surroundings for the user. The result of this work is a program that can be used to obtain real data parameters and as well as a visual sample of the influence of these parameters on a process in a chosen functions.

KEYWORDS

mathematical models, pharmacokinetics models, perfusion imaging, gamma function, log-normal function, LDRW model, simulation of models, data fitting, function optimization

HEJČ, Jakub *Simulace farmakokinetických modelu*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2010. vi s. Vedoucí práce byl Ing. Martin Mézl,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Simulace farmakokinetických modelu jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych tímto poděkovat za poskytnutí materiálů k práci, rady pro její vypracování a trpělivost vedoucímu práce Ing. Martinu Mézlovi.

V Brně dne 31. května 2010

.....

podpis

OBSAH

Úvod	8
1 Matematické modely	9
1.1 Klasifikace matematických modelů	9
2 Farmakokinetické modely	12
2.1 Základní farmakokinetické parametry	12
2.2 Kompartmentové modely pohybu léčiva v organismu	15
2.2.1 Jednokompartmentový model	16
2.2.2 Dvoukompartmentové modely	17
2.3 Modely v perfuzním zobrazení	20
2.3.1 Gamma funkce	22
2.3.2 Log-normální funkce	24
2.3.3 Local density random walk model	25
3 Realizace programové části	28
3.1 Popis programovacího prostředí	28
3.2 Simulace farmakokinetických modelů	29
3.2.1 Struktura programu	29
3.2.2 Funkční a grafické řešení	30
3.2.3 Ověření výstupu skriptu a provedení simulace	33
3.3 Data fitting a výpočet parametrů	34
3.3.1 Optimalizace funkce	35
3.3.2 Popis a zpracování dat	36
3.3.3 Struktura programu	38
3.3.4 Funkční a grafické řešení	39
3.3.5 Proložení experimentálních dat	43
4 Závěr	45
Literatura	46
A Grafické přílohy	i
A.1 Strukturní schéma hlavního skriptu	i
A.2 Grafické rozhraní pro simulaci	ii
A.3 Grafické rozhraní pro proložení dat modelem	iii
A.4 Ukázka proložení dat z ultrazvukového fantomu	iv
A.5 Ukázka proložení nefiltrovaných dat ze srdečních oddílů	v
A.6 Ukázka proložení filtrovaných dat ze srdečních oddílů	vi

ÚVOD

Cílem perfuzních zobrazovacích metod je zachytit vztah mezi anatomickou strukturou a jejím funkčním projevem. Princip vychází ze stanovení míry prokrvení. To se děje na základě sledování průběhu koncentrace kontrastní látky v krevním řečišti. Perfuzní analýza poté umožňuje získat parametry tkáně informující o změnách v krevním průtoku či jejím metabolismu. Toho lze v dnešní době využít především k diagnostice nádorových onemocnění či obstrukcí v cévním systému. Protože z reálných průběhů dat lze jen obtížně stanovit potřebné parametry, využívá se pro jejich aproximaci farmakokinetických modelů. Mezi nejhojněji využívané patří modely vycházející z matematických průběhů. Jedním z nich je log-normální model a gamma model, který má již spíše historické postavení. Často používaným modelem je také Local density random walk, který vychází z popisu diluce částic.

Část práce se zabývá teorií matematických modelů, základů farmakokinetiky a využití některých jejích metod v perfuzním zobrazení. První část je úvodem do základních pojmů z oblasti modelů. Obsahuje základní dělení modelů podle jejich vlastností a matematické výrazy používané pro jejich popis. Dále jsou nastíněny základy o modelování pohybu léčiva v organismu. Jsou popsány základní vztahy farmakokinetiky a některé výpočty důležitých parametrů. Uvedena je teorie tří základních modelů, které se pro popis perfuze využívají.

Hlavním část práce je věnována samotnému řešení. Cílem je vytvořit soubor funkcí v programovém prostředí Matlab, které umožní simulaci vybraných modelů na základě zvolených parametrů. Dále vytvořit skript, který má za úkol zvolenými modely prokládat experimentálně naměřená a stanovit parametry modelů. Posledním úkolem je realizovat pro tento balík funkcí grafické rozhraní umožňující snadný uživatelský přístup. Popis je věnován především samotnému vytváření funkcí a realizaci uživatelského rozhraní. Je nastíněna problematika optimalizačních úloh a jejich aplikace v rámci této práce.

1 MATEMATICKÉ MODELY

Matematickým modelem se rozumí popis vybraných vlastností studovaného objektu pomocí matematických výrazů. Tyto výrazy zahrnují množiny proměnných, konstant a rovnic a jsou jimi vyjádřeny vztahy mezi jednotlivými prvky zkoumaného systému, jejich chování či struktura.[17]

Činnost, která vede k sestavení takového modelu nazýváme *modelováním*. Modelování přispívá k získání představ o vazbách mezi nejrůznějšími systémy, částmi systémů, či získávání jiných parametrů souvisejících s jejich činností. S využitím matematických modelů se lze setkat téměř v každém vědním oboru od biologie, ekonomie až po studii sociálního chování, znečištění apod., kde umožňují zkoumat složité a jiným způsobem těžko postižitelné jevy.[3]

To, zda model správně interpretuje zkoumanou skutečnost, probíhá za pomoci jeho *simulace*. Tj. vyhodnocení jeho výstupní funkce po zavedení počátečních podmínek. K tomu se nejčastěji využívá diskrétního modelu (viz dále) za použití výpočetní techniky.

1.1 Klasifikace matematických modelů

Veličiny reprezentované v matematickém modelu mohou vykazovat různé vlastnosti na základě jejich vzniku či průběhu. Pro bližší popis procesů probíhajících v lidském (nebo obecně kterémkoliv) organismu a jeho lepší pochopení je uvedeno následné základní rozdělení.

Matematické modely lze klasifikovat podle:

1. Typu vyskytujících se veličin:

- *Deterministické* – jevy reprezentované v modelu jsou nenáhodné, tzn. jsou vždy definovány stejným způsobem a při simulaci za stejných počátečních podmínek je i výsledek stejný.
- *Stochastické* – kde je alespoň jedna veličina nebo funkce náhodná, tedy je dána pravděpodobností výskytu jevu, který reprezentuje.

U biologických systémů se můžeme setkat s oběma typy veličin. Příkladem může být absorpce živin v trávicího traktu. Samotný jev je deterministický, protože se vyskytuje u všech subjektů, rychlost absorpce má však stochastický charakter – její velikost je ovlivněna mnoha vedlejšími vlivy a nelze ji jednotně kvantifikovat pro všechny situace.[3]

2. Závislosti a vztahu k času:

- *Statické* – časová závislost změny veličiny není u statických modelů zohledněna. Je popsána pouze v ustáleném stavu nebo po delším časovém úseku a její hodnota nezávisí na hodnotách předchozích (model systému bez paměti). Příkladem může být vyjádření neměnné struktury, či jevy, u kterých můžeme časovou závislost zanedbat.[17]

- *Dynamické* – zahrnují parametr času a postihují tedy vývoj sledované veličiny v čase. Nabývání hodnot je závislé na předchozím stavu veličiny, jedná se proto o model systému s pamětí.

3. Závislosti změny prvků modelu:

- *Lineární* – pokud platí princip superpozice, tzn. že výstup modelu je přímo úměrně závislý na vstupních proměnných.
- *Nelineární* – model, jehož prvky nevykazují lineární závislost.

Biologické systémy jsou obecně nelineární. V určitém rozsahu veličiny lze však zanedbat drobné odchylky a tento systém reprezentovat lineárním modelem. Linearita je proto pouze aproximací reálného chování, avšak s dostatečnou přesností.

4. Spojitosti v čase:

- *Spojité* – jsou dány spojitou funkcí spojitého času, prvky takového modelu jsou definovány v každém nekonečně malém časovém okamžiku[17].
- *Diskrétní* – veličiny takového modelu jsou definovány pouze v určitých krátkých časových okamžicích. Jedná se zpravidla o vzorkovaný signál v podobě číselné posloupnosti.

Pro potřeby simulace modelů budou využity pouze modely diskrétní. Vyplývá to z nutnosti využití programového vybavení, které nedokáže principiálně se spojitým časem pracovat. Vzorky jsou však natolik málo od sebe vzdáleny, že výstupní funkci můžeme zjednodušeně uvažovat jako spojitou.

5. Popisu procesů:

- *Vnější popis* – model nerespektuje děje probíhající uvnitř a zachycuje pouze vnější projev studovaného systému. Lze ho proto nazvat také vstupní/výstupní model[3].
- *Vnitřní popis (stavový)* – vyjadřuje svými proměnnými závislosti jevů probíhajících uvnitř systému. Tím je schopný zachytit také jeho strukturu.

Aby bylo možné lépe zachytit interakce ve sledovaném systému je dominantně využíván popis stavový.

V následujícím textu je uvedena teorie stavových rovnic, používaných pro popis průběhu látky lidským tělem.

Dynamický systém je dán obecně stavovými rovnicemi, které určují vztah mezi jednotlivými veličinami, které tento systém definují. První rovnici označujeme jako stavovou (diferenciální rovnice 1. řádu) a pro lineární spojitý systém platí [3]:

$$\frac{\delta x}{\delta t} = Ax(t) + Bu(t), \quad (1.1)$$

kde A je matice vnitřních vazeb systému, B je matice vazeb systému na vstup, u vstupní proměnná a x stavová proměnná.

Druhou rovnici označujeme jako výstupní a pro lineární spojitý systém platí:

$$y(t) = Cx(t) + Du(t), \quad (1.2)$$

kde C je matice vazeb výstupu na stav, D matice vazeb vstupu na výstup a $y(t)$ výstup systému v čase t .

Pro diskretní modely za předpokladu linearity je stavový popis definován diferenční rovnicí 1. řádu a výstupní rovnicí a lze pak obdobně psát:

$$x(nt + t) = A \cdot x(nt) + B \cdot u(nt), \quad (1.3)$$

$$y(nt) = C \cdot x(nt) + D \cdot u(nt), \quad (1.4)$$

kde konstanty A, B, C, D a proměnné u, x, y mají stejný význam jako v rovnicích (1.1) a (1.2).

2 FARMAKOKINETICKÉ MODELY

Matematické modely využívané pro popis distribuce léčiva (farmaka) v lidském organismu po jeho aplikaci se nazývají *farmakokinetické modely*. Při aplikaci farmaka dochází v organismu k pochodům, které distribuci farmaka v průběhu času z větší či menší míry ovlivňují. Tyto změny pak mohou (ale nutně nemusí) být doprovázeny také odpovídajícími změnami koncentrace. Koncentrace je ve většině případech hlavním zkoumaným parametrem, a to především z hlediska její časové a prostorové změny.[2]

Cílem farmakokinetického modelu je tedy popsat děje ovlivňující distribuci farmaka vytvořením abstraktních modelů za pomoci zjednodušeného matematického popisu a vyjádřit tak změnu koncentrace farmaka v průběhu času. To nejčastěji probíhá analýzou koncentrací léčiva v tělních tekutinách.

Děje ovlivňující tyto změny lze rozdělit mezi tři základní pochody:

- 1. Resorpce** – vyjadřuje přestup léčiva z místa aplikace do krevního řečiště. Příkladem může být vstřebávání léku z trávicího traktu. Než se farmakum dostane do cílové tkáně, musí překonat biologické překážky oddělující vnější prostředí od vnitřního (sliznice, buněčné membrány, buněčné struktury). Tím dochází ke zpoždění účinku a vazbám farmaka na jiné než cílové struktury.
- 2. Distribuce** – je proces, při němž dochází k rozšíření léčiva z krve do ostatních tkání a orgánů, případně k jeho vazbě jiné struktury (nejčastěji plazmatické bílkoviny). Léčivo má tedy tendenci se postupem času rovnoměrně rozptýlit po celém organismu. Největší vliv na rychlost rozptýlení má perfuze (prokrvení) tkání, jelikož hlavní cestou distribuce léčiva je krevní oběh.[2]
- 3. Eliminace** – zahrnuje všechny pochody, přispívající k odstranění látky z organismu (vyučování a biotransformace). To vede k postupnému snižování koncentrace léčiva v organismu. Na těchto dějích se z největší části podílí funkce jater a ledvin. Eliminace ledvinami zahrnuje procesy glomerulární filtrace, aktivní exkrece a resorpce. Játra se snaží látky eliminovat nejčastěji vazbou na jiné látky. Kromě toho existují ještě další cesty eliminace např. slinami či potem.

Prostor, do kterého se farmakum šíří, lze určit několika způsoby. Pro některé modely platí, že obecně lze lidský organismus rozdělit na oddíly, mezi kterými změna distribuce na základě zmíněných pochodů probíhá. Tyto oddíly se nazývají kompartmenty a jejich význam bude probrán dále. Platí však, že za takový oddíl lze považovat jak celé tělo člověka, tak jednotlivé subsystemy jako například krevní oběh, či intersticiální tekutinu.[?]

2.1 Základní farmakokinetické parametry

Farmakokinetické parametry jsou proměnné, které podávají představu o chování a vlastnostech farmaka v organismu a popisují základní jevy, které toto chování ovlivňují. Jejich využitím lze docílit sestavení fungujícího modelu, nebo zpětně při simulaci modelu konstruovaného pro danou situaci určit některý ze základních

parametrů a usuzovat tak na vlastnosti daného farmaka nebo prostředí, ve kterém se distribuuje.[1]

Farmakokinetické parametry dělíme podle jejich závislosti na *primární* a *sekundární*.

Primární parametry závisí na fyziologických pochodech jako proudění krve, glomerulární filtrace a patří mezi ně:

- distribuční objem – V_d
- clearance – CL

Sekundární parametry závisí na parametrech primárních a jsou to především:

- plazmatický poločas eliminace – $t_{1/2}$
- plocha po křivkou – AUC
- absolutní biologická dostupnost – F_{Abs}
- absorpční a eliminační konstanta

Distribuční objem V_d

Distribuční objem vyjadřuje poměr mezi množstvím farmaka v organismu vůči jeho koncentraci v krevním oběhu. Vyjadřuje tedy hypotetický objem tekutiny, ve kterém by se farmakum muselo rozpustit, aby dosáhlo stejné koncentrace jako v plazmě [1]. Je tedy definován jako:

$$V_d = \frac{D}{c} \text{ [l/kg]}, \quad (2.1)$$

kde D je známá velikost aplikované dávky, tedy množství farmaka v těle a c koncentrace farmaka v plazmě.

Clearance CL

Clearance charakterizuje výkonnost eliminačních orgánů a zahrnuje všechny eliminační procesy. Udává, jaké množství tekutiny bude očištěno od aplikovaného farmaka za jednotku času, resp. jakou rychlostí proběhne tato eliminace a změní se jeho koncentrace v plazmě nebo obecně v biologické tekutině. Z toho vyplývá, že:

$$CL = \frac{v_e}{c} \text{ [ml/min]}, \quad (2.2)$$

kde v_e je rychlost eliminace udávána v jednotkách za čas, CL clearance a c je koncentrace látky v krvi.

Plazmatický poločas eliminace $t_{1/2}$

Vyjadřuje časový úsek, po kterém klesne koncentrace sledované látky v plazmě na polovinu své původní hodnoty. Jak již z názvu vyplývá, je tento parametr závislý na procesu eliminace.

Při vyjádření tohoto procesu eliminační konstantou proto platí vztah:

$$t_{1/2} = \frac{\ln 2}{k_e} = \frac{0,693}{k_e} \text{ [min]}, \quad (2.3)$$

kde k_e je eliminační konstanta, která je určena poměrem primárních parametrů a pro definovaný distribuční objem V_d platí tedy také:

$$t_{1/2} = \frac{0,693 \cdot V_d}{CL} \text{ [min]}. \quad (2.4)$$

Z toho vyplývá, že čím delší bude poločas eliminace, tím větší bude distribuční objem V_d , případně nižší clearance CL .

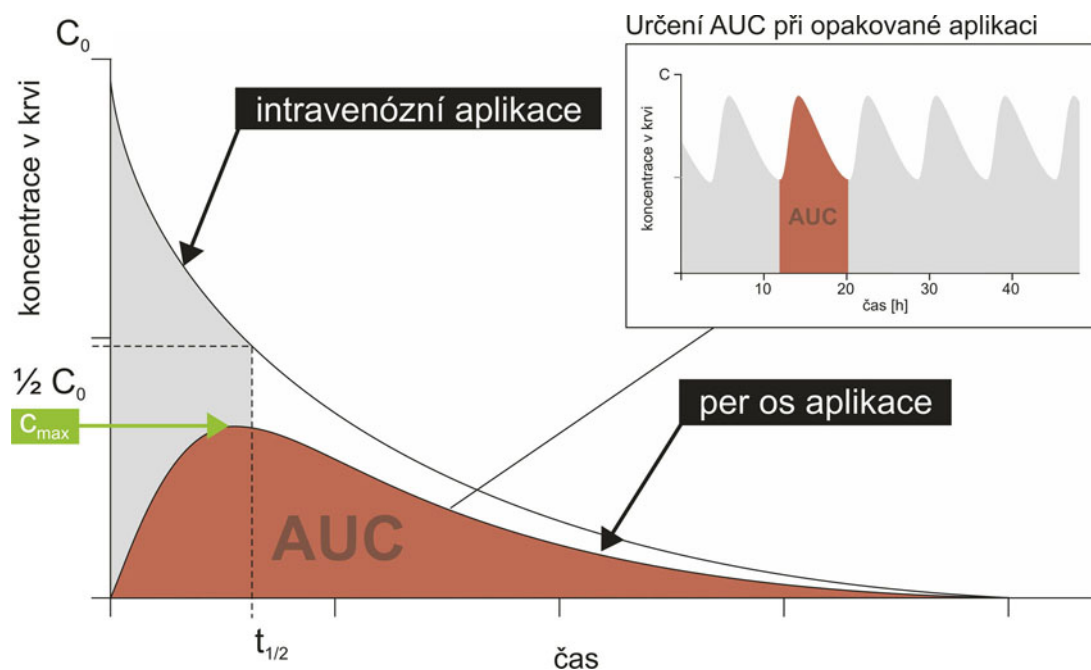
Plocha pod křivkou AUC

Myšlenou křivkou je průběh koncentrace farmaka v organismu. AUC je přímo úměrná celkovému množství měřené látky v organismu a lze pomocí ní vyjádřit biologickou dostupnost, tj. skutečné množství resorbované látky.[1]

Plocha je v průběhu času t vymezena:

- v časovém intervalu $t = < 0 : \infty >$ pro jednorázovou aplikaci
- v časovém intervalu mezi dvěma dávkami při opakované aplikaci

AUC je přímo závislá na množství látky, které bylo podáno do centrálního kompartmentu (oddílu), a to bez ohledu na rychlosti vstupu do kompartmentu. Srovnání AUC pro intravenózní (i.v.) a jiný typ aplikace má význam pro určení biologické dostupnosti (viz dále).[2]



Obr. 2.1: Znázornění některých farmakokinetických parametrů

Na obr. 2.1 je znázorněn rozdíl mezi průběhem koncentrací po i.v. a per os aplikaci. Definována je plocha pod křivkou AUC a plazmatický poločas eliminace $t_{1/2}$. Z obrázku je dále patrné, že pro určení plochy AUC při opakovaném podávání farmaka je její hodnota počítána pouze pro jeden cyklus probíhající mezi dvěma dávkami.

Absolutní biologická dostupnost F_{Abs}

Biologická dostupnost prezentuje podíl podané dávky farmaka, dostupný systémově. Protože pro i.v. aplikaci se farmakum dodává přímo do systémového oběhu, uvažujeme $F_{\text{Abs}} = 1$. Pro jiné typy aplikace proto můžeme F_{Abs} získat porovnáním právě plochou pod křivkami AUC podle[?]:

$$F_{\text{Abs}} = \frac{\text{AUC}}{\text{AUC}_{\text{i.v.}}} \cdot 100 \text{ [\%]}. \quad (2.5)$$

Absorpční konstanta k_a

Na úrovni modelování není důležité jakou cestou je látka do organismu aplikována, ale že při vstupu do kompartmentu dochází k překonávání biologických membrán absorpcí. Absorpční konstanta tedy vyjadřuje míru rychlosti absorpce a je charakterizována změnou koncentrace látky za jednotku času a ve většině případech je reprezentována kinetikou 1. řádu.[2]

Eliminační konstanta k_e

Eliminační konstanta k_e udává okamžitou rychlost eliminace vyjádřenou jako podíl farmaka eliminovaného za časovou jednotku [2]. Tento děj zahrnuje především procesy biotransformace a exkrece. V případě, že známe plazmatickou koncentraci ve dvou časových okamžicích, lze k_e vypočítat podle:

$$k_e = \frac{\ln \frac{C_1}{C_2}}{\delta t} = \frac{\ln C_1 - \ln C_2}{t_2 - t_1} \text{ [min}^{-1}\text{]}. \quad (2.6)$$

2.2 Kompartmentové modely pohybu léčiva v organismu

Kompartmentem je myšlena *idealizovaná diskrétní část prostoru*, tedy jakýsi hypotetický oddíl o určitém objemu a složení. Tento idealizovaný oddíl vykazuje tytéž vlastnosti pro všechny probíhající procesy ovlivňující sledované děje. Aby bylo možné s těmito oddíly dále pracovat, uvažuje se, že farmakum je v jeho rámci zcela *homogenně rozptýleno*.

Kompartmentové modely předpokládají, že lidský (nebo jakýkoliv) organismus lze zjednodušeně popsat jako *sérii různě navzájem propojených kompartmentů*. Pohyb částic mezi těmito kompartmenty je definován pomocí rychlostních konstant. Fyzikální vlastnosti kompartmentů lze vyjádřit některými základními parametry zmíněnými výše. Na základě těchto parametrů lze postihnout dynamické děje, které mezi jednotlivými oddíly probíhají a dále usuzovat na vlastnosti či strukturu v oblastech zájmu.[2]

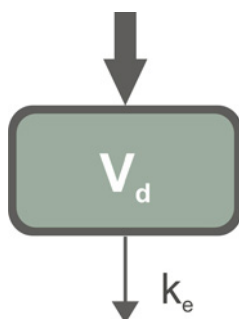
Většina dějů (difuze, filtrace, biotransformace) ovlivňující osud látky v organismu se řídí kinetikou 1. řádu, což značně usnadňuje jejich matematický popis.

Pro tyto děje platí, že:

- jejich rychlost je v každém okamžiku přímo úměrná koncentraci farmaka nebo koncentračnímu rozdílu
- rychlost děje můžeme popsat rychlostní konstantou 1. řádu nebo dobou poločasu děje

2.2.1 Jednokompartmentový model

Nejjednodušším farmakokinetickým modelem distribuce léčiva v organismu, který popisuje tělo jako jednoduchou homogenní jednotku je *jednokompartmentový model* (viz obr. 2.2).



Obr. 2.2: Blokové schéma jednokompartmentového farmakokinetického modelu

Model prezentuje situaci, kdy je změna koncentrace farmaka v plazmě následována rychlou změnou koncentrace v tkáních, i když nelze počítat, že by farmakum bylo v celém těle rozptýleno rovnoměrně. Aplikace léčiva do kompartmentu probíhá buď přímo intravenózně, nebo dochází k absorpci v případě jiné cesty (perorálně, intramuskulárně,...)[2].

Pokud dojde u jednokompartmentového modelu aplikaci látky intravenózně a jejímu okamžitému rozptýlení, je možno v tomto zjednodušeném případě změnu koncentrace v kompartmentu charakterizovat pouze eliminační konstantou k_e . Matematicky získáme diferenciální rovnici 1. řádu[3]:

$$\frac{\delta C}{\delta t} = -k_e \cdot C(t), \quad (2.7)$$

kde $C(t)$ je koncentrace sledovaného farmaka v čase t , k_e je eliminační konstanta. Řešení této rovnice dává tvar:

$$C(t) = C_0 \cdot e^{-k_e \cdot t}, \quad (2.8)$$

kde $C(0)$ je koncentrace v okamžiku podání dávky farmaka. Tato počáteční koncentrace bude odpovídat poměru dávky D a distribučního prostoru V_d :

$$C(0) = \frac{D}{V_d}. \quad (2.9)$$

Koncentrace farmaka po aplikaci i.v. či ukončení fáze absorpce pak vykazuje v čase monoexponenciální pokles, který je charakterizován největším úbytkem v čase t blížícímu se počátku aplikace a v poklesu úbytku v pozdějších fázích. Změna rychlosti úbytku je pak přímo úměrná velikosti eliminační konstanty.

V případě aplikace látky jiným způsobem než i.v. je nutné počítat s absorpční fází a započítat absorpční koeficient. Tím získáme rovnici demonstrující nejjednodušší případ průběhu hladiny v krvi, *Batemanovu funkci*[1]:

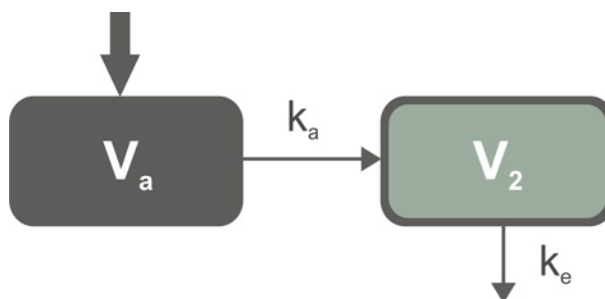
$$C(t) = a \cdot \frac{k_a}{k_a - k_e} \cdot (e^{-k_e \cdot t} - e^{-k_a \cdot t}). \quad (2.10)$$

Zde k_e , k_a jsou rychlostní koeficienty, a je konstanta, kterou lze pomocí biologické dostupnosti F , dávky D a distribučního objemu V_d definovat jako:

$$a = \frac{F \cdot D}{V_d}. \quad (2.11)$$

Absorpce látky do samotného kompartmentu a její eliminace jsou vyjádřeny dvěma protichůdnými *exponenciálními funkcemi*. Nelze ovšem říci, že by se jednalo o součet těchto procesů, jelikož eliminace je tím efektivnější, čím vyšší koncentrace látky v krvi se nachází. Souhra těchto funkcí se označuje právě jako Batemanova funkce.[1]

Jiným způsobem lze tento děj popsat, bude-li absorpce látky reprezentována nikoliv absorpční konstantou, ale samostatným kompartmentem. Tím vznikne *dvoukompartmentový model* s jednosměrnou komunikací (viz obr. 2.3). Dvoukompartmentové modely budou blíže popsány v následujícím textu.



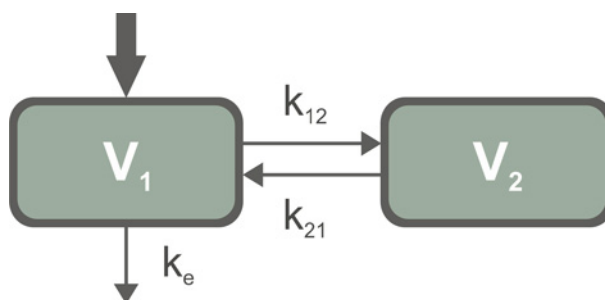
Obr. 2.3: Dvoukompartmentový model pro popis absorpce látky

2.2.2 Dvoukompartmentové modely

Vlastnosti jednotlivých částí organismu se mohou velmi různit především v souvislosti se zásobením krví, rozdělovacím koeficientem nebo permeabilitou kapilár. Tyto rozdíly však jednokompartmentový model zanedbává a může tím dojít k ovlivnění časového průběhu distribuce farmaka. Vícekompartmentové modely se tedy využívají pro podrobnější modelování, kdy je potřeba řešit vnitřní strukturu kompartmentů účastnících se na distribuci látky.[2]

Jako dostačující bude uvažován popis modelu složeného ze dvou kompartmentů - centrálního V_1 a periferního V_2 . Existuje navzájem několik kombinací v závislosti

na toku a na tom, ze kterého kompartmentu dochází k eliminaci farmaka. Pro cíl této práce bude dán V_1 jako objem krevního řečiště a periferní V_2 jako zkoumaná tkáň, do které farmakum z krve přestupuje (viz obr. 2.4).



Obr. 2.4: Blokové schéma dvoukompartimentového farmakokinetického modelu

Přísun a odsun farmaka je charakterizován absorpční konstantou k_a . V případě, kdy dochází k rychlé distribuční fázi po i.v. podání, lze k_a zanedbat a počítat s rychlým homogenním rozptýlením v celém kompartmentu. Eliminační konstantou k_e vyjadřuje proces vylučování farmaka z plazmy eliminačními mechanizmy (ledviny, játra..). Přestup látky mezi centrálním (V_1) a periferním (V_2) kompartmentem je dán rozdílem v koncentracích sledované látky a vyjádřen rychlostními konstantami k_{12} a k_{21} .

Tok mezi kompartmenty, je dán samovolným transportním dějem - *difúzí*. Tímto mechanismem je zajištěno, aby se systém, resp. dva vzájemně propojené oddíly, s různou koncentrací obsažené látky, dostaly zpět do termodynamické rovnováhy. Tento difúzní tok popisuje 1. *Fickův zákon*, který uvažuje, že hnací silou je změna koncentrace složky s polohou.[?]

Protože však některé jeho parametry v reálných podmínkách nemusíme znát, jsou vlastnosti toku shrnuty do konstant k_{12} , k_{21} . Matematicky lze model pak vyjádřit pomocí soustavy dvou diferenciálních rovnic:

$$\frac{\delta c_1}{\delta t} = -(k_e + k_{12} \cdot c_1(t) + k_{21} \cdot c_2(t)), \quad (2.12)$$

$$\frac{\delta c_2}{\delta t} = k_{12} \cdot c_1(t) - k_{21} \cdot c_2(t), \quad (2.13)$$

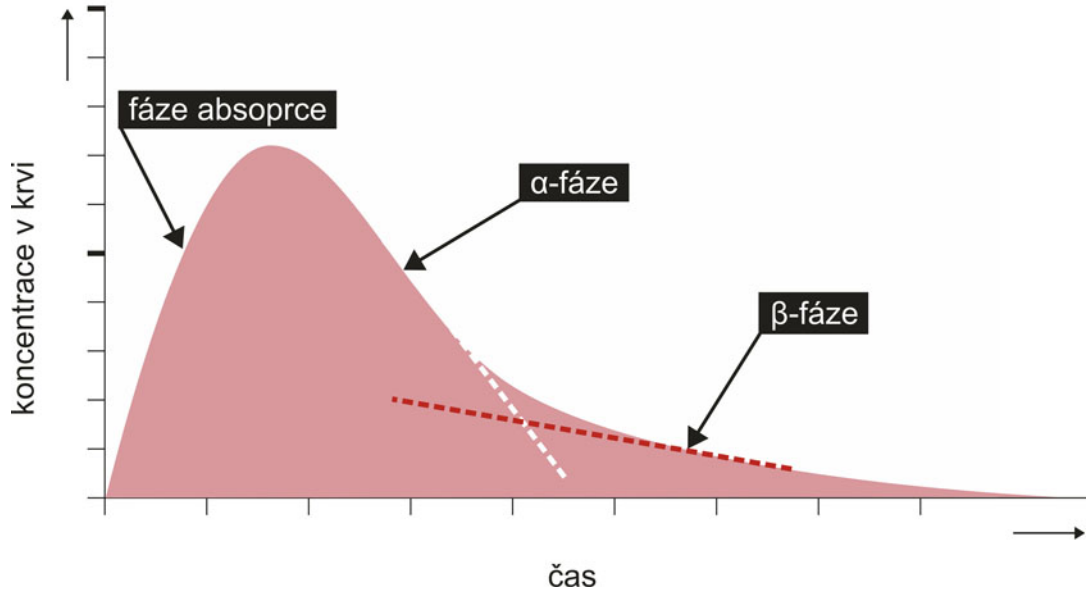
kde $c_1(t)$ a $c_2(t)$ jsou koncentrace v jednotlivých kompartmentech v čase t .

Po ukončení absorpce kdy dochází k poklesu plazmatické koncentrace látky bude výstupem v závislosti na čase bi-exponenciální křivka (viz obr 2.5).

Rychlejší fáze poklesu - α -fáze - znázorňuje distribuci farmaka z centrálního kompartmentu V_1 do tkání. Druhá, pomalejší fáze - β -fáze, eliminaci po ukončení distribuce. Poté je pokles koncentrace mono-exponenciální.[2]

Popis pomocí konvoluce

Jiný způsob vyjádření procesů v kompartmentové analýze umožňuje stanovení funkčních závislostí a využití konvoluce. Konvoluce je matematický operátor, který kombinuje dvě funkce a je definována následujícím způsobem:



Obr. 2.5: Průběh změny koncentrace v čase u dvoukompartmentového modelu

$$(f * g) \cdot (x) = \int_{-\infty}^{\infty} f(\alpha) \cdot g(x - \alpha) d\alpha, \quad (2.14)$$

kde funkci $g(x)$ značí konvoluční jádro. Hodnota konvoluce funkce f s jádrem g v bodě x je integrál ze součinu funkce f s otočenou funkcí konvolučního jádra[8].

Popis diferenciálními rovnicemi lze pak snadno nahradit funkcemi popisující zkoumaný systém. To je výhodné zejména pro stanovení parametrů těchto funkcí a snažší digitální zpracování.

Pro tento případ nebude krevní řečiště uvažováno jako samostatný kompartment, ale bude definován krevním tokem (případně funkcí), který prostupuje sérií po sobě jdoucích kompartmentů (viz obr 2.6), do kterých farmaka z krve difundují. Tyto děje jsou opět pro každý kompartment popsány příslušnou funkcí.

Bude-li uvažován různý počet kompartmentů modelu, označený $i = 1, 2, \dots, n$, je každý kompartment definován distribučním objemem V_i a přenosovou funkcí $h_i(t)$. [11]

Přenosová funkce $h_i(t)$ ovlivňuje z matematického hlediska průchod látek kompartmentem. Představuje v něm koncentraci látky v čase t jako odezvu na aplikovaný ideální bolus látky (jedná se o definované množství, rozptýlené v čase $t = 0$ v celém svém množství – lze ho tedy považovat za jednotkový impuls)[16]. $h_i(t)$ je definována jako:

$$h(t) = \frac{1}{\tau} \cdot e^{-\frac{t}{\tau}}, \quad (2.15)$$

kde τ je časová konstanta vyjadřující přechod látky do kompartmentu a je definována jako poměr distribučního objemu V_i a objemového průtoku krve F :

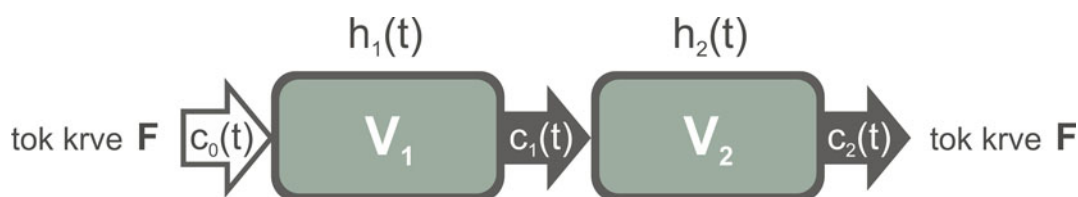
$$\tau = \frac{V_i}{F}. \quad (2.16)$$

Bude-li tok látky do každého kompartmentu dán obecně jako vstupní funkce $c_{i-1}(t)$, lze popsat koncentraci, resp. její vývoj v čase v kompartmentu, jako konvoluci výstupní funkce $c_i(t)$. Protože tok látky oddílem je ovlivňován jeho vlastnostmi definovanými pomocí $h_i(t)$, získáme výstupní funkci jako konvoluci $c_{i-1}(t)$ a $h(t)$ [11]:

$$c_i(t) = c_{i-1}(t) * h(t), \quad (2.17)$$

kde $*$ značí konvoluci funkcí.

U nejjednodušších modelů je většinou přenosová funkce definována pouze jedním parametrem - časovou konstantou k . Pokud je však cílem dosáhnout obzvláště v perfuzním zobrazení co nejlepší aproximace reálných dat, je třeba využít složitější modely s více parametry.



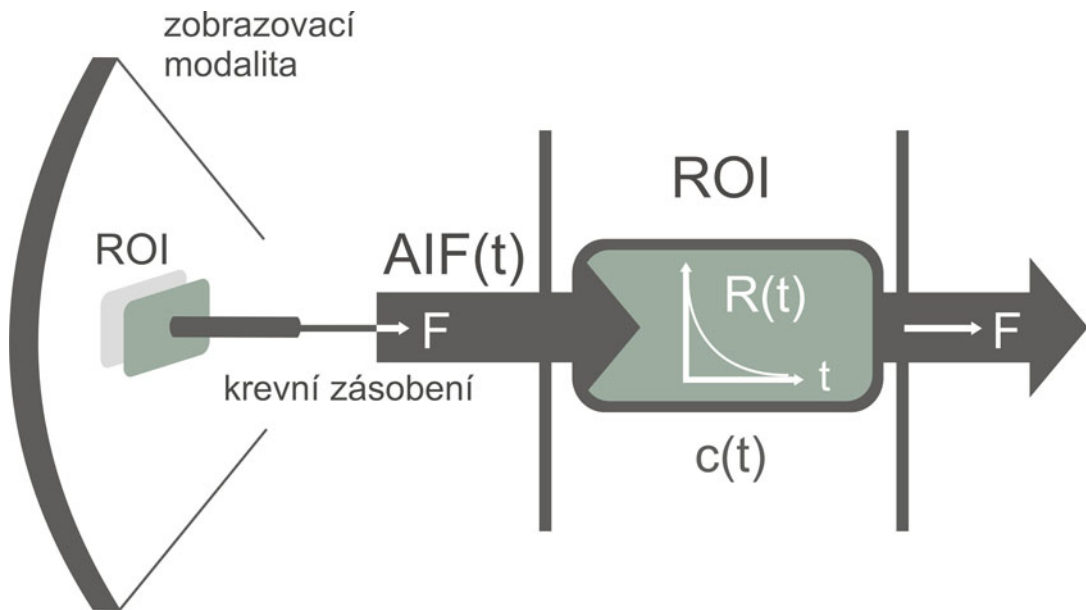
Obr. 2.6: Krevní oběh probíhajícího jednotlivými bloky modelu

2.3 Modely v perfuzním zobrazení

Cílem perfuzních zobrazovacích metod je vytvoření obrazu lokálního prokrvení tkáně (perfuze). Perfuzní analýza a některé s ní související parametry umožňují získat cenné informace o funkci tkání, jejich metabolismu, či změnách průtoku krve. Toho lze s výhodou využít pro diagnostiku a sledování vývoje nádorových onemocnění a obstrukcí v krevním řečišti. V současné době pak spíše z výzkumného hlediska i ke sledování změn v mozkové tkáni u některých onemocnění.

Oproti klasickým zobrazovacím metodám, které zobrazují především strukturu, však vyžaduje tato diagnostika odlišný přístup, neboť prokrvení tkání závisí na dynamických dějích, které je nutné přesně kvantifikovat. Principem je intravenózní aplikace bolu kontrastní látky, která pro danou zobrazovací modalitu způsobí změny v intenzitě měřeného signálu. Tyto změny intenzity jsou pak převedeny na změnu koncentrace v čase.[13]

Pro cílovou tkáň je na počátku měření stanoven tzv. *ROI* (region of interest). *ROI* vyjadřuje oblast zájmu (a vlastně tedy samostatný kompartment), pro kterou budou perfuzní parametry získávány (viz obr. 2.7).[16]



Obr. 2.7: Znázornění funkčních vztahu pro definovanou oblast zájmu

Měřená intenzita signálu pro kontrastní látku je vhodným algoritmem převedena na průběh koncentrace v čase. Tento průběh se nazývá *AIF* (arterial input function) a vyjadřuje vstupní funkci měřeného elementu do zájmové oblasti *ROI*. Její přesné stanovení je základním předpokladem pro správný výpočet kvantitativních parametrů.

Protože však je obtížné měřit nízký signál koncentrací v periferních oblastech, určuje se *AIF* měřením ve velkých vyživujících artériích. Průchodem kontrastní látky z oblasti artérie do *ROI* však dochází ke zpoždění a další disperzi látky, která skutečnou *AIF* významně ovlivňuje. Pro získání pravé *AIF* je tedy výhodné již v této úvodní fázi využít modelů pro vyjádření transportní funkce krve a tyto odchylky korigovat podle výpočtu (2.16). Možné je využití více typů funkcí, jako nejjednodušší a dostačující lze však pro $h(t)$ krevního transportu vzorec (2.14).

Vyjdeme-li opět z výpočtu (2.16), lze získat obecný vztah pro výpočet koncentrace ve sledované tkáni pomocí[10]:

$$c(t) = F \cdot c_a(t) * R(t), \quad (2.18)$$

kde $*$ značí konvoluci, F je objemový průtok krve, $R(t)$ kapilární reziduální funkce, $c_a(t)$ koncentrace v měřené artérii, vycházející z *AIF*, a $c(t)$ průběh koncentrace ve tkáni.

Reziduální funkce $R(t)$ je analogií pro přenosovou funkci $h(t)$. Jedná se tedy o průběh koncentrace v čase po aplikaci ideálního bolu látky, vyjádřenou pro *ROI*. Průběh $R(t)$ tedy dává základní informace o tkáni, kterou kontrastní látka prochází. Její odhad lze získat na základě dekonvoluce funkce (2.17).[10]

Protože je $R(t)$ uvažována jako odpověď na jednotkový impulz, lze uvažovat, že hodnota v $R(0)$ je určena úplným množstvím měřené látky a je tedy přímo úměrná

objemovému průtoku F tkání [18]. Pomocí známé $R(t)$ a F lze určit distribuční objem v regionu jako:

$$V_{\text{ROI}} = \frac{\int_0^{\infty} c(t) dt}{\int_0^{\infty} c_a(t) dt}, \quad (2.19)$$

kde V_{ROI} značí distribuční objem, $c(t)$ koncentraci látky v tkáni a $c_a(t)$ měřenou koncentraci určenou z *AIF*.

Jedním z dalších důležitých parametrů je doba průchodu farmaka danou tkání, která může být významně ovlivněna až už změnou tkání samotné, tak omezením průtoku krve. Z anglického *mean transit time* se označuje jako *MTT*. Její hodnotu lze získat dvěma způsoby:

- Bude-li $R(t)$ aproximována modelem exponenciální funkce, pak lze pomocí *MTT* vyjádřit jako:

$$R(t) = e^{-\frac{t}{\text{MTT}}}. \quad (2.20)$$

- Při známých hodnotách průtoku krve a distribučního objemu tkáně V_{ROI} v jednodušším případě:

$$V_{\text{ROI}} = \frac{\int_0^{\infty} c(t) dt}{\int_0^{\infty} c_a(t) dt}. \quad (2.21)$$

Z uvedených vztahů je patrné, že stanovení perfuzních parametrů je řetězcem na sebe navazujících kroků. Ty ve větší či menší míře využívají aproximace za pomoci funkcí. Aproximace je nutná zejména kvůli složitosti průběhu reálně měřených dat, které jsou významně ovlivněny fyziologickými procesy, měřicí technikou i přídatným šumem. V rámci této práce byly vybrány tři funkce, se kterými je možné se v dnešní době setkat při popisu perfuzních křivek:

- *Gamma funkce*
- *Log-normální funkce*
- *Local density random walk (LDRW) model*.

2.3.1 Gamma funkce

Gamma funkce je jedna z častých voleb při modelování transportních dějů v krevním oběhu. Základ gamma funkce vychází ze zobecnění faktoriálu pro čísla z oboru komplexních i reálných čísel. Popisuje gamma pravděpodobnostního rozložení veličiny s typicky asymetrickým tvarem. Funkce je nejčastěji určena dvěma parametry nazývanými obecně jako *scale parametr* a *shape parametr*. Pokud je scale parametr symbolicky vyjádřen jako λ a scale parametr jako γ , udává λ praktický rozsah veličiny a γ ovlivňuje tvar křivky.

Oba parametry pak souvisí se střední hodnotou μ a rozptylem σ^2 náhodné veličiny a jsou dány jako:

$$\mu = \frac{\gamma}{\lambda}, \quad (2.22)$$

$$\sigma^2 = \frac{\gamma}{\lambda^2}. \quad (2.23)$$

Vyjádřením gamma funkce s takto definovanými parametry γ a λ je rovnice ve tvaru:

$$f(m, \lambda, \gamma, t) = m \cdot \frac{\lambda^\gamma \cdot t^{\gamma-1} \cdot e^{-\lambda \cdot t}}{\Gamma(\gamma)}, \quad (2.24)$$

kde t je parametr času a m lineární násobitel. $\Gamma(\gamma)$ označuje úplnou gamma funkci, pro kterou dále platí vztah:

$$\Gamma(1) = \left(\frac{1}{2}\right), \quad (2.25)$$

$$\Gamma(\gamma + 1) = \gamma \cdot \Gamma(\gamma). \quad (2.26)$$

Z uvedených vzorců (2.25) a (2.26) vyplývá, že funkce skutečně vychází z určení faktoriálu parametru γ :

$$\Gamma(\gamma) = (\gamma - 1)!, \quad (2.27)$$

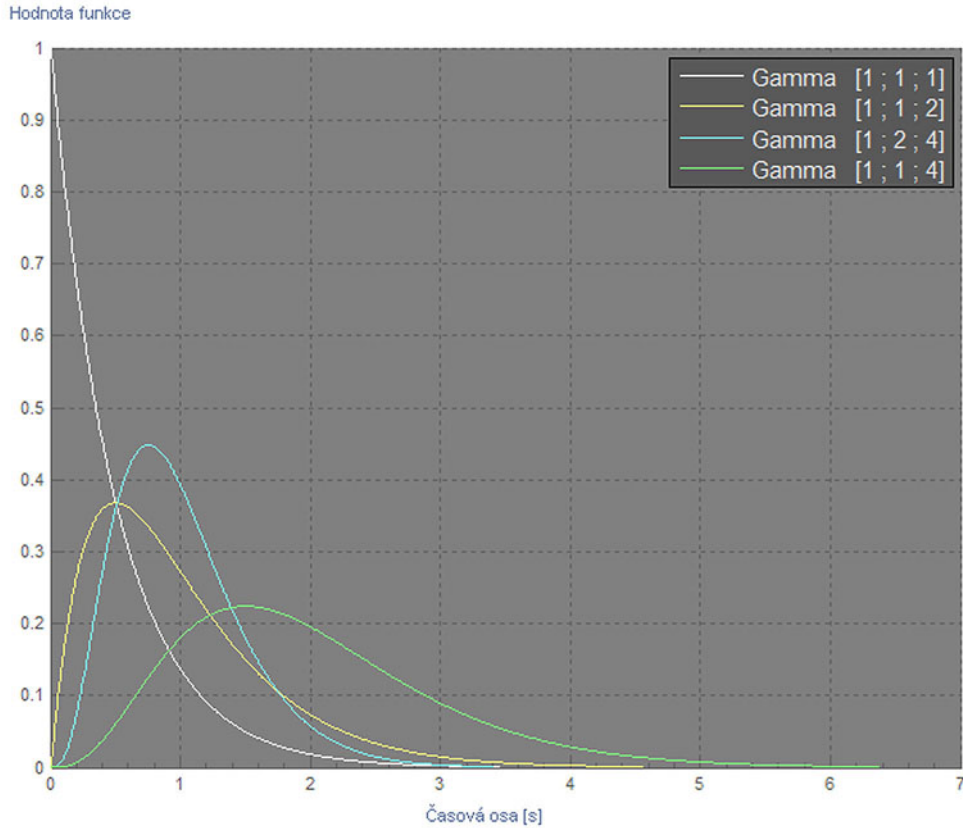
kde pro parametr $\gamma \in \mathbb{N}$ dochází ke zjednodušení na tzv. Erlangovo pravděpodobnostní rozložení. Pro $\gamma \in \mathbb{N}$ i $\gamma \in \mathbb{R}$ je nutné zajistit podmínku $\gamma > 0$, aby nedošlo k divergenci funkce.

Je-li gamma funkce dána do souvislosti s popisem farmakokinetických dějů, jsou γ a λ parametry distribuce zkoumané látky, resp. změny její koncentrace c v čase v jednotkách [min^{-1}]. Strmost jejího nástupu je významně ovlivňována disperzí látky v krevním řečišti.

Typické průběhy gamma funkce ukazuje obrázek (2.8) pro následující vybrané hodnoty parametrů m , λ a γ :

- $m = 1, \lambda = 1, \gamma = 1$
- $m = 1, \lambda = 1, \gamma = 2$
- $m = 1, \lambda = 2, \gamma = 4$
- $m = 1, \lambda = 1, \gamma = 4$

Z obrázku (2.8) je patrné, že vzájemný poměr obou parametrů působí na asymetrii křivky. Blíží-li se $\frac{\lambda}{\gamma} \approx 1$, je maximum položeno blíže k počátečním hodnotám, jak lze zjistit i ze vzorce (2.22). Větší vzájemný poměr způsobuje snížení strmosti vzestupu i poklesu křivky a podstatně se snižuje její asymetrie. Zvláštním případem je průběh v případě, kdy $\lambda = 1, \gamma = 1$. Dochází ke zjednodušení funkce (2.24) a průběhem je exponenciální pokles z maximální hodnoty. Tato situace by odpovídala aplikaci ideálního bolu látky s okamžitým rozptýlením.



Obr. 2.8: Typické průběhy gamma funkce pro zadané parametry

2.3.2 Log-normální funkce

Log-normální funkce je matematickým popisem log-normálního rozložení náhodné veličiny X . Náhodná veličina X má log-normální rozložení, jestliže logaritmus náhodné veličiny X má rozložení normální. X může nabývat pouze kladných hodnot. K určení log-normální funkce je třeba dvou parametrů σ a μ . Parametr σ určuje směrodatnou odchylku a μ střední hodnotu X normálního rozložení pro případ logaritmu X . Jejich souvislost s log-normálním rozložením je pak dána vztahy:

$$E(X) = e^{\mu + \frac{1}{2}\sigma^2}, \quad (2.28)$$

$$\sigma_2^2(X) = e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2}, \quad (2.29)$$

kde $E(X)$ nabývá významu střední hodnoty a $\sigma_2^2(X)$ rozptylu log-normálního rozložení.

Výsledný obecný matematický výraz pro log-normální funkce pak může být zapsán jako:

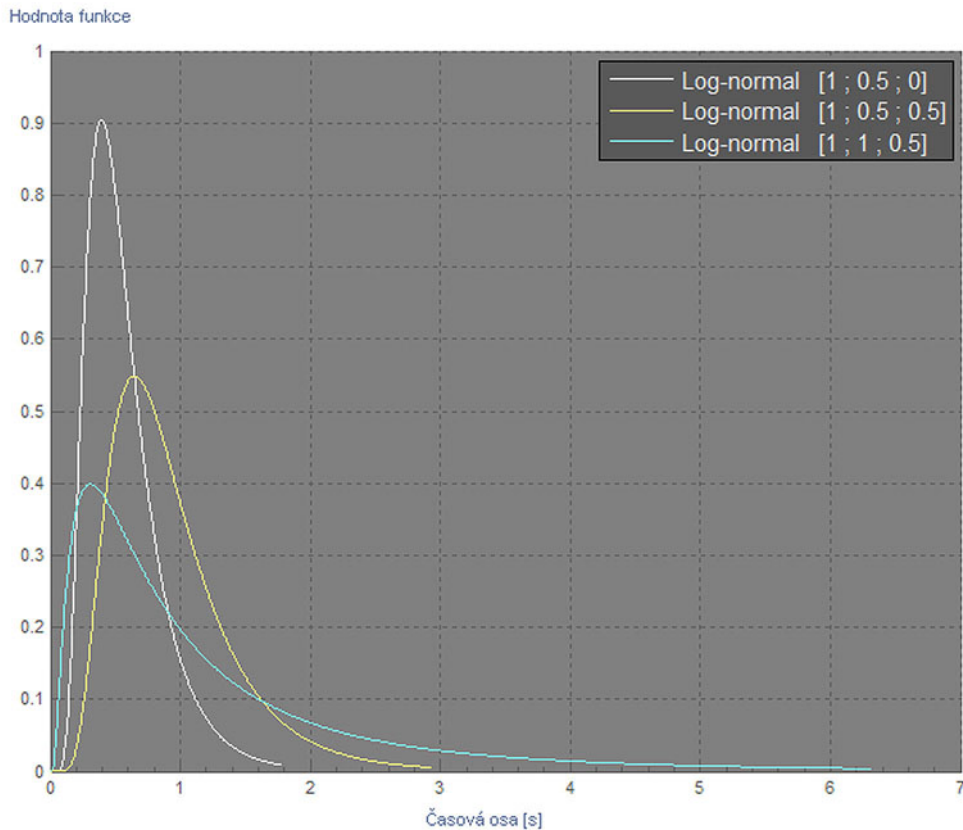
$$f_{(m,\mu,\sigma,t)} = m \cdot \frac{1}{t \cdot \sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{(\ln(t)-\mu)^2}{2 \cdot \sigma^2}}, \quad (2.30)$$

kde m je lineární násobitel a t určuje parametr času.

Průběh funkce vykazuje větší asymetrii křivky než je tomu u gamma funkce. Hodnoty maxima jsou posunuty blíže k počátku, proto lze log-normální funkcí dobře aproximovat data s prudkým nárůstem hodnot a pozvolným poklesem. To může být výhodné pro zachycení průběhu nízkých koncentrací či intenzit signálu.[9]

Pro ukázkou průběhů log-normální funkce na obrázku (2.9) byly zvoleny následující hodnoty parametrů m , σ a μ :

- $m = 1, \sigma = 0.5, \mu = 0$
- $m = 1, \sigma = 0.5, \mu = 0.5$
- $m = 1, \sigma = 1, \mu = 0.5$

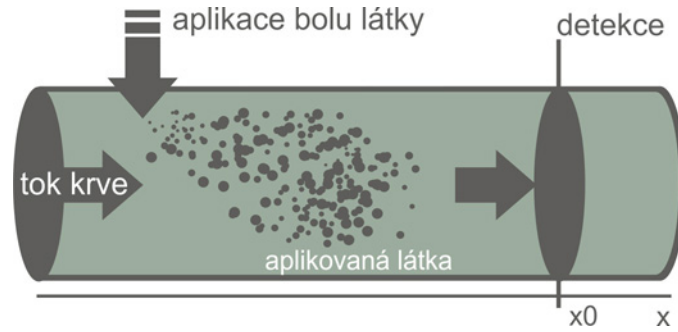


Obr. 2.9: Průběhy log-normálního modelu pro zadané parametry

Je patrné, že parametr σ v souvislosti s rozptylem má vliv především na tvar křivky. Vyšší hodnoty vedou ke zpomalení dosažení maxima a podstatnému prodloužení poklesu hodnot. Změna parametru μ pak určuje dosažené maximum, jak lze očekávat pro změnu střední hodnoty rozložení náhodné veličiny X .

2.3.3 Local density random walk model

Local density random walk (LDRW) model vychází z experimentálního popisu průchodu sledované látky konečně dlouhou trubicí naplněnou tekutinou (viz obrázek 2.10). Předpokladem pro zavedení modelu je konstantní proudění kapaliny a rychlá aplikace bolu látky. Postupné rozředování (diluce) látky je ovlivněna jednak prouděním a dále Brownovým pohybem, vyvolaným interakcí částic kapaliny a aplikované látky.[4]



Obr. 2.10: Schéma experimentálního LDRW modelu

Poloha jednotlivých částic v prostoru a čase je dána náhodným procesem. Rozložení náhodné veličiny tedy může být popsáno statistickým parametrem střední hodnoty μ a rozptylu σ . To vede k následujícímu popisu pomocí pravděpodobnostní funkce v časovém intervalu nT [4]:

$$f(x, nT) = \frac{1}{\sqrt{2\pi n\sigma^2}} \cdot e^{-\frac{(x-n\mu)^2}{2n\sigma^2}}, \quad (2.31)$$

kde platí, že x je prostorová souřadnice.

Jestli-že pro nekonečně malé časové přírůstky bude $nT = t$, a dále $\frac{\sigma^2}{T} = \alpha$ a $\frac{\mu}{T} = u$, vede funkce na výraz popisující Brownův pohyb částic (Wienerův proces):

$$f(x, t) = \frac{1}{\sqrt{2\pi t\alpha}} \cdot e^{-\frac{(x-tu)^2}{2t\alpha}}. \quad (2.32)$$

Rovnici (2.1) lze odvodit pro určení koncentrace aplikovaného množství látky D , rozptýlené ve známém objemu V_d . Pro koncentraci látky v závislosti na čase t , souřadnici x a jejím pohybu v proudící kapalině podle (2.30) lze odvodit, že:

$$C(x, t) = \frac{D}{V_d} \cdot f(x, t), \quad (2.33)$$

kde V_d reprezentuje sledovaný úsek trubice s proudící kapalinou.

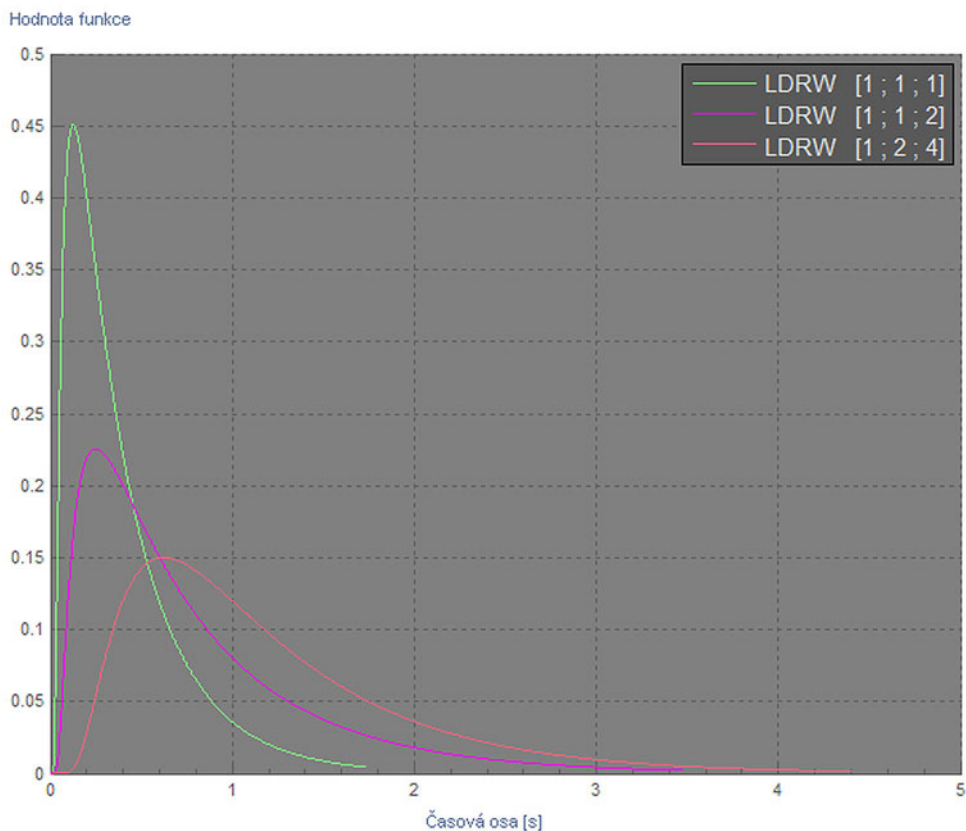
Podle [4] lze dalšími úpravami docílit výsledného tvaru funkce reprezentující LDRW model pro hodnoty koncentrace ve vzdálenosti x_0 :

$$f(m, \lambda, \mu, t) = m \cdot \frac{1}{\mu\Phi} \cdot e^\lambda \cdot \sqrt{\frac{\lambda\mu}{2\pi t}} \cdot e^{-\frac{\lambda}{2} \cdot \left(\frac{t}{\mu} + \frac{\mu}{t}\right)}, \quad (2.34)$$

pro kterou platí, že tok kapaliny $\Phi = uV_d$, $m = D$ a $\lambda = \frac{\mu u^2}{\alpha} = \frac{\mu\Phi^2}{\alpha V_d^2}$. Parametr λ pak souvisí s asymetrií sklonu křivky.[4]

Následné parametry m , λ a μ byly využity pro demonstraci průběhu křivky (viz obrázek 3.10) LDRW modelu:

- $m = 1, \lambda = 1, \mu = 1$
- $m = 1, \lambda = 1, \mu = 2$
- $m = 1, \lambda = 2, \mu = 4$



Obr. 2.11: Průběhy LDRW modelu pro zadané parametry

Maximální koncentrace látky je dosaženo pro $t = \mu \cdot 2\lambda^{-1} (\sqrt{1 + 4\lambda^2} - 1) < \mu$. Z křivek na obrázku (2.11) je opět patrná výrazná asymetrie k počátečním hodnotám a průběh je velmi podobný log-normální funkci. Zvyšující se parametr λ způsobuje větší symetrii průběhu a posunuje dosažené maximum směrem od počátečních hodnot. To naznačuje, že λ úzce souvisí s časem difúze částic v prostředí a její hodnota bude ovlivněna dobou přítoku sledovaného farmaka do místa detekce.[4]

3 REALIZACE PROGRAMOVÉ ČÁSTI

Cílem práce bylo vytvořit soubor funkcí, které realizují simulaci zvolených modelů. Uživatel by měl mít možnost volby modelu a jeho parametrů. V další části bylo úkolem využít tyto modely pro proložení experimentálně naměřených dat. Proložení by mělo být realizováno pomocí vhodné optimalizační metody. Na výstupu je získána optimalizovaná křivka dobře aproximující data a její parametry. Ovládací prvky a výstupy jsou zasazeny do grafického rozhraní. To umožňuje lepší intuitivní ovládání a uživatelskou přístupnost.

3.1 Popis programovacího prostředí

Pro vytváření skriptů i grafického rozhraní byl použit software od firmy Math-Works – *Matlab* ve verzi R2009b. Matlab je programovací prostředí a skriptovací jazyk určený především pro vědecko-technické výpočty. Uživatelům umožňuje práci v širokém spektru oborů od numerických výpočtů, vyvíjení algoritmů, zpracování a analýzy dat, modelování a počítačovou simulaci, až po výslednou prezentaci dat v několika rozměrech. Je tedy zřejmé, že Matlab poskytuje uživatelům širokou škálu nástrojů. Jeho předností oproti jiným programovacím jazykům jsou takzvané *toolboxy*. Jedná se o speciální kolekci předprogramovaných funkcí, které jsou dostupné odděleně. Jednotlivé toolboxy obsahují funkce, které obsluhují často řešené problémy v oboru, pro který je toolbox vytvořen. To značně ulehčuje práci při psaní skriptů, protože není nutné vytvářet vlastní funkce pro zpracování dat od nejnižší úrovně. Výhodou je také vlastní implementované prostředí pro objektové programování a vytváření grafických aplikací – graphic user interface (GUI). Prostředí GUI je využito při tvorbě uživatelské aplikace v této práci.

Základ zpracování dat v Matlabu spočívá ve vytváření datových polí. Matlab definuje své proměnné i zpracovávaná data jako vektory či matice hodnot. Toto přináší snadné provádění numerických výpočtů i zobrazování dat.

Výchozím formátem programu Matlab je skriptový soubor *m-file* s koncovkou **.m*. Do toho souboru se ukládají programové kódy, případně je zde možné definovat samostatně pracující funkce. Grafické rozhraní je zpracováno do souboru *figure* s koncovkou **.fig*. Zde je možné vytvořit grafický návrh v implementovaném editoru. Stejně tak je možné tento návrh provést pomocí příkázů objektového programování v *m-file* souboru.

Optimization ToolboxTM 5

Optimization toolbox je rozšířením výpočetního prostředí Matlab. Obsahuje často využívané algoritmy pro optimalizační úlohy od metody nejmenších čtverců až po kvadratické programování. Umožňuje široký rozsah nastavení pro jednotlivé metody a je tedy vhodnou volbou pro snadné zavedení optimalizace. Vybrané funkce Optimization toolboxu budou použity v rámci této práce k prokládání dat modelem.

3.2 Simulace farmakokinetických modelů

Simulací rozumíme interpretaci chování určitého systému za předem stanovených podmínek. Myšleným systémem v této práci je část krevního řečiště po aplikaci určitého množství látky. Sledovanou vlastností je průběh koncentrace této látky v čase. Protože simulace bude probíhat za pomoci výpočetní techniky, je nutné znát matematický popis sledované vlastnosti pro určení jejího průběhu. K tomuto účelu budou sloužit modely uvedené v kapitole 2.3:

- Gamma funkce
- Log-normální funkce
- LDRW model

U všech modelů jsou vstupní podmínky určeny jejich jednotlivými parametry. Aby měla simulace smysl, musí být základní funkcí tvořeného skriptu možnost vlastní volby jednotlivých parametrů. Tím je možné sledovat odezvu modelovaného systému za různých podmínek.

Skript dále musí obsahovat volbu vybraných modelů. Aby bylo možné jejich porovnání pro různé vstupní parametry, je nutné zajistit uložení a vizualizaci všech získaných dat. Data by měla být vhodným způsobem prezentována uživateli. To se týká zejména jejich popisu pro lepší identifikaci a kontroly nad obsahem datové struktury. Zahrnutí těchto funkcí pod kontrolu pomocí grafického rozhraní, by mělo ve výsledku přinést uživateli intuitivní a snadné ovládání celého skriptu.

3.2.1 Struktura programu

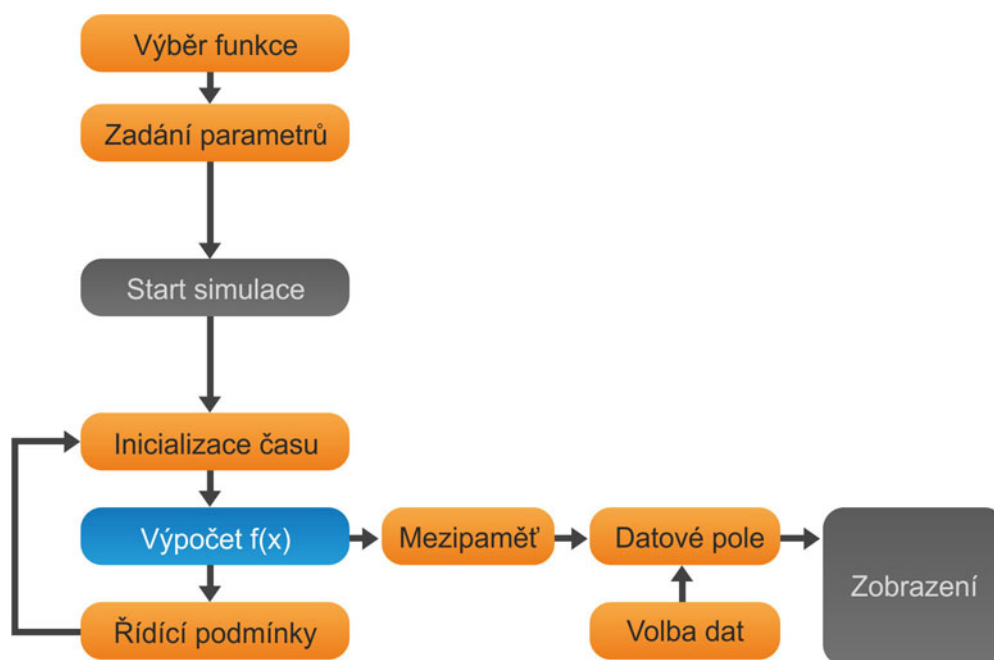
Samotné programové řešení je rozděleno do několika bloků. Hlavní skript je umístěn v souboru *Farmakokin_modely.m*. Jedná se o část, která je spjata s grafickým rozhraním GUI a umožňuje využít objektového programování. Obsahuje všechny základní funkce řídící proces simulace. Mezi ty patří získávání informací o simulovaném modelu a jeho parametrech, řízení samotného procesu simulace, ovládání datové struktury a vizualizace dat. Funkce jsou přiřazeny k jednotlivým grafickým prvkům GUI. To se děje pomocí generované zpětné vazby, která odkazuje na daný prvek a při interakci uživatele dojde k inicializaci podřízené funkce.

Další bloky jsou určeny samotným modelům. Každý model má vlastní m-file, který obsahuje jeho matematické vyjádření. Následuje výčet souborů:

- *gammafunkce.m*
- *lognormal.m*
- *ldr.w.m*

Skripty s modely jsou definovány jako *funkce*. Mají popsánu hlavičku s názvem funkce, která odpovídá názvu souboru. Funkce je pod tímto názvem volána z hlavního skriptu a předává z něj vstupní proměnné. Zpětně vrací výpočet funkční hodnoty pro daný model a zadané parametry.

Vztahy mezi jednotlivými funkčními strukturami ukazuje obrázek (3.1). Jedná se o základní popis nejdůležitějších částí programu a znázornění jejich vzájemné provázanosti.



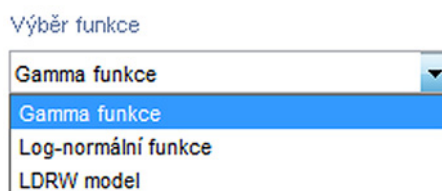
Obr. 3.1: Základní schéma skriptu pro simulaci modelů

3.2.2 Funkční a grafické řešení

V této kapitole jsou popsány části skriptu a jejich návaznost na GUI. Přibližuje význam jednotlivých funkcí a použitých postupů při řešení problému. Součástí jsou také ukázky grafického řešení uživatelského prostředí.

Volba funkce

Volba výchozí funkce, pro kterou bude prováděna simulace, je realizována v GUI pomocí prvku *popupmenu*. Jedná se o standardní rolovací menu, do kterého byly položky nastaveny implicitně, viz obrázek (3.2).



Obr. 3.2: Grafický prvek pro volbu funkce

Volba v menu vrátí hodnotu indexu zvoleného řetězce. Pomocí rozhodovacího pravidla se potom do proměnné *rovnice* vygeneruje řetězec s názvem volané funkce.

Následuje ukázka části zdrojového kódu pro volbu gamma funkce:

```
vyber = (get(handles.Menu_vyberfce, 'Value'));  
if vyber == 1  
    rovnice = genvarname('gammafunkce', who);
```

Proměnná *rovnice* pak slouží při spuštění simulace přímo k vyvolání dané funkce.

Volba parametrů

Pro zadání parametrů funkce slouží prvky typu *edittext*. Jedná se o textová pole, která umožňují vlastní editaci uživatelem. Pro každý parametr je vytvořen samostatný prvek, obecně pojmenovaný *m*, *p1* a *p2*. Textová pole předávají programu proměnné typu *string*, z toho důvodu je zajištěn převod na číselné hodnoty pomocí funkce *str2double*.

Vhodně zadané parametry jsou nutné pro správný průběh simulace. Pro každý parametr jsou proto před spuštěním simulace kontrolovány omezující podmínky. Ty jsou pro LDRW a gamma model dány jako:

- $m, p1, p2 \neq \emptyset$
- $m, p1, p2 \in (0; \infty)$

Pro log-normální model:

- $m, p1, p2 \neq \emptyset$
- $m, p1 \in (0; \infty)$
- $p2 \in \langle 0; \infty \rangle$

Časovač

Časovačem je myšlen jednoduchý kód, založený na postupném rovnoměrném zvyšování hodnoty. Je dán vektorem *t* o jednom prvku, s inicializační hodnotou $t = 0$. Jeho význam spočívá v generování posloupnosti čísel, která reprezentuje diskretní časové okamžiky v průběhu simulace. K inkrementu hodnoty dochází vždy v jednom cyklu po ukončení výpočtu funkční hodnoty v předchozím časovém okamžiku. Hodnota po ukončení cyklu je stanovena na základě výpočtu $t = t + 0.01$. Tím dochází k výpočtu pro dlouhou posloupnost čísel a zaručuje jemný rastr při vykreslování křivek.

Datové pole

Datovým polem je struktura, která sdružuje určitý počet prvků a vytváří tak úložiště informací. Matlab nabízí několik typů datových polí, které se liší typem zapisovaných dat. Pro potřebu skriptu pro simulaci byla jako nejvhodnější vybrána struktura typu *cellarray*. Tento typ pole umožňuje ukládat číselné vektory různých velikostí, textové řetězce i vnořená pole s dalšími prvky. Datové pole je základním prvkem, který umožňuje uživateli vícenásobné provádění simulace, přehledný popis uložených dat a možnost jejich výběru. Jeho inicializace probíhá ihned při spuštění skriptu pomocí příkazu $Z = \{\}$ a je zpřístupněna všem podřazeným funkcím jako *globální proměnná Z*.

Pole *Z* s uloženými daty vykazuje následující strukturu:

$$Z(j, i) = \{\text{Polozka1}, Y$$

Polozka1 značí proměnnou ve formě textového řetězce, do které jsou ukládány hodnoty zvolené uživatelem. Proměnná má popisný charakter a slouží ke generování výpisu datové struktury. Hodnoty jsou proto ukládány ve formě textového řetězce v pořadí *název funkce, m, p1, p2*. Proměnná *Y* je řádkový vektor hodnot. Obsahuje funkční hodnoty, které byly vráceny simulovaným modelem. K jednotlivým prvkům se přistupuje pomocí indexování. Index *j* patří pozici řádku a vrací všechny data uložená po uskutečněné simulaci. Index *i* umožňuje výpis dat z jednotlivých sloupců pole a oddělit tak textový řetězec od vypočítaných dat.

Výpočet funkčních hodnot a řídicí podmínky

Výpočet funkčních hodnot slouží k získání průběhu simulované funkce. V prvním kroku dochází k inicializaci vstupních proměnných. Těmi jsou, jak bylo popsáno v předchozím textu, parametry *m, p1, p2* a hodnota časovače *t*. Výstupní hodnoty jsou počítány v cyklu pro každý časový interval. Následně jsou hodnoty postupně ukládány do řádkového vektoru *Y*, jak ukazuje část zdrojového kódu:

```
while i < n
    Y(i)= eval(['rovnice '(m,p1,p2,t)']);
    if i > 2 && (Y(i)-Y(i-1))<0 && Y(i)<0.01*max(Y)
        n=n;
    else
        n=n+1;
    end
    t=t+0.01;
    i=i+1;
end
```

Samotný výpočet neprobíhá v hlavním skriptu, ale je řešen voláním externí funkce. Ta je definována proměnnou *rovnice*. Její obsah je ovlivněn volbou v uživatelském rozhraní. Volání externí funkce se opakuje v každém cyklu až do splnění řídicích podmínek.

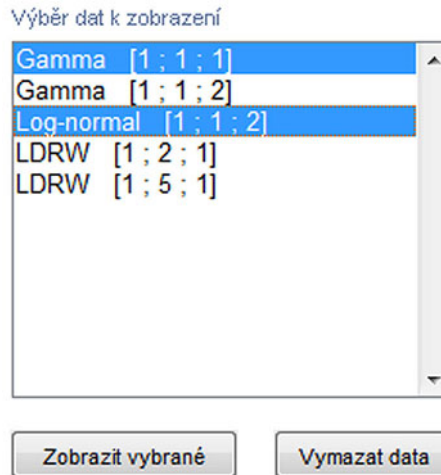
Podmínky ukončení výpočtu se kontrolují pro každé dvě poslední funkční hodnoty. Jejich stanovení je patrné v uvedené části zdrojového kódu. Pro dané hodnoty se spočítá diference a porovná se se zadanou hodnotou. Je-li výsledný gradient záporný, je splněna první podmínka ukončení. Druhá podmínka vyplývá z dosažení definované minimální hodnoty. Pomocí funkce $\max(Y)$ je vrácena nejvyšší dosažená hodnota. Je-li dosaženo poklesu funkce a funkční hodnota přesáhla $0.01\max(Y)$, řídicí proměnné dosáhnou hodnoty $i=n$ a výpočet se ukončí.

Následující část zdrojového kódu ukazuje externí funkci pro výpočet hodnot modelu LDRW. Parametry jsou předávány v obecném tvaru a k jejich předefinování dochází až v samotné funkci, aby byla zajištěna univerzálnost kódu.

```
function y=ldrw(m,p1,p2,t)
    lambda = p1; mi=p2;
    y=m*(1/mi)*exp(lambda)*sqrt((lambda*mi)/(2*pi*t))
        *exp(-(lambda/2)*((t/mi)+(mi/t)));
```

Volba dat a jejich zobrazení

Volba dat dává uživateli možnost přehledněji pracovat s jednotlivými průběhy funkcí. Pro tento účel byl v grafickém prostředí definován prvek *listbox* s názvem *Seznam1*. Jedná se o standardní výčtový seznam, viz obrázek (3.3). Po nastavení ve vlastnostech prvku *maxValue*>1 seznam umožňuje vícenásobnou volbu. Položky jsou tvořeny textovým řetězcem a jeho obsah je generován z prvního sloupce datového pole globální proměnné $Z(j,1)$.



Obr. 3.3: Grafický prvek pro výběr dat

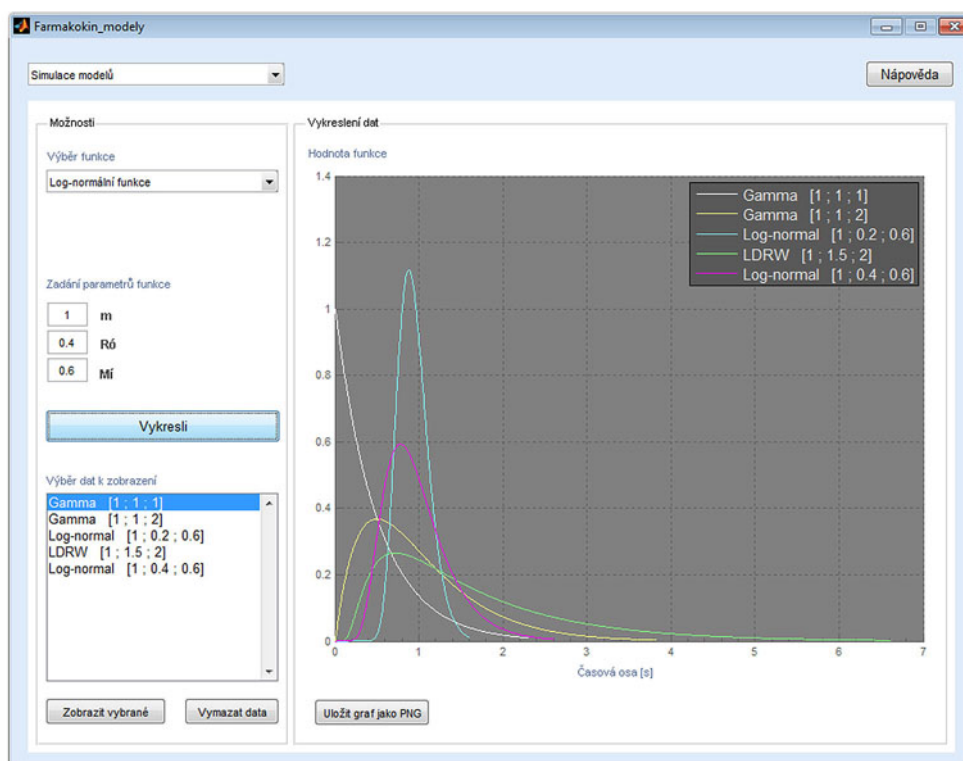
Manipulace s prvkem *Seznam1* je provedena pomocí indexování. Při označení požadované položky je do funkce vrácena hodnota její pozice v seznamu. Poté se převede indexace do datového pole $Z(j,2)$. Číslo dvě znamená, že funkce bude přistupovat k číselným vektorům příslušných křivek. Tyto jsou potom vykresleny do grafu.

Samotné vykreslování je realizováno pomocí funkce `plot()`. Vykreslení probíhá stejným způsobem jak pro nově získaná data, tak pro výběr z prvku *Seznam1*. Jako grafický výstup je zvolen základní dvourozměrný graf s osami x a y . Osa x reprezentuje průběh času a je cejchována v jednotkách [s]. Osa y udává příslušnou hodnotu funkce a je normalizována dle vstupního parametru m .

3.2.3 Ověření výstupu skriptu a provedení simulace

Skript byl spuštěn pomocí příkazové řádky v programovém prostředí Matlab. Ihned po spuštění je uživateli nabídnuto grafické prostředí pro obsluhu vnitřních funkcí, viz obrázek (3.4). V aplikaci jsou patrné všechny prvky potřebné pro provedení simulace. Celá aplikace je rozdělena do dvou částí. Levá část *Možnosti* obsahuje volbu modelu a nastavení vstupních proměnných. V této části je též prováděn výpis zpracovaných dat a možnost jejich volby či vymazání. Pravá část *Vykreslení dat* slouží k samotnému grafickému výstupu. Jeho dominantou je dvou-rozměrný graf se simulovanými křivkami a možnost uložení do grafického formátu *Portable Network Graphics*. Pro testovací úlohu byly zvoleny takové hodnoty funkcí, aby byly průběhy všech křivek zřetelné. Pro zadané parametry i modely proběhla simulace v pořádku

a výpočet byl vždy ukončen v souladu se zadanými kritérii, jak je vidět na obrázku (3.4).



Obr. 3.4: Ukázka uživatelské aplikace se simulovanými modely

3.3 Data fitting a výpočet parametrů

Pojmem data fitting rozumíme proces vytváření takových funkcí, které co nejméně s co nejmenší odchylkou prokládají měřená data. Cílem je získat vztahy mezi proměnnými, které definují danou funkci. Toho se v analýze perfuzních křivek využívá k přesnému stanovení parametrů sledované tkáně. Složitost průběhu dat určuje složitost samotné metody pro proložení. Nejjednodušší úlohy využívají jednoduché lineární funkce – přímkou nebo polynomy různých řádů. Obecně však u biologických dat lze linearitu předpokládat pouze ve zjednodušených případech. Pro přesnou aproximaci se tedy využívá více či méně složité nelineární průběhy funkcí.

Nelineární funkce nejčastěji využívané pro modelování perfuzních křivek byly již popsány v předchozím textu. Následující část proto bude zaměřena na proložení experimentálních dat právě těmito modely. Matematický popis modelů je znám, ale jejich průběh závisí na vnitřních parametrech funkce. Cílem je tedy vytvořit skript, který stanoví parametry modelů s co největší přesností vzhledem k průběhu reálných dat. Tento proces se nazývá *optimalizace funkce*.

3.3.1 Optimalizace funkce

Optimalizace je procesem, který se snaží dosáhnout nejlepšího řešení daného matematického problému. Optimalizační úloha hledá takové řešení pro parametry, aby zkoumaná (účelová) funkce nabývala co nejmenší, nebo co největší hodnoty.[?] Protože zde řešeným problémem je snížit odchylku modelu od experimentálních dat na co nejnižší úroveň, bude úloha definována pouze jako hledání:

$$\min f(x), \quad x \in \mathfrak{R}. \quad (3.1)$$

Je-li účelová funkce jednorozměrná, tedy má-li pouze jednu proměnnou, vychází hledání extrémů z určení derivace pro podezřelé body. Derivace v bodě x_0 vyjadřuje graficky směrnici tečny a je definována pomocí limity:

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}. \quad (3.2)$$

Je-li pro sledovaný bod $f'(x) = 0$, jedná se o stacionární bod, ve kterém se nachází extrém funkce. Minimum nastává v nalezeném extrému v případě, že funkce je na daném intervalu konvexní. Pro každý bod v tomto intervalu potom platí $f''(x) < 0$.

Pro funkce více proměnných se problém stává složitějším. Jednoduché derivace přechází na derivace parciální. Vektor parciálních derivací pro funkci n proměnných je potom gradientem funkce, určeným jako:

$$\text{grad } f(x_1, \dots, x_n) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right). \quad (3.3)$$

Obdobný postup lze použít pro určení konvexnosti a konkávnosti. Druhé derivace pak přechází na druhé parciální derivace vícerozměrné funkce a jejich vyjádřením je Hessova matice ve tvaru:

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}. \quad (3.4)$$

Pomocí uvedených vztahů (3.3) a (3.4) lze nyní stanovit lokální extrémy funkce více proměnných. Za podmínky, kdy $H(f)$ je pozitivně definitní matice, tj. symetrická s vlastními hodnotami > 0 , a $\text{grad } f(x) = 0$, nachází se v bodě x lokální minimum.

Optimalizační úlohy lze klasifikovat mnoha způsoby podle jejich přístupu při určování extrémů funkce. U některých metod lze tento přístup omezit pomocí vstupních podmínek. Odlišuje se také matematický postup, kterým je úloha řešena. Obecně nelze určit jednu metodu jako tu nejlepší. Je třeba zkoumat jejich kvality z hlediska přesnosti i robustnosti pro konkrétní úlohu, kterou je potřeba optimalizovat.

Pro optimalizaci farmakokinetických modelů v této práci byl zvolen algoritmus *Levenberg-Marquardt*. Tento optimalizační algoritmus je součástí balíku funkcí *Matlab Optimization Toolbox 5*. To přináší výhody zejména pro jeho snadné nastavení.

Levenberg-Marquardt

Levenberg-Marquardt algoritmus (dále jen LMA) patří mezi gradientní metody minimalizace obecně nelineárních funkcí. Gradientní metoda je taková, která využívá pro stanovení numerického řešení odhadu gradientu, viz výpočet (3.3). LMA vychází z nelineární Newtonovy metody nejmenších čtverců, kde jsou optimální parametry stanoveny pomocí minimalizace kvadratické funkce [5]:

$$\min \sum_i f_i(x) = \sum_i (y_i - f_i(x))^2. \quad (3.5)$$

Hledané extrémy pro vícerozměrné funkce využívají výpočtu Hessovy matice $H(f)$. Nevýhodou Newtonovy metody je, že $H(f)$ nemusí být vždy pozitivně definitní a nedochází poté k nalezení minima. LMA obchází tento problém stanovením oblasti kolem zkoumaného bodu zvané *trust region*. Pro toto okolí se následně volí korekce tak, aby byl výpočet Hessovy matice validní. [?]

Zásadní modifikací je určování gradientního spádu funkce. Pro jednotlivé funkční hodnoty jsou počítány vektory parciálních derivací –gradient. Jejich uspořádáním do matice je získána *Jacobiho matice*. Konvergence k minimální hodnotě je pak zajištěna „spádem“ ve vektorovém poli vytvořeném Jacobiho maticí [?]:

$$J(f) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \quad (3.6)$$

Kombinace obou technik způsobuje v podstatě rozhodování mezi dvěma extrémy. Jeden je určen pomocí Newtonovy metody, druhý pomocí gradientního spádu.

Levenberg-Marquardt využívá pro optimalizaci iterační metody výpočtu. Před počátkem optimalizace je nutné stanovit počáteční odhad parametrů. V každém iteračním kroku je pak předchozí odhad nahrazen přesnějším výsledkem až do nalezení minima.[?] Obecně lze tento proces vyjádřit jako:

$$\beta_{i+1} = \beta_i + J_i \cdot \lambda, \quad (3.7)$$

kde β je vektor parametrů funkce, J_i Jacobiho matice a λ je parametr určující krok metody.

Pro většinu případů postačuje odhad stanovit na $\beta = [1, 1, \dots]$, aby metoda konvergovala. V případech kdy se tak nestane, je nutné určit odhad blíže k výslednému optimu. Nižší přesnost, kterou LMA vykazuje oproti jiným metodám, je nahrazena dobrou konvergencí metody robustností při optimalizaci dat.[5]

3.3.2 Popis a zpracování dat

Pro otestování vytvořeného skriptu byl použit balík experimentálně získaných dat. Data jsou uložena ve formátu **.mat*, který spadá pod programovací jazyk Matlab. Strukturou jsou řádkové vektory hodnot, které reprezentují průběh veličiny v čase. Data se načítají pomocí dialogového okna. V případě že soubor obsahuje více řádkových vektorů, je načten pouze první z nich. Po načtení je hodnota amplitudy

normalizována tak, aby $y=y/\text{sum}(y)$. Časová osa je cejchována v jednotkách [10 vzorků/s]. V následujícím textu je uveden výčet dat a způsob jejich získání.

Data 1:

- *exp13_vybery.mat*
- *exp14_vybery.mat*
- *exp15_vybery.mat*
- *exp17_vybery.mat*

První balík dat pochází z experimentálního ultrazvukového měření na fantomu. Fantom byl sestaven pomocí *dialyzátoru*, *silikonových trubiček* a *peristaltické pumpy*. Trubičkami je pomocí pumpy čerpán fyziologický roztok. Systém je ponořen do kapaliny, aby docházelo k lepšímu šíření ultrazvukové vlny. Část, kterou protéká roztok je rozdělena na tři sekce pro měření. Vstupní (AIF) a výstupní (VOF) funkce jsou reprezentovány širšími trubičkami před a za dialyzátorem. Samotný dialyzátor představuje oblast zájmu (ROI). Měření proběhlo po aplikaci bolu ultrazvukové kontrastní látky. Tomografická rovina pro snímání ultrazvukového signálu byla vedena přes všechny tři oblasti. Intenzita signálu pak byla zvlášť určena pro každý region[7].

Data 2–1:

- *srdce_roi_kor_filt.mat*
- *srdce_roi_kor2_filt.mat*
- *srdce_roi_nkor_filt.mat*
- *srdce_roi_nkor2_filt.mat*

Data 2–2:

- *srdce_roi_kor_nfilt.mat*
- *srdce_roi_kor2_nfilt.mat*

Skupina souborů *Data 2–1* a *Data 2–2* obsahuje perfuzní křivky z práce[6]. Tyto jsou získány snímáním ultrazvukovou sondou z lidského srdce. Data byla snímána po aplikaci ultrazvukové kontrastní látky (mikrobublinky) a reprezentují změnu intenzity signálu v čase. Ta je přímo úměrná koncentraci kontrastní látky. Křivky byly snímány ve zvolené ROI v oblasti srdečního myokardu. V části *Data 2–2*: byla dodatečně provedena filtrace průměrujícím filtrem s oknem šířky 3-5 vzorků.[6]

Data 3–1:

- *srdce_aif_kor_filt.mat*
- *srdce_aif_kor2_filt.mat*
- *srdce_aif_nkor_filt.mat*
- *srdce_aif_nkor2_filt.mat*

Data 3–2:

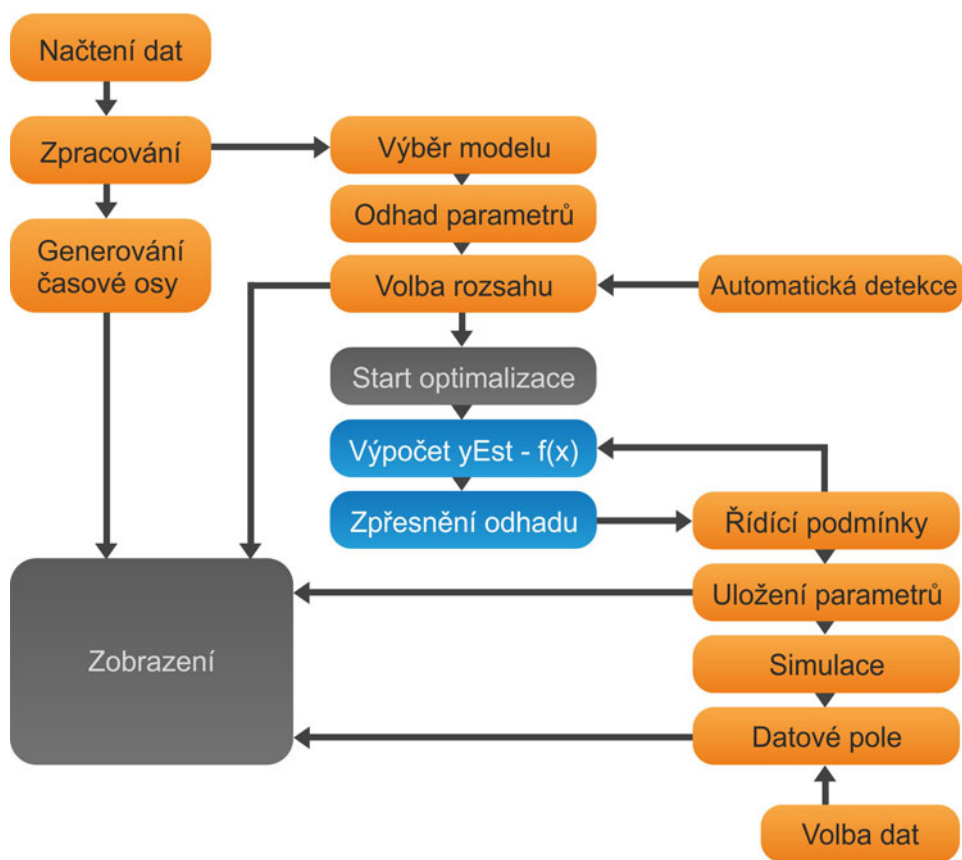
- *srdce_aif_kor_nfilt.mat*
- *srdce_aif_kor2_nfilt.mat*
- *srdce_aif_nkor_nfilt.mat*
- *srdce_aif_nkor2_nfilt.mat*

Třetím souborem dat jsou taktéž signály uvedené v rámci práce [6]. Jedná se o hodnoty intenzity signálu vstupní funkce (AIF) při snímání ultrazvukovou sondou. Snímání AIF bylo provedeno v levé srdeční komoře a v posteriorním úseku myokardu.

Data 3-2: byla po snímání upravena filtrací pomocí průměrovacího filtru s oknem délky 7 vzorků.[6]

3.3.3 Struktura programu

Většina funkcí skriptu pro proložení dat modelem je navázána na řešení simulace do souboru *Farmakokin_modely.m*. Do grafického rozhraní byla přidána možnost volby mezi oběma programy. Výměna panelů je realizována pomocí skrývání a zobrazování příslušných prvků. Načtení dat do programového prostředí probíhá pomocí volby v dialogovém okně. Ihned po načtení dochází k jejich zpracování, vygenerování časové osy a grafické prezentaci.



Obr. 3.5: Schéma skriptu pro proložení dat modelem

Hlavní funkční částí je samotný proces optimalizace. Před jeho spuštěním je provedeno nastavení odhadu počátečních parametrů a volba funkce. Rozšířením skriptu je možnost volby rozsahu dat určených pro výpočet pomocí kaliperů. Spuštěním optimalizace jsou volány externí skripty s jednotlivými modely. Ty byly opět rozděleny do tří nezávislých souborů:

- *gamma_fitting.m*
- *lognormal_fitting.m*
- *ldrw_fitting.m*

Soubory jsou definovány jako samostatné funkce a vrací hlavnímu skriptu hodnoty nutné pro výpočet minima. Hodnoty jsou během dalšího procesu předány funkci pro optimalizaci. Protože je využit *Optimization toolbox*, jedná se o volání externího souboru s touto funkcí. Jejím výstupem jsou optimalizované parametry. Poté dochází k vizualizaci parametrů v GUI pomocí tabulky a je provedena simulace s vykreslením výsledné křivky. Prvky určené pro manipulaci se seznamem dat a grafický výstup zůstávají totožné pro oba podprogramy řešené v rámci této práce. Stručné schéma funkcí celého skriptu je uvedeno na obrázku (3.5).

3.3.4 Funkční a grafické řešení

V této části následuje vysvětlení programového kódu a samotné aplikace pro proložení dat modelem. Jako na stěžejní cíl této práce je popis zaměřen především na samotný proces optimalizace a nastavení jejích parametrů. Ty prvky, které jsou sdílené se skriptem pro simulaci modelů a byly již popsány v předchozím textu v podkapitole 3.2.2, nebudou již blíže specifikovány.

Načtení a zpracování dat, generování časové osy

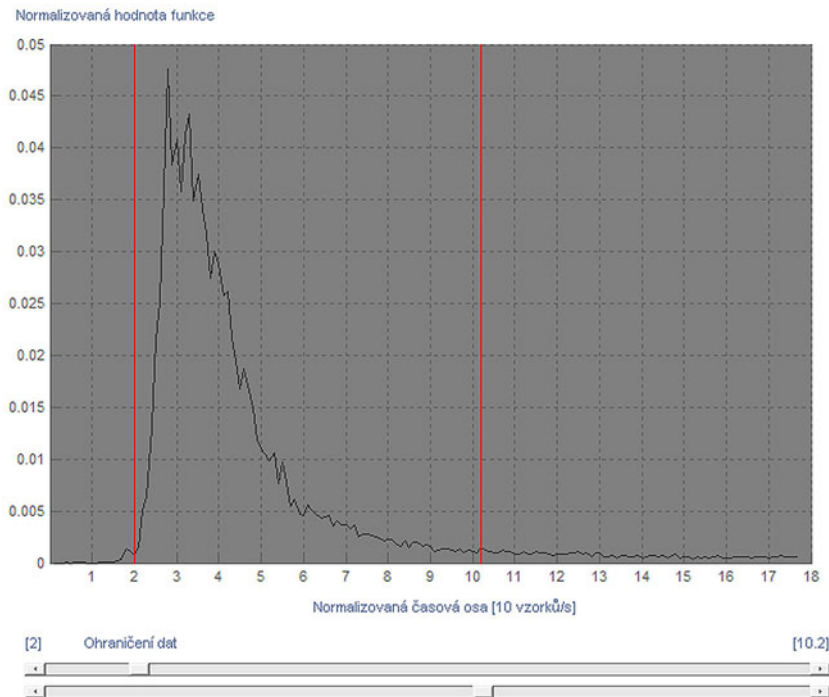
Obě funkce pro načtení i zpracování dat jsou přiřazeny pod grafický prvek tlačítka *Nahrát data*. Načítání je realizováno v dialogovém okně pomocí příkazu `uigetfile`, který vrací skriptu adresu a název souboru. Volba souboru je omezena na datovou strukturu Matlabu s koncovkou **.mat*. Po otevření se ze zvoleného souboru načítá první řádkový vektor hodnot do proměnné *YData*. Data v proměnné jsou následně normalizována. Výpočet probíhá pomocí sumace celého vektoru tak, aby výsledný součet všech hodnot byl jednotkový. Pro délku vektoru je následně vygenerována časová osa do proměnné *XData* v kroku 0.1. Hodnoty časové osy jsou určeny pro úlohu optimalizace a simulace výsledné funkce. Po zpracování proměnné *YData* dochází k vizualizaci načtených dat do dvourozměrného grafu. Časová osa je po vykreslení dat normalizována v jednotkách [*10 vzorků/s*]. Realizaci ukazuje obrázek (3.5) a následující výpis části zdrojového kódu:

```
[FileName,PathName,FilterIndex] = uigetfile({'*.mat'},'Otevřít M-file');
filename = sprintf('%s%s',PathName,FileName);
Opened=open(filename);
if isstruct(Opened)
    Opened=struct2cell(Opened);
else
end
YData=Opened{1,1};
YData(YData<0)=0;
Sum_YData=sum(YData);
YData=YData/Sum_YData;
```

Volba rozsahu dat

Ve většině případech jsou při měření získána data v širším rozsahu, než je vzrůst a pokles sledované fyzikální veličiny. Pokud nejsou tyto hodnoty předem odstraněny, mají významný vliv na výsledek optimalizace. Volba rozsahu dat proto slouží k nastavení rozmezí, pro které se bude optimalizační úloha počítat. Pro určení rozsahu

byly v uživatelském prostředí vytvořeny dva objekty typu *Slider*. Jedná se o grafické posuvníky, které umožňují oboustranné ohraničení dat. Rozsah posuvníku určuje velikost vektoru se vstupními daty. Délka kroku posuvu je stanovena na jeden vzorek.



Obr. 3.6: Ukázka ohraničení dat

Při nastavení posuvníku dochází pomocí příkazu `get(hObject, 'Value')` k získání aktuální pozice. Pozice je ukládána do proměnné `crop1` a `crop2`. Následně je přepočítána vzhledem k rozsahu dat a na příslušnou pozici je vykreslena svislá osa, viz obrázek (3.6). Při spuštění optimalizace jsou z pozičních souřadnic určeny limitní indexy příslušných hodnot ve vektoru proměnné `YData`. V rámci těchto limitů dochází k vytvoření nového datového vzorku a časové osy pro výpočet optima. Ukázkou je část zdrojového kódu pro výpočet limitů.

```

if isempty(crop1) && isempty(crop2)
    limit=[1,length(XData)];
elseif isempty(crop1)==0 && isempty(crop2)
    limit=[crop1*10,length(XData)];
elseif isempty(crop1) && isempty(crop2)==0
    limit=[1,crop2*10];
elseif (isempty(crop1) && isempty(crop2))==0
    if crop1<crop2
        limit=[crop1*10,crop2*10];
    elseif crop1>crop2
        limit=[crop2*10,crop1*10];
    end

XXData=(0.1:0.1:(((limit(2)/10)-(limit(1)/10))+0.1));
YYData=YData(limit(1):limit(2));

```

Pro navrácení kaliperů do výchozích pozic, je generováno tlačítko *Zrušit ohraničení*. Jemu přiřazená funkce zajišťuje anulování pozic kaliperů a změnu indexace na celý rozsah dat.

Nalezení nejlepšího proložení

Možnost nalezení nejlepšího proložení je založena na správném určení rozsahu dat, pro který bude funkce optimalizována. K jeho aktivaci dochází po zaškrtnutí příslušného políčka v GUI. Změna rozsahu dat vychází z detekce náběžné hrany signálu pomocí určování konečných diferencí jeho vzorků. Po spuštění probíhá od počátečních hodnot výpočet diferencí a jejich ukládání do vektoru *G*. Vektor je průměrován pomocí příkazu `mean(G)` a ve chvíli kdy dojde k nárůstu gradientu dvojnásobek průměru předchozích diferencí, je výpočet ukončen. Tato pozice je uložena a později využita pro ohraničení, jak ukazuje následující část zdrojového kódu:

```
while (YYData(i+1)-YYData(i))<=2*(mean(G))
    i=i+i;
    G(i)=abs(YYData(i+1)-YYData(i))
    mean(G)
end
limit(1)=i;
```

Nevýhodou tohoto algoritmu je, že nedokáže s dostatečnou přesností postihnout veškeré průběhy signálu. Při rychlém nárůstu křivky, vysoké amplitudě a případně přítomném šumu dochází k falešné detekci v oblasti velkých výkyvů signálu. Při příliš nízkém gradientu metoda nedokáže příslušnou hranu zachytit vůbec. Jedná se proto spíše o orientační detekci. Lepší výsledky by mohla přinést podstatně větší hustota vzorkování nebo modifikace rozhodovacího pravidla podle vlastností signálu.

Volba vstupních parametrů

Vstupní parametry pro optimalizaci zahrnují především výběr požadovaného modelu a zadání počátečního odhadu jeho proměnných. Výběr modelu je řešen obdobně jako v podkapitole 3.2.2 pomocí prvku *popupmenu*. Pro volbu vstupních parametrů jsou určeny tři textová pole typu *edittext* s názvy *m*, *p1* a *p2*. Pro každý model jsou implicitně nastaveny výchozí hodnoty, které zaručují obecně dobrou konvergenci, následujícím způsobem:

- Gamma funkce [1 1,5 1]
- Log-normální funkce [1 1 1]
- LDRW model [1 1 1]

Pokud dojde k divergenci výpočtu, je možné v editačním poli stanovit odhad blíže k předpokládaným hodnotám. Pomocí deklarace (`get(hObject,'String')`) jsou poté hodnoty předány v podobě textového řetězce do skriptu a následně převedeny příkazem `str2double` na vektor proměnných.

Optimalizace funkce, řídicí podmínky

Část skriptu, která řeší úlohu optimalizace, využívá především příkazů *Optimization toolboxu*. Výchozí metodou je algoritmus *Levenberg-Marquardt*. Optimalizační funkce je volána pomocí příkazu `lsqnonlin`. V základním nastavení je určena k řešení

nelineárního prokládání dat pomocí metody nejmenších čtverců. Funkce je obecně definována jako:

```
x = lsqnonlin(fun,x0,lb,ub,options).
```

Vstupní proměnné lb a ub udávají omezení pro danou metodu. Protože LMA neumožňuje omezující podmínky, jsou ve skriptu definovány jako prázdný vektor. *Options* ovlivňuje nastavení optimalizace. Vektor $x0$ předává hodnoty odhadu parametrů a výstupem x je nalezené minimum daného problému. Metoda `lsqnonlin` vyžaduje dále ke své činnosti uživatelem definovanou funkci fun . Pro jednotlivé modely je tato funkce definována v externím skriptu, viz podkapitola 3.3.3.

Externí skript je volán opakovaně při každém iteračním kroku optimalizace. Skriptu jsou předány vstupní data a hodnoty počátečního, nebo již zpřesněného odhadu parametrů. Jejím základním úkolem je stanovit odchylku mezi experimentálními daty a vypočítaným průběhem na základě pro dané parametry. Výstupem v každém cyklu je vektor odchylek f stanovený jako $YYData - yEst$. Ten je opakovaně předáván optimalizačnímu algoritmu až do nalezení minima.

Výpočet gradientu potřebného pro LMA probíhá automaticky při volání příkazu `lsqnonlin`. Gradient je určen pomocí metody konečných diferencí. V porovnání s uživatelsky definovanou Jaccobiho maticí podává tato metoda pro diskretní hodnoty stejně přesné výsledky s vyšším výkonem.

Parametr *options* je definován pomocí příkazu `optimset`. `Optimset` vytváří strukturu textových řetězců a hodnot, které umožňují nastavení optimalizační úlohy. V rámci této práce bylo použito následující nastavení některých důležitých parametrů:

- *Algorithm* - 'levenberg-marquardt
- *DerivativeCheck* - 'off'
- *DiffMaxChange* - 0,1 (výchozí nastavení)
- *DiffMinChange* - 10^{-8} (výchozí nastavení)
- *Jacobian* - 'off' (výchozí nastavení)
- *MaxIter* - 400 (výchozí nastavení)
- *TolFun* - 10^{-12}
- *TolX* - 10^{-12}
- *ScaleProblem* - 'none' (výchozí nastavení)

Parametry *DerivativeCheck* a *Jacobian* slouží pouze pro kontrolu výpočtů v případě, kdy jsou realizovány uživatelem. *Algorithm* nastavuje pro optimalizaci metodu LMA. Hodnoty pro *DiffMaxChange* a *DiffMinChange* vyhovovaly rozsahu hodnot vstupních dat po jejich normalizaci a jsou proto ponechány ve výchozím nastavení.

Jako řídicí podmínky pro ukončení iteračního cyklu slouží parametry *TolX*, *TolFun* a *MaxIter*. *TolX* a *TolFun* udávají minimální požadovanou odchylku pro funkční hodnoty v závislosti na průběhu v čase. Při dosažení udávané přesnosti je optimalizace ukončena a vráceno nalezené minimum. *MaxIter* nastavuje maximální počet cyklů, které mohou proběhnout a poté ukončuje výpočet. Výchozí hodnota je pro potřeby tohoto skriptu dostačující.

Simulace, prezentace výsledků

V případě kdy proběhne optimalizace úspěšně, jsou parametry uloženy do vektoru proměnných *par*. Pro získané parametry je provedena simulace a výsledná křivka je prezentována společně s experimentálními daty vykreslením do grafu. Prezentace hodnot získaných parametrů je realizována pomocí tabulky v uživatelském rozhraní. Buňky tabulky jsou uspořádány zvlášť pro každý model. Hodnoty jsou předávány ihned po ukončení optimalizace.

Parametry funkcí

	m	lambda	gamma
Gamma	0.0874	2.9350	6.7585
	m	ro	mí
Log-normal	0.0920	0.4201	0.8179
	m	lambda	mí
LDRW	0.0914	5.4165	2.0768

Obr. 3.7: Ukázka tabulky s optimalizovanými parametry

Datové pole, volba dat a jejich zobrazení

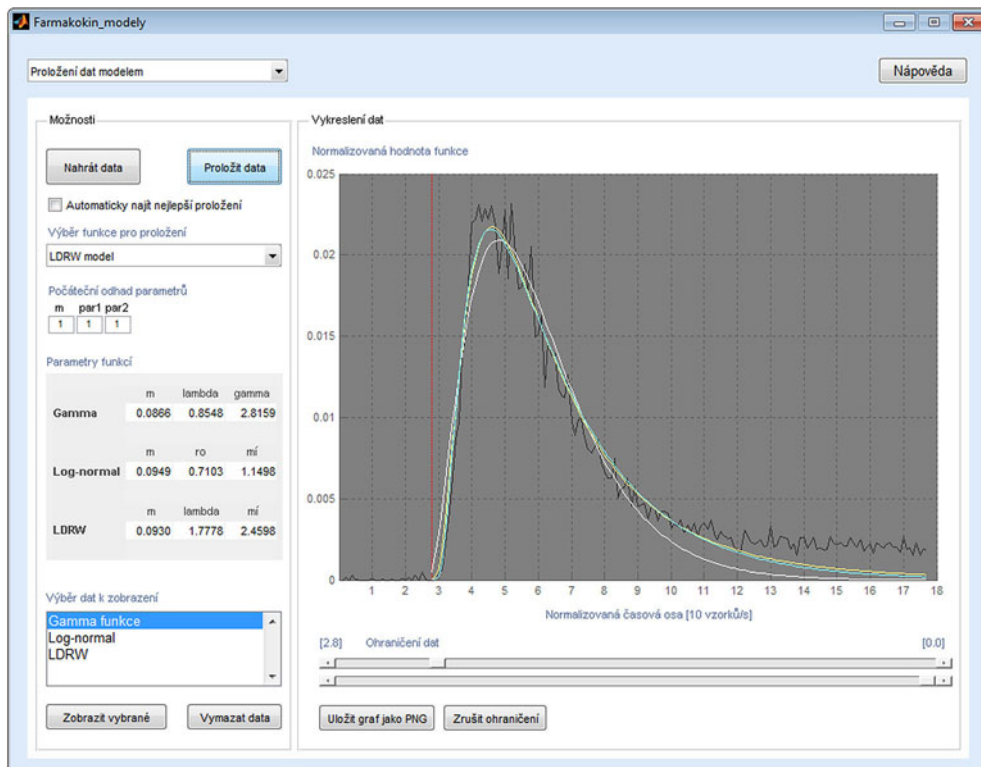
Ukládání dat pro jejich grafickou prezentaci je řešeno obdobně jako u simulace modelů. Výchozí strukturou je globální proměnná *Mat*, do které se vykreslované funkce ukládají ve formě řádkového vektoru. Součástí struktury je také příslušný sloupec pro uložení textového řetězce s názvem modelu. Přístup k jednotlivým polím je řešen pomocí indexace a jejich výběr je stanoven porovnáním textových řetězců. Následná vizualizace je opět provedena ve formě dvourozměrného grafu.

3.3.5 Proložení experimentálních dat

Soubor s hlavním skriptem *Farmakokin_modely.m* byl spuštěn z příkazové řádky prostředí Matlab. Po zvolení možnosti *Proložení dat modelem* jsou automaticky načteny prvky pro ovládání skriptu a pro vizualizaci výstupů. Pro proložení byl využit balík dat, který byl popsán v podkapitole 3.3.2. Na křivky byly náhodně aplikovány zvolené modely a proveden výpočet parametrů pro ohraničená i neohraničená data. Porovnání jednotlivých modelů pro křivku z ultrazvukového fantomu ukazuje obrázek (3.8).

Je patrné, že log-normální i LDRW model mají velmi podobný průběh a získaná data aproximují s dobrou přesností. Gamma funkce zaostává ve strmosti počáteční fáze a dochází tak k větším odchylkám v dosažených maximech. Poměrně mírný pokles amplitudy v závěrečné fázi průběhu je způsoben pomalým odtokem farmaka ze sledované oblasti. Modely v této fázi již svým rozsahem nejsou schopny daný průběh pokrýt.

Další vybrané obrázky proložených dat jsou umístěny v příloze A. V příloze A.4 je prokládána křivka z ultrazvukového fantomu lognormálním a gamma modelem. Z výsledků je patrný vliv hodnot mimo samotný průběh koncentrace. Náběžná hrana je špatně detekována a dochází k významným odchylkám oproti ohraničeným datům. Průběhy v příloze A.5 a A.6 jsou ukázkou proložení dat ze snímání lidského srdce



Obr. 3.8: Ukázka proložení experimentálních dat log-normálním, gamma a LDRW modelem

ultrazvukovou sondou. Nefiltrované signály jsou zobrazeny v A.5 s proložením log-normálním a gamma modelem. Oba modely aproximují takto zarušené průběhy prakticky totožně. V porovnání s vyfiltrovanými daty v C.3 lze však obtížně určit, jestli výsledné křivky odpovídají skutečnému průběhu veličiny. U přílohy A.6 bylo jako ukázka použito proložení všemi třemi modely v kombinaci s určením ohraničení. V tomto případě se projevila nevýhoda statického nastavení počátečního odhadu a pro divergenci výpočtu bylo nutné upravit vstupní parametry. Z křivek je patrné, že v případě menší strmosti je vzestupná část nejlépe aproximována gamma modelem.

4 ZÁVĚR

V rámci bakalářské práce byly popsány základní pojmy z matematických modelů, farmakokinetických jevů a možnosti jejich aplikace v perfuzním zobrazování. Byly vybrány a popsány tři modely, které se nejčastěji používají pro perfuzní analýzu, log-normální, gamma a local density random walk model. V rámci práce byl dále pro tyto modely vytvořen soubor skriptů, který umožňuje jejich simulaci a stanovení parametrů při prokládání experimentálních dat. Aproximace experimentálních dat hraje v dnešní době důležitou úlohu v perfuzních zobrazovacích metodách pro stanovení parametrů sledovaných tkání. Pro realizaci bylo zvoleno programové prostředí Matlab a jeho externí modul pro optimalizační úlohy Optimization toolbox.

Ověření výsledků práce bylo provedeno simulací všech modelů pro náhodně zvolené parametry a prokládáním souboru perfuzních křivek. Při simulaci byl získán přehled o vlivu parametrů na průběh funkce a ověřena funkčnost skriptu pro simulaci. Problematické však bylo vykreslování pro vyšší hodnoty parametrů, především z důvodu výpočetní náročnosti. Prokládání experimentálních dat bylo provedeno pro obdržený vzorek souborů. Znatelné byly především rozdíly při použití gamma modelu, který nedokázal vhodně aproximovat strmé nárůsty amplitudy. Log-normalní i LDRW model dosahovaly téměř stejných výsledků a je patrné, že pro perfuzní křivky je jejich použití vhodnější. Vlastnosti gamma modelu však dokázali dominovat u křivek s nižším gradientem nárůstu a rovnoměrnějším rozložením. Během prokládání se ukázalo, že využití implicitně nastaveného odhadu vstupních parametrů není vždy vhodné. Pro některé průběhy docházelo k divergenci výpočtů a přerušení běhu programu. Tento problém byl vyřešen možností uživatelské volby parametrů, přestože by si daný problém zasloužil přístup spíše přes hlubší analýzu prokládaných dat. Rovněž byl odzkoušen vliv ohraničení dat na výslednou optimalizaci. Toto bylo provedeno jednak pomocí ručního nastavení v grafickém rozhraní, a jednak pomocí automatické detekce náběžné hrany. Je zřejmé, že jednoduchý algoritmus detekce založený na výpočtu diferencí není schopen pokrýt veškeré nároky. Ruční nastavení je tak alternativou při jeho selhání. Z uvedeného závěru jsem přesvědčen, že je nutné nadále zlepšovat všechny prvky perfuzní analýzy, aby dosažené výsledky byly co nejpřesnější. To se týká jak používaných modelů, tak algoritmů pro jejich optimalizaci a vhodné získávání a zpracování dat.

LITERATURA

- [1] LULLMANN, Heinz. - MOHR, Klaus. - WEHLING, Martin. *Farmakologie a toxikologie* Vydání 1. české. Praha: Grada, 2002. 696 s. ISBN 80-7169-976-4.
- [2] DOSTÁLEK, Miroslav. - a kol. *Farmakokinetika* 1. vydání. Praha: Grada, 2006. 220 s. ISBN 80-247-1464-7.
- [3] HOLČÍK, Jiří. - FOJT, Otakar. *Modelování biologických systémů* 1. vydání. Brno: Vysoké učení technické, 2001. 120 s. ISBN 80-214-2023-5.
- [4] CORNELIUS T. L. *Biomechanical systems technology* University of California, Los Angeles, USA, 2007. 328 s ISBN 978-981-270-981-3.
- [5] FLETCHER, R. *Practical method of optimization* Department of Mathematics, University of Dundee, Scotland, UK. 2000. 451 . ISBN 471-915-475.
- [6] KOLAR, R., JIRIK, R., HARABIS, V., MEZL. M., BARTOS, M. *Advanced methods for perfusion analysis in echocardiography* Department of Biomedical Engineering, FEEC, Brno University of Technology, Czech republic. 2010 5 s.
- [7] MEZL, M., JIRIK R., HARABIS V., KOLAR R. *Quantitative Ultrasound Perfusion Analysis In Vitro* Department of Biomedical Engineering, FEEC, VUT, Brno, Czech Republic. 2000. 451 s. ISBN 471-915-475.
- [8] JAN, J., KOZUMPLÍK, J., PROVAZNÍK, I., SZABÓ, Z. *Analysis of Biomedical Signals and Images* Brno: Brno University of Technology, VUTIUM Press, 2000. 408 s. ISBN 80-214-1610-6.
- [9] MELOUN, Milan - MILITKÝ, Jiří. *Kompendium statistického zpracování dat* 1. vydání. Academia, 2002. 766 s.
- [10] LUDEMANN, Lutz - PROCHNOW, Daniel - et al. Simultaneous Quantification of Perfusion and Permeability in the Prostate Using Dynamic Contrast-Enhanced Magnetic Resonance Imaging with an Inversion-Prepared Dual-Contrast Sequence. *Annals of Biomedical Engineering* 2009, VOL. 37, No. 4, s. 749-762.
- [11] XUCAI, Chen - SCHWARZ, Karl Q. - et al. A Mathematical Model for the Assessment of Hemodynamic Parameters Using Quantitative Contrast. *IEEE Transactions on Biomedical Engineering* 1998, VOL. 45, No. 6, s. 754-745.
- [12] SHARON, Chena - HO-LING, Liuc - YIHONG, Yangb - et al. Determination of arterial input function in dynamic susceptibility contrast MRI using group independent component analysis technique. *Nuclear Instruments and Methods in Physics Research* 2006, A 569 , s. 617-621.
- [13] CHENG, Yang - KARCZMAR, Gregory S. - et al. Estimating the Arterial Input Function Using Two Reference Tissues in Dynamic Contrast-Enhanced MRI Studies: Fundamental Concepts and Simulations. *Magnetic Resonance in Medicine* 2004, No. 52, s. 1110-1117.

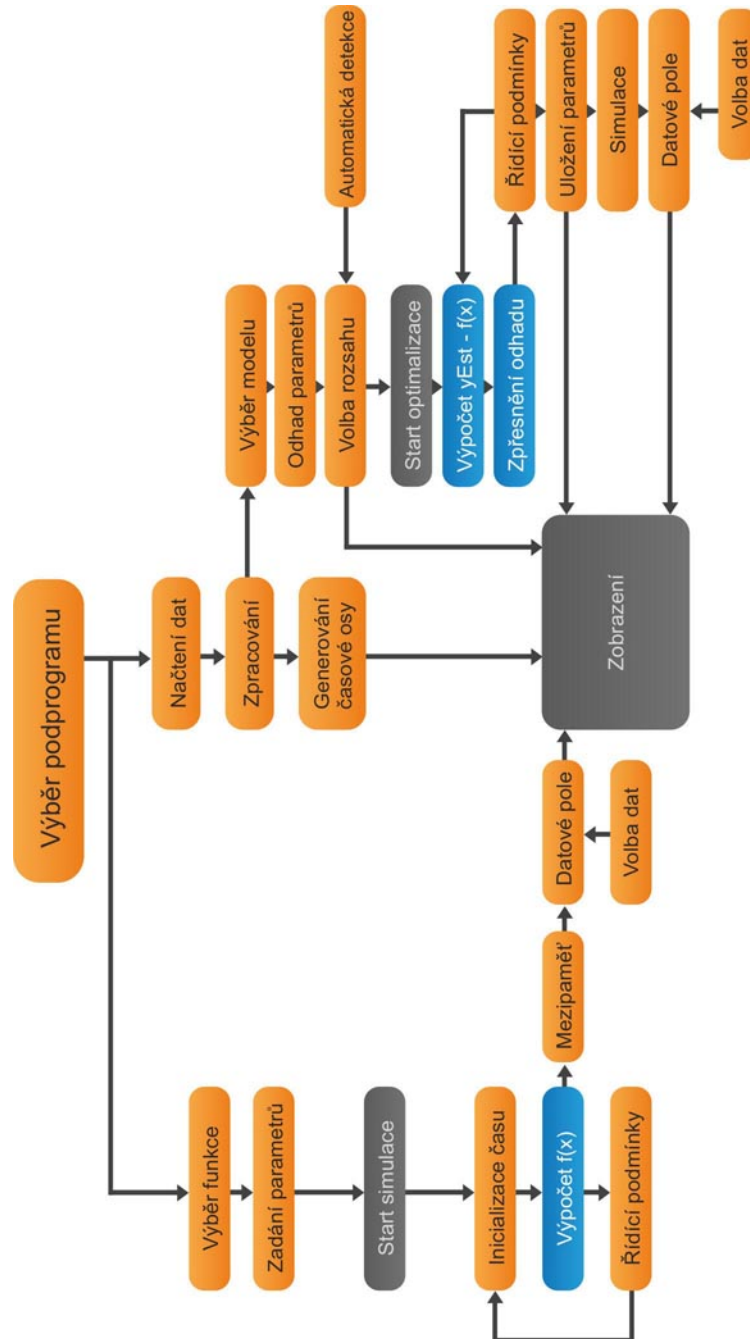
- [14] THOMPSON, Howard K. - STARMER, Frank - Whalen, Robert E. - et al. Indicator Transit Time Considered as a Gamma Variate. *Circulation Research* 1964, no.14, s. 502-515.
- [15] CIOCAN, Eugenia - CIOCAN, Razvan. Optimized numerical pharmacokinetics model for optical molecular probes based on diffusion coefficients in matrigel measured using fluorescence imaging. *31st Annual International Conference of the IEEE* 2009, sborník, s. 4925-4928.
- [16] KOLÁŘ, R. - JIŘÍK, R. - BARTOŠ, M. - HARABIŠ, V. - MEZL, M. *Nové metody perfusní analýzy v chokardiografii*. Brno, přednáška, VUT, 2009.
- [17] HŘEBÍČEK, J - ŠKRDLA, M. *Úvod do matematického modelování*. skriptum. 2006, s. 80.
- [18] RUMINSKI, J. - KARCZEWSKI, . *Automatic Recognition of the Arterial Input Function in MRI Studies* Gdansk, University of Technology

SEZNAM PŘÍLOH

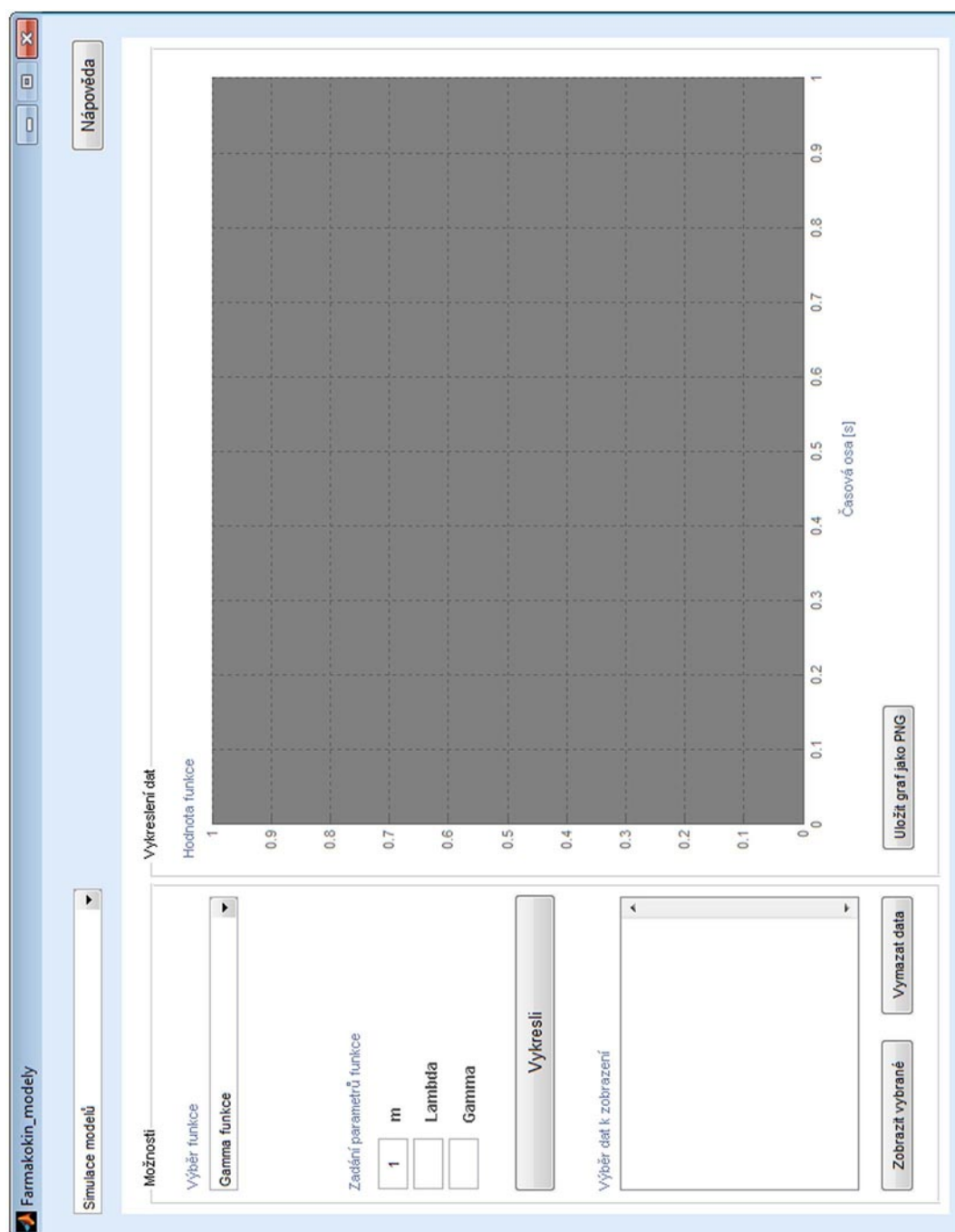
A Grafické přílohy	i
A.1 Strukturní schéma hlavního skriptu	i
A.2 Grafické rozhraní pro simulaci	ii
A.3 Grafické rozhraní pro proložení dat modelem	iii
A.4 Ukázka proložení dat z ultrazvukového fantomu	iv
A.5 Ukázka proložení nefiltrovaných dat ze srdečních oddílů	v
A.6 Ukázka proložení filtrovaných dat ze srdečních oddílů	vi

A GRAFICKÉ PŘÍLOHY

A.1 Strukturní schéma hlavního skriptu



A.2 Grafické rozhraní pro simulaci



A.3 Grafické rozhraní pro proložení dat modelem

Farmakokin_modely

Proložení dat modelem

Nápověda

Možnosti

Nahrát data

Proložit data

Automaticky najít nejlepší proložení

Výběr funkce pro proložení

Gamma funkce

Počáteční odhad parametrů

m par1 par2

1 1.5 1

Parametry funkcí

	m	lambda	gamma
Gamma	0	0	0
Log-normal	0	0	0
LDRW	0	0	0

Výběr dat k zobrazení

Zobrazit vybrané

Vymazat data

Vykreslení dat

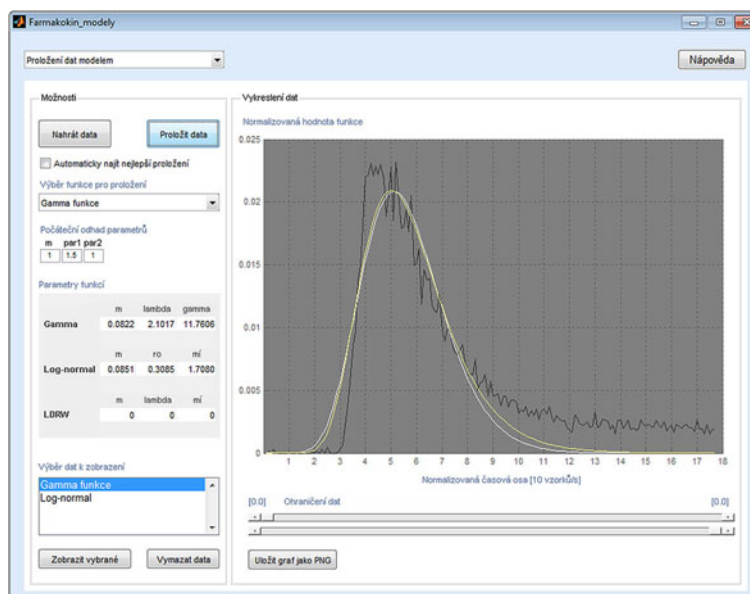
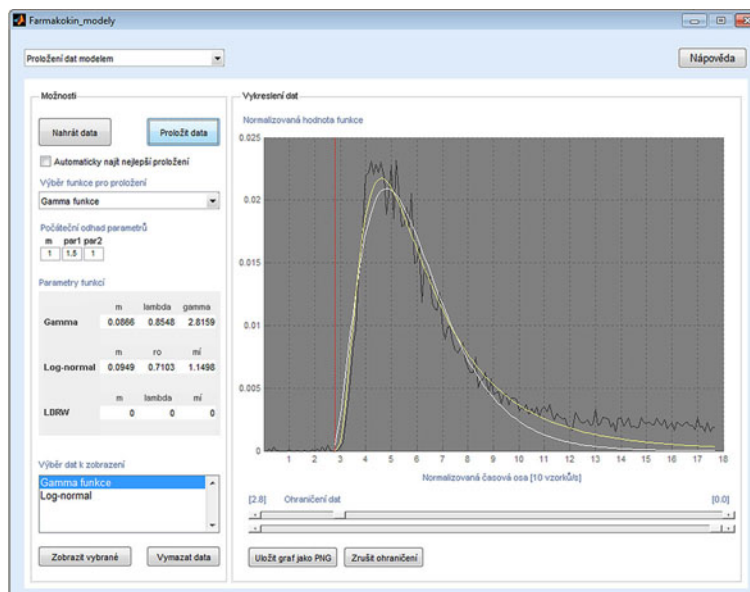
Normalizovaná hodnota funkce

Normalizovaná časová osa [10 vzorků/s]

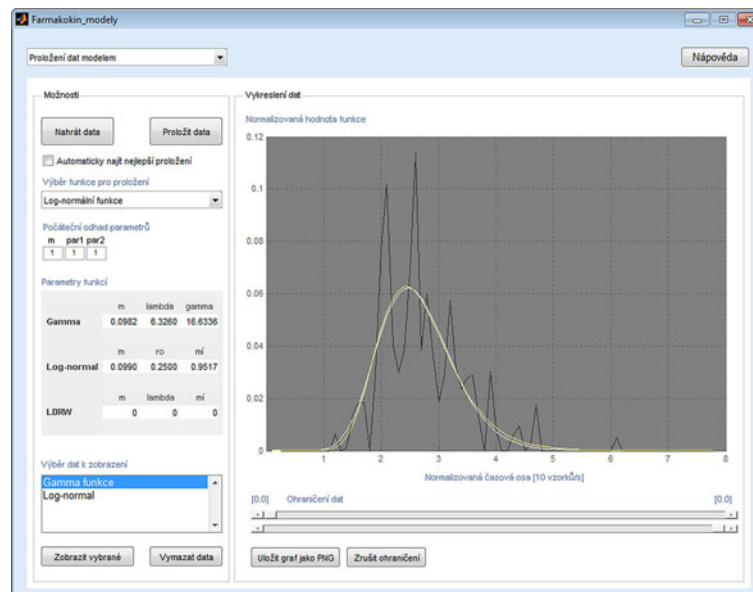
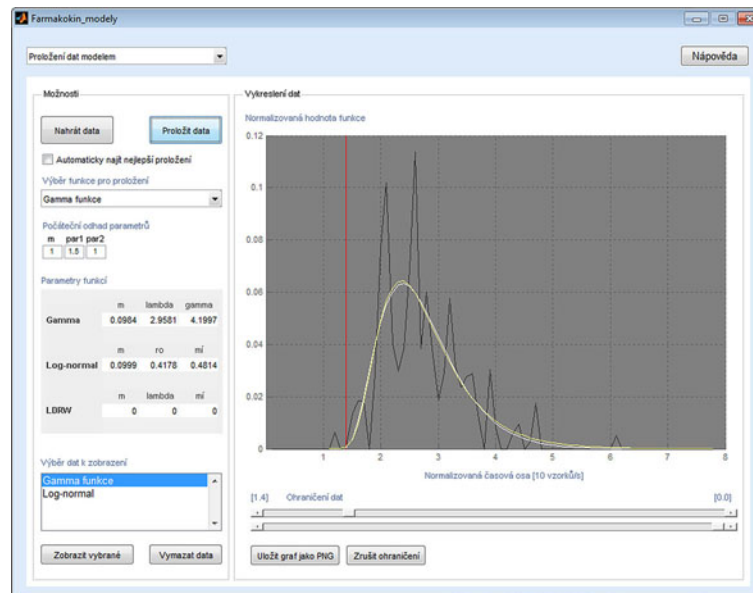
[0.0] Ohraničení dat [0.0]

Uložit graf jako PNG

A.4 Ukázka proložení dat z ultrazvukového fantomu



A.5 Ukázka proložení nefiltrovaných dat ze srdečních oddílů



A.6 Ukázka proložení filtrovaných dat ze srdečních oddílů

