

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Metody počítačového modelování

Užití metod počítačového modelování při analýze biomedicínských signálů

Plzeň 2019

Martin Úbl (A18P0008P)
ublm@kiv.zcu.cz

Obsah

1	Úvod	2
1.1	Homeostáza glukózy	2
1.2	Systém léčby inzulinovou terapií	3
1.3	Motivace	4
1.4	Cíl práce	5
2	Modelování v oblasti léčby diabetu	6
2.1	Kompartmentový model	6
2.1.1	Jednokompartmentový model	7
2.1.2	Dvoukompartmentový model	7
2.1.3	Vícekompartmentový model	8
2.2	Model pacienta	9
2.2.1	Výchozí dílčí modely	9
2.2.2	Výsledný model	14
2.3	Simulace	15
2.3.1	Časové události	16
2.3.2	Provázání simulací	16
3	Numerické řešení obyčejných diferenciálních rovnic	18
3.1	Diferenciální rovnice	18
3.2	Přibližné řešení	19
3.2.1	Eulerova metoda	20
3.2.2	Určení chyby	20
3.2.3	Rungovy-Kuttovy metody	22
3.2.4	Volba kroku	25
3.2.5	Rungovy-Kuttovy metody s adaptivním krokem	26
4	Implementace	28
4.1	Numerické řešení ODR	28
4.2	Syntetický problém	30
4.3	Pozměněný Bergmanův model	30
4.4	SmartCGMS	31
5	Testování	32
6	Závěr	38

1 Úvod

Diabetes mellitus, též hovorově *cukrovka*, je civilizační choroba. Jejím hlavním symptomem je trvale zvýšená koncentrace glukózy v krvi. Vysoká hladina této koncentrace způsobuje progresivní poškození orgánů, které může vést k trvalým následkům. Typickými komplikacemi v dlouhodobém měřítku jsou například neuropatie (poškození periferních nervů), nefropatie (poškození jater), retinopatie (poškození oční sítnice) a další[1].

Mezi akutní komplikace patří hypo- a hyperglykemie. Jako hyperglykemie je označován stav, ve kterém má pacient koncentraci glukózy v krvi vyšší, než 11 mmol/L. Jako hypoglykemie naopak stav, kdy je tato koncentrace nižší, než 3.5 mmol/L. Koncentrace glukózy v krvi zdravého pacienta se pohybuje v rozmezí 3.5 – 5.0 mmol/L v preprandiální fázi, a až 7.0 mmol/L v postprandiální[1].

1.1 Homeostáza glukózy

Jako homeostázu glukózy označujeme kontinuální proces udržování stálé hladiny koncentrace glukózy v krvi. Podílí se na ní jednak trávicí soustava příjmem cukrů v potravě, jednak buňky lidského těla, které glukózu spotřebovávají, ale zejména jde o slinivku břišní, která produkuje příslušné regulační hormony. V neposlední řadě se na homeostáze podílí i játra a další části lidského těla[2].

V trávicím traktu je po příjmu potravy započato trávení. Složitější sacharidy se rozkládají na monosacharidy, mezi které se řadí například glukóza, fruktóza a galaktóza. V konečném důsledku jsou ale všechny přeměněny na glukózu. Glukóza se skrze stěny střev vstřebává do krve a zvyšuje tak svou koncentraci. Na zvýšení hladiny glukózy reagují β -buňky slinivky břišní, které zvýší produkci hormonu inzulinu. Inzulin jednak umožňuje prostup glukózy do buněk skrze buněčnou stěnu, a jednak podporuje ukládání glukózy v podobě zásobního polysacharidu glykogenu. Glykogen se ukládá v játrech a v omezeném množství i ve svazech. Pokud je hladina koncentrace glukózy příliš nízká, α -buňky slinivky břišní začnou produkovat hormon glukagon, díky kterému je zásobní glykogen přeměněn zpět na glukózu. Inzulin tedy v konečném důsledku koncentraci glukózy snižuje, glukagon naopak zvyšuje[2].

Diabetes však může způsobit v těle určité změny, které narušují homeostázu glukózy. Diabetes 1. typu je autoimunitním onemocněním, při kterém

imunitní systém těla ničí β -buňky slinivky břišní, a tedy snižuje produkci inzulínu. Od první manifestace obvykle trvá několik měsíců až roků do doby, kdy tělo diabetika kompletně přestane produkovat vlastní inzulín. Pacient je tedy závislý na exogenním inzulínu.

Oproti tomu tělo diabetika 2. typu inzulín stále produkuje, ovšem jeho objem nestačí pro funkci metabolismu. Hovoříme tedy o zvýšené inzulínové rezistenci. Neléčený diabetes 2. typu může vyústit v nadprodukcii inzulínu, „vyčerpání“ β -buněk, až jejich úplnému zničení. Diabetes 2. typu v tomto případě přechází do diabetu 1. typu. Léčbou je v případě diabetu 2. typu podávání per-orálních antidiabetik, které snižují inzulínovou rezistenci.

Náš výzkum se zabývá léčbou pacientů odkázaných na léčbu inzulínovou terapií. To jsou v drtivé většině pacienti s diabetem 1. typu[2].

1.2 Systém léčby inzulínovou terapií

Inzulín je dávkován zařízením zvaným inzulínová pumpa, a to ve dvou režimech – bolusovém a bazálním[3]. Bolusová dávka inzulínu je jednorázová dávka o větším objemu. Tu si pacient prostřednictvím inzulínové pumpy dává typicky před jídlem pro kompenzaci obsahu cukrů v jídle, případně jako dodatečnou korekci kdykoliv během dne. Bazální inzulín je dávkován pumpou automaticky v krátkých intervalech a výrazně menších objemech. Tento režim dávkování kompenzuje funkce bazálního metabolismu.

Pacient je vybaven systémem pro kontinuální monitoring glukózy (Continuous Glucose Monitoring System, CGMS). Tento systém sestává ze senzoru a bezdrátového vysílače. Senzor se skládá z elektrod a enzymu, který spouští v podkožní tkáni reakci s glukózou a vytváří malý elektrický proud. Ten je díky elektrodám změřen a převeden na koncentraci glukózy v podkoží[4]. Bezdrátový vysílač tuto hodnotu posílá do sběrného zařízení v krátkých intervalech, typicky jednou za 5 minut. Sběrným zařízením může být například mobilní telefon nebo inzulínová pumpa. CGMS se musí pravidelně kalibrovat, aby měřené hodnoty koncentrace glukózy v podkoží byly co nejpřesnější. Kalibrace je prováděna naměřenou koncentrací glukózy v krvi. Tato koncentrace je měřena extrahováním kapky krve na testovací proužek. Proužek je poté zasunut do glukometru, který obdobným způsobem, jaký je použit u podkožních senzorů, změří koncentraci glukózy v krvi. Koncentrace v krvi a v podkoží však nejsou stejné. CGMS často používají vnitřní model pro odhad skutečné hodnoty.

Inzulínová pumpa dává bolusový inzulín vždy na požádání pacienta. Oproti tomu bazální inzulín je dávkován na základě předem nastavené hod-

noty rychlosti infuze.

V moderním pojetí léčby inzulinovou terapií se inzulinová pumpa připojuje na CGMS. Na základě přijímaných hodnot může upravit dávkování bazálního inzulinu. O výpočet rychlosti dávkování bazálního inzulinu se stará regulátor. V současnosti nejvýraznějšími druhy řízení jsou modelově-prediktivní řízení (MPC), proporciálně-integračně-derivační řízení (PID) a řízení na bázi fuzzy logic.

Jediným úkolem takového regulátoru je udržet hladinu koncentrace glukózy co nejblíže cílové nastavené hladině a minimalizovat čas strávený mimo bezpečné rozmezí. Kvalita regulace se posuzuje dle vybrané metriky – nejčastěji jde o součet absolutních diferencí (resp. jejich druhých mocnin) nebo procento času v zadaném rozsahu. Existují však i kompozitní metriky, které např. sdružují uvedené metriky se stabilitou.

Vývoj nových a modifikace stávajících regulátorů a přidružených systémů je doprovázena in-silico testováním. Takové testování však vyžaduje kompletní model prostředí, ve kterém by byl regulátor v budoucnu uplatněn. V tomto případě jde o model pacienta s diabetem 1. typu, model senzoru a inzulinové pumpy.

1.3 Motivace

Na základě současného stavu poznání v oblasti léčby diabetu bylo již navrženo několik řídicích algoritmů. Tyto algoritmy zakládají na MPC[6], PID[5], fuzzy logic nebo experimentálně i jiných paradigmatech[7]. Žádná z těchto prací se však nezabývá návrhem inzulinové pumpy a její architektury, stejně jako vhodností daných řídicích algoritmů pro použití v rámci zařízení s nízkým příkonem a tedy omezenými výpočetními a paměťovými prostředky.

Rovněž v současnosti navrhované řídicí algoritmy mají pevnou podobu a jsou při přenosu na cílového pacienta laděny pouze jejich parametry. Nelze jednoznačně určit, jakou podobu má model pro optimální léčbu inzulinovou terapií. Základem pro jakékoliv ověřování vlastností takového algoritmu je mít dostatečně přesné prostředí, které poskytne jednoznačnou metriku určující úspěšnost regulace. Centrálním prvkem takového prostředí je model pacienta, který na základě nastavených hodnot dávkování inzulinu poskytuje odezvu v podobě změny koncentrací glukózy v příslušných kompartmentech, zejména v krvi.

Jednou z možností, jak se přiblížit optimálnímu modelu je použít genetické programování pro odvození takového modelu[8]. Tato metoda však vyžaduje větší množství datových množin s co možná nejlepším průběhem

inzulinové terapie. Vyvstává tedy problém, jak takové datové množiny získat. Rozhodli jsme se proto vytvořit počítačovou a mobilní hru, která bude analogicky tento problém převádět do podoby jiné – např. udržování herní postavy co nejlépe cílové čáře za soustavného vyrovnávání se s externími vlivy prostředí. Tyto vlivy jsou opět analogicky převedené podněty z reálného scénáře, jako např. příjem potravy nebo tělesná aktivita. Pro takovou hru je nutné mít k dispozici model pacienta, který se bude starat o generování pohybu herní postavy a její reakce na hráčské a vnější podněty.

Modely pacienta jsou však parametrizovány sadou parametrů. Model je tedy potřeba identifikovat. Na základě měřených dat od reálných pacientů, poskytnutých v rámci spolupráce s Fakultní nemocnicí Plzeň bude možné rovněž genetickým programováním provést výpočet těchto parametrů. K tomu je ovšem rovněž potřeba mít co nejpřesněji zachycený model pacienta s možností opakovaně spouštět simulaci, pozorovat změny v průbězích a na základě nich provádět výpočet metriky a další manipulace s parametry.

V současnosti existuje několik takovýchto prostředí. Nejznámějším prostředím je simulátor T1DMS, popř. jeho novější varianta, DMMS.R[9]. Jedná se ale o uzavřené prostředí, které pouze poskytuje rozhraní pro případný řídicí algoritmus. Tato prostředí jsou však plně implementována v jazyce Matlab a Simulink a jsou licencována na každé jednotlivé zařízení. To například znemožňuje spuštění na mobilních zařízeních široké veřejnosti. Taková implementace navíc přináší výkonnostní problémy – vzhledem k tomu, že genetické algoritmy potřebují k nalezení suboptimálního řešení statisíce iterací nad jedním modelem, hledání parametrů by trvalo potenciálně velmi dlouho[10].

1.4 Cíl práce

Cílem této práce je tedy navrhnout a implementovat takový model pacienta, který bude co možná nejpřesněji reflektovat reálnou situaci. V první části se práce zabývá modelem pacienta, konkrétně jeho jednotlivými částmi a jejich provázáním. Druhá část se zabývá numerickým řešením takového modelu, aby bylo možné model integrovat do počítačové a mobilní aplikace. Třetí část se zabývá implementací modelu a numerickým řešením v rámci malé testovací aplikace a existujícího frameworku SmartCGMS, vyvíjeného na Katedře informatiky a výpočetní techniky Fakulty aplikovaných věd ZČU v Plzni. Čtvrtá část se zabývá ověřením vlastností navrženého a implementovaného řešení. Celá práce je zakončena pátou částí, ve které je shrnut aktuální stav a nastíněny další cesty vývoje.

2 Modelování v oblasti léčby diabetu

2.1 Kompartimentový model

V oblasti farmakokinetiky je nejčastějším přístupem k modelování tzv. modelování kompartmentů[11]. Takový přístup modeluje jednotlivé tzv. kompartmenty, které představují koncentraci látky (popř. množství, tok) v jednom prostředí. Typicky se takové modely dělí na jedno, dvou a vícekompartmentové modely. V této terminologii často hovoříme o dvou dějích – distribuce a eliminace. Distribucí se rozumí přechod určitého objemu látky v čase z jednoho kompartmentu do jiného. Eliminací se naopak rozumí odbourávání určitého objemu látky z konkrétního kompartmentu, ale i z celého systému. Všechny tyto jevy musí vykazovat exponenciální charakter – průběh musí být popsateľný exponenciální funkcí, resp. jejich součtem[12].

Obvyklým zápisem takového modelu je diferenciální rovnice, popř. soustava diferenciálních rovnic v případě více kompartmentů. Každá jedna rovnice modeluje změnu právě jedné koncentrace (množství, toku) látky.

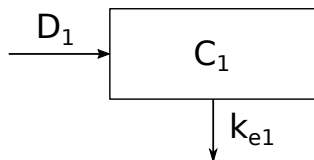
V těchto modelech jsou pro jednotlivé děje použity distribuční, resp. eliminační konstanty. Pro každé dva kompartmenty α a β , mezi kterými probíhá distribuce látky ve směru od α do β , označujeme tento koeficient jako $k_{\alpha,\beta}$. Pro každý kompartment γ , ve kterém dochází k eliminaci látky, je zavedena eliminační konstanta $k_{e\gamma}$ (někdy jen k_γ). Každý kompartment má rovněž definovaný počáteční stav.

Následná simulace s užitím takového modelu vyžaduje kromě počátečních stavů navíc definice vstupů. Obvykle pro každý kompartment α , ve kterém lze uvažovat vstup v libovolném čase simulace, je zaveden vstup D_α . Vstupy nemusí být zahrnuty do rovnic popisujících tento model, pokud nemají popsáný časový průběh. Pak jsou výsadou simulačního prostředí, kde působí jako jednorázové difference v rámci časových událostí a nelze je zachytit v rovnicích. Více o simulacích a časových událostech v kapitole 2.3.

V jiných oblastech je rovněž využíváno velmi podobných postupů – například v Markovských modelech lze analogicky nalézt podobný vzor, kde se rovněž předpokládá exponenciální charakteristika všech přechodů[13].

2.1.1 Jednokompartmentový model

Model s jedním kompartmentem obvykle modeluje vstřebávání látky, například léčiva. Takový model zahrnuje pouze jeden kompartment C_1 , počáteční stav $C_1(0)$ a eliminační děj s konstantou k_{e1} . Volitelně pak vstup D_1 . Tato situace je zachycena na obrázku 2.1.



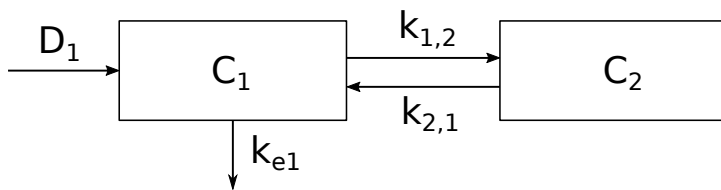
Obrázek 2.1: Model jednoho kompartmentu C_1 s jedním vstupem D_1 a eliminačním dějem popsaným konstantou k_{e1}

Diferenciální rovnice zachycující takový model je zapsána jako:

$$\frac{dC_1}{dt} = -k_{e1} \cdot C_1(t) \quad (2.1)$$

2.1.2 Dvoukompartmentový model

Ve farmakokinetice je mnohem častěji používán model dvou kompartmentů, a to zejména kvůli difúzním vlastnostem jednotlivých prostředí lidského těla. Zahrnuje dva kompartmenty C_1 a C_2 , počáteční stavy $C_1(0)$ a $C_2(0)$, distribuční děje popsané konstantami $k_{1,2}$ a/nebo $k_{2,1}$ a eliminační konstantu k_{e1} a/nebo k_{e2} , dle kompartmentu, ze kterého k eliminaci dochází. Opět mohou být definovány vstupy D_1 a D_2 .



Obrázek 2.2: Model dvou kompartmentů C_1 a C_2 s jedním vstupem D_1 , eliminačním dějem kompartmentu C_1 popsaným konstantou k_{e1} a distribučním dějem mezi kompartmenty C_1 a C_2 v obou směrech popsaných konstantami $k_{1,2}$ a $k_{2,1}$

V tomto případě hovoříme o dělení kompartmentů na dva druhy: centrální a periferní. Za centrální je většinou zvolen právě jeden kompartment, který je např. lépe prokrevovaný a výměna látek je rychlejší, případně je v souvislosti s ním popsáno nejvíce distribučních a eliminačních jevů. Na obrázku

2.2 je znázorněn model dvou kompartmentů, kde lze za centrální kompartment zvolit C_1 a za periferní C_2 .

Takový model lze popsat soustavou rovnic:

$$\begin{aligned}\frac{dC_1}{dt} &= -k_{e1} \cdot C_1(t) - k_{1,2} \cdot C_1(t) + k_{2,1} \cdot C_2(t) \\ \frac{dC_2}{dt} &= +k_{1,2} \cdot C_1(t) - k_{2,1} \cdot C_2(t)\end{aligned}\tag{2.2}$$

2.1.3 Vícekompartmentový model

Složitější farmakokinetické systémy mohou být popsány modely více kompartmentů. Takový model pak obsahuje množinu n kompartmentů

$$C = \{C_1, C_2, \dots, C_n\},$$

množinu počátečních stavů

$$C_0 = \{C_1(0), C_2(0), \dots, C_n(0)\},$$

distribuční jevy popsané podmnožinou koeficientů

$$K = \{k_{p,q} | 1 \leq p, q \leq n \wedge p \neq q\} = \{k_{1,2}, k_{1,3}, \dots, k_{n,n-1}\},$$

eliminační jevy podmnožinou koeficientů

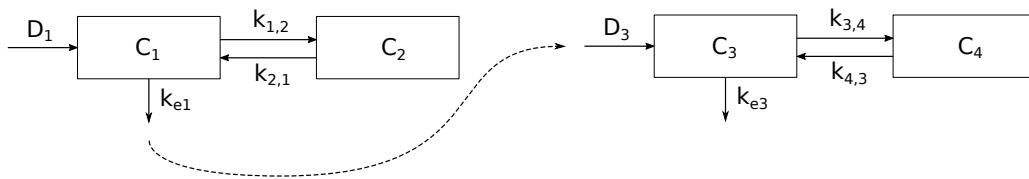
$$K_e = \{k_{e1}, k_{e2}, \dots, k_{en}\}$$

a opět volitelně podmnožinu vstupů

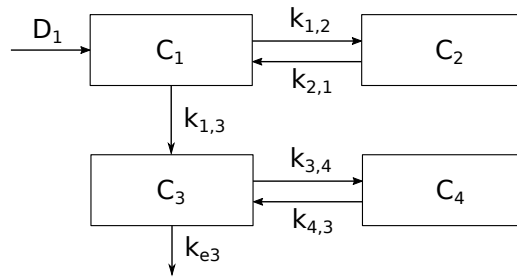
$$D = \{D_1, D_2, \dots, D_n\}.$$

Takový model však ve velké množině případů vznikne provázáním několika dílčích modelů s menším množstvím kompartmentů[14]. Typicky se například může ukázat, že dva oddělené systémy v lidském těle na sobě nějakým způsobem závisí. Modelují se však odděleně a následně je provedeno tzv. provázání. To často spočívá v konverzi eliminačního děje na děj distribuční, kdy je nahrazen vstup druhého modelu distribuční vazbou z modelu prvního. Příklad takových dvou modelů lze vidět na obrázku 2.3. Výsledné provázání modelů lze pak vidět na obrázku 2.4, kdy byl eliminační jev kompartmentu C_1 navázán na vstup kompartmentu C_3 .

Praktickou implikací z provázání dvou modelů tímto stylem je nahrazení eliminační konstanty konstantou distribuční, zde nahrazení k_{e1} za $k_{1,3}$. Dále je tímto odebrán vstup do jednoho z kompartmentů, kde ho nahrazuje nově vzniklá distribuční vazba. Zde byl odebrán vstup D_3 a byl nahrazen distribučním jevem popsaným konstantou $k_{1,3}$. Distribuce nemusí probíhat pouze v jednom směru – pak může nahrazovat část vstupu původního kompartmentu.



Obrázek 2.3: Dva oddělené modely dvou kompartmentů, C_1 a C_2 , C_3 a C_4 , kde lze fyziologicky předpokládat, že eliminační jev kompartmentu C_1 je vázán na vstup kompartmentu C_3 ; modely jsou provázány na obrázku 2.4.



Obrázek 2.4: Provázané modely z obrázku 2.3; eliminační jev kompartmentu C_1 byl navázán na vstup kompartmentu C_3 .

2.2 Model pacienta

Existuje řada modelů diabetického pacienta. Mezi nejznámější patří model Bergmanův[15] a Hovorkův[16]. Tyto modely se však zřídka používají ve své původní podobě a prakticky je jejich implementace rozšířena o modely dalších částí, které se na homeostáze glukózy v lidském těle podílejí.

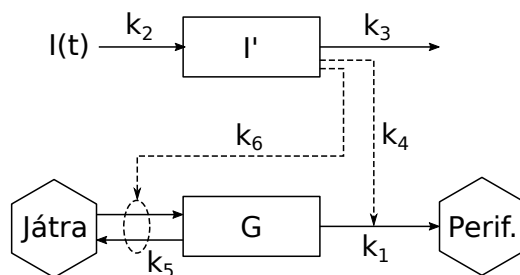
Oba tyto modely a jejich deriváty jsou navrženy jako kompartmentové. Modelují tedy koncentrace látek v jednotlivých prostředích a definují vazby mezi kompartmenty, kde probíhá distribuce látek. Tyto modely vždy v nějaké podobě zahrnují kompartmenty glukózy v krvi a inzulinu v krevní plazmě. Další kompartmenty bývají velmi specifické pro každý model.

2.2.1 Výchozí dílčí modely

Základním, dodnes referenčním modelem homeostázy glukózy je model Bergmanův[15]. Tento model zahrnuje v původní podobě zvané *minimální Bergmanův model* pouze dva kompartmenty: kompartment glukózy v krvi a kompartment aktivního inzulinu¹.

¹v originále „remote insulin pool“, jde o koncentraci v takovém prostředí, kde se inzulin skutečně spotřebovává

Minimální model glukózy



Obrázek 2.5: Minimální bergmanův model popisující základní vztah koncentrace glukózy a inzulinu.

Na obrázku 2.5 je znázorněn minimální Bergmanův model. Tato podoba přímo reflektuje představu o procesech utilizace glukózy a inzulinu. V této podobě však nekoresponduje přímo s podobou kompartmentového modelu. Je potřeba nejdříve zavést dodatečný kompartment X , který představuje již zmíněný kompartment aktivního inzulinu:

$$X(t) = (k_4 + k_6) \cdot I'(t) \quad (2.3)$$

Původní model nemodeluje jeho efekt podobou samostatného kompartmentu. Namísto toho předpokládá, že je přímo svázán s kompartmentem inzulinu v krevní plazmě. Kromě vázaného kompartmentu X navíc zavedeme substituce:

$$\begin{aligned} p_1 &= k_1 + k_5 \\ p_2 &= k_3 \\ p_3 &= k_2 \cdot (k_4 + k_6) \end{aligned} \quad (2.4)$$

Tento model pak lze popsat soustavou diferenciálních rovnic:

$$\begin{aligned} \frac{dG(t)}{dt} &= -(p_1 + X(t)) \cdot G(t) + p_1 \cdot G_b \\ \frac{dX(t)}{dt} &= -p_2 \cdot X(t) + p_3 \cdot (I(t) - I_b) \end{aligned} \quad (2.5)$$

Tabulka 2.1 obsahuje výčet parametrů a veličin pro tento model. V těle diabetického pacienta pak může G_b být vysoké (až za hranicí hyperglykémie) a I_b je předpokládáno nižší, než u zdravého člověka vzhledem k absenci vlastní produkce inzulinu.

Minimální model inzulinu

Nyní je nutné zavést model inzulinové kinetiky. Vstupem do modelu z kapitoly 2.2.1 je $I(t)$, jakožto produkce nebo příjem inzulinu z libovolného

$G(t)$	[mg/dL]	koncentrace glukózy v krvi
$X(t)$	[mU/L]	koncentrace aktivního inzulinu
$I(t)$	[mU/L]	vstupní produkce inzulinu
$I'(t)$	[mU/L]	koncentrace inzulinu v krevní plazmě
I_b	[mU/L]	bazální hladina inzulinu
G_b	[mg/dL]	bazální hladina glukózy
k_1	[1/min]	vliv glukózy na utilizaci v periferich
k_2	[1/min]	míra přenosu krevního inzulinu do aktivního
k_3	[1/min]	míra degradace krevního inzulinu
k_4	[L/(min*mU)]	vliv inzulinu na utilizaci glukózy v perifériích
k_5	[1/min]	vliv glukózy na spouštění glykogeneze a glykogenolýzy
k_6	[L/(min*mU)]	vliv inzulinu na spouštění glykogeneze a glykogenolýzy

Tabulka 2.1: Veličiny a parametry minimálního modelu glukózy

prostředí. Bergman popsal model zdravé slinivky[15] rovnicí:

$$\frac{dI(t)}{dt} = \gamma \cdot [G(t) - p_5]^+ \cdot t - n \cdot (I(t) - I_b) \quad (2.6)$$

Tato rovnice popisuje tzv. endogenní produkci² inzulinu. Tabulka 2.2 obsahuje výčet parametrů a veličin tohoto modelu.

$I(t)$	[mU/L]	endogenní produkce inzulinu
$G(t)$	[mg/dL]	koncentrace glukózy v krvi
I_b	[mU/L]	bazální hladina inzulinu
p_5	[mg/dL]	prahová hladina koncentrace glukózy, při které začne slinivka produkovat inzulin
γ	[mU/(mg/dL)]	míra produkce inzulinu
n	[1/min]	míra odbourávání inzulinu

Tabulka 2.2: Veličiny a parametry minimálního modelu inzulinu

Nicméně diabetik 1. typu nemá vlastní produkci inzulinu vlivem sníženého počtu nebo absence β -buněk slinivky břišní. Tato rovnice tedy pro takového pacienta neplatí.

Pro modelování vnější, tzv. exogenní infuze inzulinu je nutné zavést model odlišný. Takový model předpokládá, že je prováděna infuze inzulinu do těla pacienta. Ten je možné vpravit intravenózně přímo do krve, mnohem častěji se však inzulin vpravuje do podkožní tkáně, odkud se vstřebává do krve. Zavedme proto zjednodušený model Kobayashyho[21]:

²produkce hormonu orgánem s vnitřní sekrecí

$$\begin{aligned}\frac{dI(t)}{dt} &= p_7 \cdot I_{sc}(t) - p_4 \cdot (I(t) - I_b) \\ \frac{dI_{sc}(t)}{dt} &= \frac{U(t)}{V_I} - p_7 \cdot I_{sc}(t)\end{aligned}\quad (2.7)$$

Tabulka 2.3 popisuje veličiny a parametry tohoto modelu.

$I(t)$	[mU/L]	koncentrace dostupného inzulínu
$I_{sc}(t)$	[mU/L]	koncentrace inzulínu v podkožní tkáni
I_b	[mU/L]	bazální hladina inzulínu
p_7	[1/min]	míra vstřebávání podkožního inzulínu
n	[1/min]	míra odbourávání inzulínu
$U(t)$	[mU/min]	rychlost exogenní infuze inzulínu
V_I	[L]	distribuční objem podkožní tkáně

Tabulka 2.3: Veličiny a parametry pozměněného minimálního modelu inzulínu

Minimální model vstřebávání karbohydrátů

Metabolický model musí zahrnovat i příjem potravy. Veškerá energie, kterou lidské tělo v jakékoliv podobě ukládá a spotřebovává, má původ v pokrmech, které přijímá. Tuto energii tvoří skupina karbohydrátů (dále jen CHO), které se vstřebávají v trávicí soustavě. Z těchto CHO jde o cukry, škrob a celulózu. Hlavní podíl mají cukry. Ty se dále dělí na monosacharidy (např. glukóza, fruktóza, galaktóza), oligosacharidy (např. sacharóza, laktóza, maltóza) a polysacharidy. V potravě typicky předpokládáme mono- a oligosacharidy.

Zde model vstřebávání CHO popsal Hovorka[18]. Skládá se ze dvou dalších kompartmentů D_1 (střeva) a D_2 (žaludek). Model dále uvažujeme v modifikované variantě – v té je parametr uváděn jako koeficient v násobení jako převrácená hodnota namísto původního dělení:

$$\begin{aligned}\frac{dD_1(t)}{dt} &= -p_8 \cdot D_1(t) + p_9 \cdot D_2(t) \\ \frac{dD_2(t)}{dt} &= -p_9 \cdot D_2(t) + A_g \cdot D(t) \\ D(t) &= 1000 \cdot d(t)\end{aligned}\quad (2.8)$$

Tabulka 2.4 obsahuje výčet veličin a parametrů daného modelu.

V tento moment je nutné rozšířit původní minimální model glukózy o příjem CHO, jelikož tento děj má přímý vliv na koncentraci glukózy v krvi. Rozšíříme proto model daný rovnicí 2.2.1:

$$\frac{dG(t)}{dt} = -(p_1 + X(t)) \cdot G(t) + p_1 \cdot G_b + p_8 \cdot \frac{D_1(t)}{V_g} \quad (2.9)$$

$D_1(t)$	[mg]	množství CHO ve střevech
$D_2(t)$	[mg]	množství CHO v žaludku
$D(t)$	[mg/min]	přijímané množství CHO
$d(t)$	[g/min]	absolutní rychlost příjmu CHO
p_8	[1/min]	rychlost vstřebávání CHO do krve
p_9	[1/min]	rychlost přechodu CHO do střev
A_g	(bezrozm.)	vstřebávací konstanta (kolik procent CHO se vstřebá)

Tabulka 2.4: Veličiny a parametry pozměněného minimálního modelu CHO

Rovnice byla rozšířena o poslední člen, který popisuje distribuční jev ze střevního kompartmentu do glukózového. Příjem glukózy je rozprostřen do celého kompartmentu, proto je nutné dělit hmotnost distribučním objemem glukózy, abychom obdrželi koncentraci. Z důvodu jednoduchosti předpokládáme rovnoměrné rozložení látky v kompartmentu.

Model podkožní glukózy

V reálném scénáři není však měřena koncentrace glukózy v krvi tak často, aby bylo možné pouze na základě těchto hodnot rozhodnout o optimální dávce inzulínu, a tedy efektivně léčit diabetes. Kontinuálně jsou měřeny hodnoty z podkožní tkáně, kam je zaveden již zmíněný CGMS senzor. Tyto hodnoty jsou senzorem poskytovány jednou za krátkou dobu, nejčastěji 5 minut. Tato koncentrace se mění velmi pomalu, a proto je bezpečné mezi těmito hodnotami křivku interpolovat a získat tak kontinuální signál.

Pro potřeby modelování je však nutné představit model, který zachycuje rozdíl mezi koncentrací glukózy v krvi a v podkoží. Tento vztah byl v minulosti modelován[22] jako:

$$\frac{dG_{sc}(t)}{dt} = \frac{G(t) - G_{sc}(t)}{r} - R. \quad (2.10)$$

$G(t)$	[mg/dL]	koncentrace glukózy v krvi
$G_{sc}(t)$	[mg/dL]	koncentrace glukózy v podkoží
r	[min]	časové zpoždění difúzního jevu
R	[mg/(dL*min)]	absolutní rozdíl v toku glukózy

Tabulka 2.5: Veličiny a parametry modelu podkožní glukózy

Tabulka 2.5 obsahuje výčet veličin a parametrů modelu podkožní glukózy, který zjednodušeně zachycuje difúzní jev mezi krevním a podkožním kom-

partmentem glukózy. Tento jev ale není lineární a nerespektuje farmakokinetické vlastnosti dané situace.

Z tohoto důvodu je lepší uvažovat model jiný. Ze standardní sady modelů lze použít například model Steil-Rebrinové[20], který tento vztah zachycuje lépe. Je však stále příliš jednoduchý – předpokládá, že změna v podkožním kompartmentu glukózy je přímo úměrná koncentraci glukózy v krvi a její první derivaci. Kromě tohoto modelu lze využít model difúzní[19], který odstraňuje některé tyto nedostatky a na reálných datech vykazuje vyšší přesnost. Původní podoba difúzního modelu je popsána rovnicí:

$$\begin{aligned} G_{sc}(\varphi(t)) &= p \cdot G(t) + cg \cdot G(t) \cdot (G(t) - G_{sc}(t)) + c \\ \varphi(t) &= t + \Delta t + k \cdot G_{sc}(t) \cdot \frac{G_{sc}(t) - G_{sc}(t-h)}{h} \end{aligned} \quad (2.11)$$

Model je navržen jako predikční, proto obsahuje funkci $\varphi(t)$. Ta pro zadaný predikční časový interval Δt vyrovnává rozdíly mezi různými pacienty, které způsobují jinou odezvu a jiný čas propagace změn mezi kompartmenty. Pro potřeby této práce však není nutné provádět predikci, ale rekonstrukci koncentrace glukózy v podkoží z kontinuálně dostupných koncentrací v krvi. Proto předpokládejme, že

$$\begin{aligned} \Delta t &= 0 \\ k &= 0, \end{aligned} \quad (2.12)$$

a pak lze z modelu vyjádřit $G_{sc}(t)$:

$$G_{sc}(t) = \frac{p \cdot G(t) + cg \cdot G^2(t) + c}{1 + cg \cdot G(t)} \quad (2.13)$$

Pro dodržení formátu lze převést tento model do podoby diferenciální rovnice zjednodušeně takto:

$$\frac{dG_{sc}(t)}{dt} = \frac{\frac{p \cdot G(t) + cg \cdot G^2(t) + c}{1 + cg \cdot G(t)} - G_{sc}}{h} \quad (2.14)$$

2.2.2 Výsledný model

Nyní je potřeba provést provázání těchto modelů, aby bylo možné získat komplexní představu o celém systému. Každý dílčí model byl již připraven tak, aby ho bylo možné provázat s jiným, souvisejícím. Výsledný model je tedy popsán touto sadou diferenciálních rovnic:

$$\begin{aligned}
\frac{dG(t)}{dt} &= -(p_1 + X(t)) \cdot G(t) + p_1 \cdot G_b + p_8 \cdot \frac{D_1(t)}{V_g} \\
\frac{dX(t)}{dt} &= -p_2 \cdot X(t) + p_3 \cdot (I(t) - I_b) \\
\frac{dI(t)}{dt} &= p_7 \cdot I_{sc}(t) - p_4 \cdot (I(t) - I_b) \\
\frac{dI_{sc}(t)}{dt} &= \frac{U(t)}{V_I} - p_7 \cdot I_{sc}(t) \\
\frac{dD_1(t)}{dt} &= -p_8 \cdot D_1(t) + p_9 \cdot D_2(t) \\
\frac{dD_2(t)}{dt} &= -p_9 \cdot D_2(t) + A_g \cdot D(t) \\
\frac{dG_{sc}(t)}{dt} &= \frac{p \cdot G(t) + cg \cdot G^2(t) + c}{1 + cg \cdot G(t)} - G_{sc}
\end{aligned} \tag{2.15}$$

V této práci dále předpokládáme počáteční stavy dle tabulky 2.6.

Veličina	$G(0)$	$G_{sc}(0)$	$X(0)$	$I(0)$	$D_1(0)$	$D_2(0)$	$U(0)$	$d(0)$
Poč. stav	100 $\frac{mg}{dL}$	95 $\frac{mg}{dL}$	0 $\frac{mU}{L}$	0 $\frac{mU}{L}$	0 mg	0 mg	0 $\frac{mU}{min}$	0 $\frac{g}{min}$

Tabulka 2.6: Počáteční stavy svázaného modelu

Dále uvažujeme parametry vypsané v tabulkách 2.7 a 2.8, které byly přejaty z publikací Bergmana[17], Hovorky[18] a jiných, navazujících prací.

Parametr	p_1	p_2	p_3	p_4	p_7	p_8	p_9
Hodnota	0.028735	0.028344	5.035e-05	0.3	0.04	0.05	0.05

Tabulka 2.7: Parametry svázaného modelu, část 1

Parametr	G_b	V_g	I_b	V_I	A_g	p	cg	c
Hodnota	95	15.4	9.2	12	0.7	0.929	-0.037	1.308

Tabulka 2.8: Parametry svázaného modelu, část 2

2.3 Simulace

Pro praktické využití modelu při řešení výpočetních úloh je nutné realizovat simulaci. Rozlišujeme několik druhů simulací v závislosti na spojitosti simulovaných veličin nebo na podnětech, které simulaci řídí[23]. Jsou jimi například simulace spojitá, diskrétní, událostně řízená a další. Každý takový druh simulace obsahuje tzv. simulační čas, který v závislosti na druhu popisuje buď komprimovaný reálný čas (např. u spojitě simulace) nebo tzv. logický čas (u událostně řízené simulace; představuje kompresi celého stavu simulace)[24].

V kontextu výše navržených modelů jsou všechny veličiny spojité. Celá simulace by proto mohla být spojitá a pro reprezentaci dat provedeno vzorkování simulovaných veličin. Nicméně takto navržená simulace by musela mít přesně stanovené průběhy vstupních funkcí a z podstaty by nemělo být nutné kdykoliv do simulace zasahovat. Výstupem by byl pouze kompletní průběh simulované veličiny na základě předem definovaných vstupů.

To ale není žádoucí vzhledem k důvodům uvedeným v kapitole 1. Simulovaný scénář zde představuje několikadenní časový úsek. Hlavní vlastností simulací je tzv. časová komprese[23], kdy je možné simulovat výpočetně takový úsek za výrazně kratší dobu. Pro čistě výpočetní úlohy není stanovena spodní hranice časové komprese – čím rychleji simulace proběhne, tím lépe. Nicméně v případě, že nad takovou simulací bude vystavěna počítačová hra je naopak nutné definovat škálu pro časovou kompresi. Hráč navíc přímo ovlivňuje vstupní veličiny v průběhu simulace. Generátor problémů takové hry může rovněž generovat vstupy náhodně, případně na základě hráčských rozhodnutí. Je tedy nutné definovat schéma, díky kterému bude možné takové podněty do simulace zavést.

2.3.1 Časové události

Typickým přístupem pro injektování takových podnětů je využití principu tzv. časových událostí[25]. Simulační prostřední spravuje prioritní frontu událostí, ve které se řadí vstupní podněty dle simulačního času, ve kterém mají nastat. V momentě, kdy simulační čas dosáhne hodnoty času události na vrcholku fronty, je simulace pozastavena a příslušné veličiny a vstupy jsou odpovídajícím způsobem upraveny. Poté simulace pokračuje dál.

2.3.2 Provázání simulací

Simulace s modelem pacienta a externími událostmi je situací, která nastane například v již zmíněné počítačové hře. Pro účely odvozování modelů inzulinové terapie a identifikaci modelu pacienta je však nutné provést buď rozšíření simulačního prostředí, nebo jeho provázání s jiným prostředím.

Na Katedře informatiky a výpočetní techniky je vyvíjet simulační framework SmartCGMS[26]. Tento framework z pohledu simulační terminologie je vystavěn jako prostředí pro událostně řízené simulace, ovšem s možností řídit celou simulaci diskrétními vstupy.

Jako jediná možnost, jak efektivně provázat tyto dva rozdílné simulační přístupy se naskytá integrace spojitě simulace tak, že bude ve výchozím stavu pozastavena. Simulace bude definovat synchronizační událost, která

bude sloužit pro „taktování“ spojitě simulace. Příchozí synchronizační událost způsobí, že bude simulace pokračovat, ale jen po krátký časový úsek. Tento úsek je daný intervalem CGMS měření, tzn. 5 minut. Následně budou nové hodnoty všech veličin oznámeny do simulačního prostředí. Spojitá část simulace se znovu pozastavuje a je vyčkáno na další synchronizační událost.

3 Numerické řešení obyčejných diferenciálních rovnic

V minulé kapitole byl nastíněn význam diferenciálních rovnic v kontextu modelování v oblasti léčby diabetu. Diferenciální rovnice modelují míru změny dané veličiny (derivaci funkce, která ji popisuje). Jedná se o velmi užitečný nástroj pro modelování biologických procesů, jelikož velká část takových procesů vykazuje exponenciální charakter. Tato kapitola se krátce zabývá popisem diferenciálních rovnic, ve druhé části je popsáno numerické řešení pomocí Rungových-Kuttových metod.

3.1 Diferenciální rovnice

Diferenciální rovnice je taková rovnice, ve které se vyskytuje proměnná a její n tá derivace ($n > 0$). Nejvyšší stupeň derivace n_{max} libovolně obsažené proměnné určuje pak tzv. řád diferenciální rovnice.

Tyto rovnice lze rozdělit na dva druhy:

- obyčejné diferenciální rovnice (ODR) – obsahují pouze funkce jedné proměnné a jejich derivace
- parciální diferenciální rovnice (PDR) – obsahují funkce i více proměnných a jejich derivace

Dle vztahů obsažených funkcí lze dělit rovnice ještě dle kritéria linearity:

- lineární – rovnice, ve které se každá instance funkce a jejích derivací se vyskytuje pouze lineárně
- nelineární – všechny ostatní rovnice

Tato práce se dále zabývá pouze obyčejnými diferenciálními rovnicemi prvního řádu (lineárními i nelineárními), které jsou v kompartmentových modelech využívány.

Obecně lze například uvést příklad obyčejné diferenciální rovnice prvního řádu ve tvaru:

$$\frac{dy(x)}{dx} = y(x) \quad (3.1)$$

U takové rovnice lze analyticky odvodit řešení:

$$y(x) = C \cdot e^x \quad (3.2)$$

Zde je C libovolná integrační konstanta a dostáváme tedy obecné řešení dané rovnice. Dosazením konkrétní hodnoty C bychom obdrželi tzv. partikulární řešení. Taková hodnota je determinována na základě podmínek, které jsou předem stanoveny – typicky se jedná např. o počáteční podmínku ve formě:

$$y(x_0) = y_0. \quad (3.3)$$

Při takto zadané úloze hovoříme o tzv. počáteční (Cauchyho) úloze. Kromě úloh počátečních existují například úlohy tzv. okrajové. V takových případech je zadána dvojice hodnot funkce nebo jejích derivací. Tato práce se však zabývá výhradně úlohami počátečními.

3.2 Přibližné řešení

Řešením diferenciální rovnice prvního řádu se rozumí hledání takové funkce, která splňuje rovnici:

$$\frac{dy(x)}{dx} = f(x, y). \quad (3.4)$$

Pro takové řešení je nutné mít explicitně zadanou počáteční podmínku (hodnotu y_0) v nultém bodě (x_0), hovoříme-li o počátečních úlohách. Pro řešení diferenciálních rovnic ve strojovém prostředí je možné zvolit několik způsobů. Prvním je symbolické analytické řešení. Problémem takového řešení je ale nutnost dodat veškerá pravidla a postupy, které s analytickým řešením souvisí. Takto řešit by tedy bylo možné pouze jednoduché rovnice. Navíc by bylo nutné apriori znát veškeré náležitosti systému (např. časové události), a to v daném kontextu nemůžeme zaručit.

Další možností je k řešení diferenciálních rovnic využít numerických metod. Řešením pak v některých případech není spojitá funkce, která splňuje výše uvedenou rovnici, ale funkce diskrétní, která řešení dostatečně přesně aproximuje.

Numerické řešení na svém počátku přejímá výchozí stav – bod $[x_0, y_0]$. Každý další krok vygeneruje nový bod $[x_{i+1}, y_{i+1}]$. Výsledkem takového numerického řešení je tedy množina uspořádaných dvojic $\{[x_0, y_0], [x_1, y_1], \dots\}$.

Algoritmus vždy vybere následující x_{i+1} dle nastaveného kroku h , popř. adaptivní funkce determinující krok $h(\epsilon(x))$. Poté je do výpočtu zahrnut jeden nebo více předchozích bodů a je vypočtena hodnota y_{i+1} odpovídající funkční hodnotě v x_{i+1} .

Obecně lze předpis explicitních metod uvažujících k předchozích hodnot zapsat jako:

$$y_{n+1} = f(x_n, y_n, x_{n+1}, x_{n-1}, y_{n-1}, \dots, x_{n-k+1}, y_{n-k+1}) \quad (3.5)$$

Pro $k = 1$ (jednokrokové metody) lze použít zjednodušený zápis:

$$y_{n+1} = f(x_n, y_n, x_{n+1}) \quad (3.6)$$

Metody implicitní pak mají předpis:

$$y_{n+1} = f(x_n, y_n, x_{n+1}, y_{n+1}, x_{n-1}, y_{n-1}, \dots, x_{n-k+1}, y_{n-k+1}) \quad (3.7)$$

3.2.1 Eulerova metoda

Nejjednodušší metodou numerického řešení diferenciálních rovnic je tzv. Eulerova metoda[31]. Jedná se o jednokrokovou metodu s pevným krokem h . Vychází z definice derivace, kdy nahrazuje derivaci členem

$$\frac{dy(x)}{dx} \approx \frac{y(x_{n+1}) - y(x_n)}{h}, \quad (3.8)$$

kde $h = x_{n+1} - x_n$. Pak lze rovnici v původním tvaru 3.4 přepsat jako:

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \quad (3.9)$$

Krok h lze volit v závislosti na vlastnostech dané funkce a požadované přesnosti. Více o volbě kroku pojednává kapitola 3.2.4. Eulerova metoda je příkladem velmi jednoduchého přístupu k numerickému řešení ODR. Vykazuje však poměrně velkou chybu, jelikož funkci na intervalu $< x_n; x_{n+1} >$ aproximuje lineárně, tedy předpokládáme, že v rámci jednoho kroku se derivace nemění, popř. mění, ale pouze nepodstatně. Nicméně chyba se po čas řešení kumuluje a numerické řešení na velkých intervalech může vykazovat velké chyby.

3.2.2 Určení chyby

Jedním z hlavních kritérií výběru metody je její přesnost. Numerické řešení libovolného problému zřídka poskytuje přesné hodnoty – takřka vždy poskytuje hodnoty takové, které dostatečně přesně aproximují exaktní řešení.

Označme hodnotu produkovanou numerickou metodou jako $\hat{z}$ a exaktní řešení jako z .

Standardně můžeme uvažovat například chybu absolutní, definovanou jako:

$$\epsilon = |z - \hat{z}| \quad (3.10)$$

Taková chybová metrika však nebere v potaz jakýkoliv kontext problému, a tedy neříká nic o tom, jak je chyba skutečně signifikantní. Pro některé problémy může být absolutní chyba velikosti 10^{-5} příliš velká, zatímco pro jiné může být dostatečně malá.

Místo této metriky lze využít chybu relativní, definovanou jako:

$$\eta = \left| \frac{z - \hat{z}}{z} \right| \quad (3.11)$$

Tato metrika uvažuje škálu hodnot poměřením absolutní chyby s přesným řešením. Tím odpadá problém s absolutními jednotkami a výsledkem je poměr. Navíc pokud lze chybu vyjádřit jako násobek 10^{-k} , lze předpokládat, že v aproximovaném výsledku je přibližně k nejvíce signifikantních míst správně (v dekadickém zápisu). Jiným, lepším řešením by bylo správně škálovat celý problém tak, aby absolutní chyba dávala poměřitelné výsledky. To pak umožňuje poměřovat mezi sebou i více stejně škálovaných různých problémů. Relativní chyba by sice poskytovala poměřitelné výsledky, ale špatně škálovaná sada problémů by mohla vést k jiným problémům při jejich numerickém řešení, např. ztrátu přesnosti vlivem strojové reprezentovatelnosti čísel.

Při posuzování rychlosti konvergence se dále používá obecné vyjádření pomocí notace „velké O“ a „malé o“. Pro libovolné dvě funkce $f(h)$ a $g(h)$ říkáme, že

$$f(h) = O(g(h)), h \rightarrow 0, \quad (3.12)$$

pokud existuje C takové, že

$$\left| \frac{f(h)}{g(h)} \right| < C, \quad (3.13)$$

pro dostatečně malá h . Respektive, vztah lze vyjádřit jako

$$|f(h)| < C \cdot |g(h)| \quad (3.14)$$

pro dostatečně malá h . To znamená, že $f(h)$ se blíží nule alespoň tak rychle, jako $g(h)$. Funkce $g(h)$ je obvykle volena jako h^p . Pak hovoříme o chybě $O(h^p)$, tedy chybě řádu p .

O něco silnějším předpokladem je notace „malé o“. Pro libovolné dvě funkce $f(h)$ a $g(h)$ říkáme, že

$$f(h) = o(g(h)), h \rightarrow 0, \quad (3.15)$$

pokud

$$\left| \frac{f(h)}{g(h)} \right| \rightarrow 0, h \rightarrow 0. \quad (3.16)$$

Pokud hodnoty vykazují $O(h^p)$, hovoříme o chybě řádu p . Čím vyšší je řád chyby, tím je metoda přesnější.

3.2.3 Rungovy-Kuttovy metody

Významnou skupinou metod numerického řešení diferenciálních rovnic jsou Rungovy-Kuttovy metody. Tyto metody se dále dělí na explicitní, vložené a implicitní.

Pro diferenciální rovnici

$$\frac{dy(x)}{dx} = f(x, y) \quad (3.17)$$

je numerické řešení aproximováno předpisem

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i, \quad (3.18)$$

kde k_i je definováno jako:

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + c_2 h, y_n + h(a_{21} k_1)) \\ k_3 &= f(x_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)) \\ &\dots \\ k_s &= f(x_n + c_s h, y_n + h \sum_{j=1}^s a_{sj} k_j). \end{aligned} \quad (3.19)$$

Zde je $A = [a_{ij}]$ maticí koeficientů pro odhad derivací v mezikrocích (zvaná „Rungova-Kuttova matice“), $b = [b_i]$ je váhový vektor a $c = [c_i]$ je vektor uzlů. Dále s představuje počet stupňů dané metody.

Každá Rungova-Kuttova metoda je stabilní, právě tehdy, když platí, že

$$\sum_{j=1}^i a_{ij} = c_i, \quad (3.20)$$

pro $i = 2, 3, \dots, s$.

Rungova-Kuttova matice, váhový a uzlový vektor se nejčastěji reprezentují v podobě Butcherovy tabulky[29]. Obecnou podobu lze vidět v tabulce 3.1.

c_1	a_{11}	a_{12}	$\cdots$	a_{1s}
c_2	a_{21}	a_{22}	$\cdots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s,s}$
	b_1	b_2	$\cdots$	b_s

Tabulka 3.1: Obecný předpis Butcherovy tabulky pro Rungovy-Kuttovy metody

Pro každou z Rungovu-Kuttovu metod, tedy každou definici Rungovy-Kuttovy matice, váhového a uzlového vektoru, lze odvodit lokální chybu (v jednom kroku). Pokud lze chybu zapsat jako $O(h^p)$, metoda vykazuje chybu řádu p a sama je tedy metodou řádu $p - 1$.

Explicitní Rungovy-Kuttovy metody

Takzvané explicitní Rungovy-Kuttovy metody se vyznačují tím, že jejich Rungova-Kuttova matice je dolní trojúhelníkovitá.

Již zmíněná Eulerova metoda je rovněž převeditelná na explicitní Rungovu-Kuttovu metodu s Butcherovou tabulkou zapsanou v tabulce 3.2. V této podobě se nazývá *dopředná Eulerova metoda*. Tato metoda vykazuje lokální chybu druhého řádu – jedná se tedy o metodu prvního řádu.

0	0
	1

Tabulka 3.2: Butcherova tabulka dopředné Eulerovy metody

Jako další příklady explicitních Rungových-Kuttových metod lze uvést metodu druhého řádu – Heunovu[32] (tabulka 3.3), metodu třetího řádu – Kuttovu[30] (tabulka 3.4) a metodu čtvrtého řádu – pravidla 3/8[31] (tabulka 3.5). Existuje jich však celá řada.

Vložené Rungovy-Kuttovy metody

Jako vložené Rungovy-Kuttovy metody¹ se označují takové, které kromě hlavní metody řádu p zahrnují ještě dodatečnou metodu řádu $p - 1$ [27].

¹z angl. originálu „embedded Runge-Kutta methods“

0		
1	1	
	$\frac{1}{2}$	$\frac{1}{2}$

Tabulka 3.3: Butcherova tabulka Heunovy metody

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Tabulka 3.4: Butcherova tabulka Kuttovy metody

0				
$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Tabulka 3.5: Butcherova tabulka metody pravidla 3/8

Ta se používá pro odhad lokální chyby v daném kroce. Obecná Butcherova tabulka pro vložené metody je znázorněna v tabulce 3.6.

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$		$\ddots$		
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s,s-1}$	
	b_1	b_2	$\cdots$	b_{s-1}	b_s
	b_1^*	b_2^*	$\cdots$	b_{s-1}^*	b_s^*

Tabulka 3.6: Obecný předpis Butcherovy tabulky pro vložené Rungovy-Kuttovy metody

Tento odhad se využívá pro adaptivní velikost kroku. Více o těchto metodách pojednává kapitola 3.2.5. Příkladem může být například metoda pátého řádu Dormand-Prince[28] s odhadem chyby metodou čtvrtého řádu. Její Butcherova tabulka je znázorněna v tabulce 3.7. Metoda Dormand-Prince je implementována v prostředí Matlab a Simulink pod názvem ODE45[33].

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Tabulka 3.7: Butcherova tabulka metody Dormand-Prince

Implicitní Rungovy-Kuttovy metody

Rungovy-Kuttovy metody označené jako implicitní již nevykazují dolní trojúhelníkovitou vlastnost Rungovy-Kuttovy matice. Tyto metody mají určité výhody a jsou vhodnější pro řešení tzv. tuhých rovnic².

Oproti metodám explicitním a vloženým, při řešení implicitními metodami se může stát, že bude zapotřebí výrazně větší počet výpočetních operací.

Příkladem implicitní metody je například Eulerova zpětná metoda, jejíž Butcherova tabulka je znázorněna v tabulce 3.8

1	1
	1

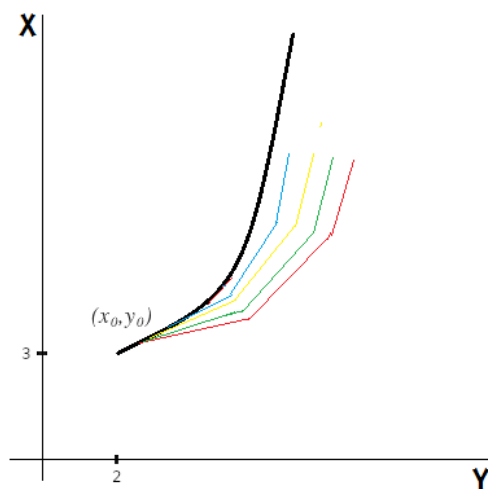
Tabulka 3.8: Butcherova tabulka zpětné Eulerovy metody

Tato práce se dále implicitními metodami nezabývá.

3.2.4 Volba kroku

Pro každou iterativní metodu je kritická volba kroku, se kterým bude daná metoda fungovat. Příliš velký krok může vést ke snížené přesnosti, příliš nízký krok k příliš velkému množství iterací a tedy větší výpočetní náročnosti.

²z angl. originálu „stiff equations“



Obrázek 3.1: Eulerova metoda aplikovaná na diferenciální rovnici $\frac{dy(x)}{dx} = x^2 + 3x$ s počáteční podmínkou $f(2) = 3$; modrou barvou je znázorněna aproximace s nejmenším krokem; zdroj: brilliant.org

Velikost kroku nelze určit univerzálně – různé problémy vyžadují různou velikost kroku. Efekt různé velikosti kroku pro Eulerovu metodu lze vidět na obrázku 3.1.

Aby nebylo nutné explicitně žádat nastavený správný krok při spouštění metody, a tedy nutně nastavovat větší, než je v každém bodě nutný, definují se tzv. Rungovy-Kuttovy metody s adaptivním krokem.

3.2.5 Rungovy-Kuttovy metody s adaptivním krokem

Pro řízení velikosti kroku byly definovány Rungovy-Kuttovy metody vložené. Ty obsahují dvě metody, jednu pro výpočet hodnoty v další iteraci a druhou, nižšího řádu, pro odhad lokální chyby.

V každé iteraci jsou vypočteny hodnoty dalšího kroku pomocí obou těchto metod – použitím vektoru b je získáno řešení y_{n+1} , použitím vektoru b^* je získáno řešení y_{n+1}^* . Odhadem chyby pak může být

$$\epsilon = |y_{n+1} - y_{n+1}^*| \quad (3.21)$$

při použití chyby absolutní. Pro řešení je pak předem zvolen práh chyby ϵ_0 . Pokud platí, že $\epsilon > \epsilon_0$, je řešení zamítnuto, velikost kroku je zmenšena a iterace se opakuje s novou délkou kroku. Tento proces se může několikrát opakovat, pokud je zmenšení kroku nedostatečné.

Pokud je však $\epsilon < \epsilon_0$, řešení je akceptováno a pro další iteraci může

být krok naopak prodloužen – je totiž možné, že současná velikost kroku je pro daný úsek diferenciální rovnice příliš krátká, a tedy se zbytečně provádí výpočty, které lze zredukovat bez významné ztráty přesnosti.

Strategie změny délky kroku

Jako nejjednodušší variantou výpočtu nové délky kroku se jeví binární dělení[34]. Pro tuto strategii platí, že nová délka kroku h_{new} se řídí pravidlem:

$$h_{new} = \begin{cases} h/2, & \epsilon \geq \epsilon_0 \\ 2h, & \epsilon < \epsilon_0 \end{cases} \quad (3.22)$$

Existují však metody pro přesnější odhad nové délky kroku. Kupříkladu Rungova-Kuttova-Fehlbergova metoda se řídí pravidlem[31]:

$$h_{new} = \begin{cases} \beta h (\frac{\epsilon_0}{\epsilon})^{\frac{1}{p+1}}, & \epsilon \geq \epsilon_0 \\ \beta h (\frac{\epsilon_0}{\epsilon})^{\frac{1}{p}}, & \epsilon < \epsilon_0, \end{cases} \quad (3.23)$$

kde β je bezpečnostní koeficient z intervalu $(0; 1)$ (obvykle $\beta \simeq 1$) a p je stupeň metody.

S adaptivním krokem však vyvstává problém v kontextu zmíněných simulací – každé dvě po sobě jdoucí hodnoty od sebe musí být vzdáleny vždy stejně. Tato vzdálenost je v tomto případě definována jako rozestup mezi měřeními podkožního senzoru, tedy 5 minut.

Prakticky je tento problém řešitelný tak, že je v rámci každé iterace algoritmu, která by normálně představovala pětiminutový úsek, spravována vnitřní proměnná h_p akumulující již ураženou vzdálenost v této iteraci. Iterace algoritmu je dokončena právě tehdy, když tato proměnná nabyde hodnoty pěti minut.

Tento problém je v případě binárního dělení vyřešen implicitně pro zmenšování kroku – počet zbývajících kroků se dvojnásobí. Pro zvětšování kroku je však nutné zavést omezení – dvojnásobit krok pouze tehdy, je-li zbývajících počet kroků sudý. Zbývajících počet kroků lze snadno odvodit, jelikož v případě binárního dělení vždy existuje celočíselná konstanta C , pro kterou platí, že $h_p + C \cdot h = 5$.

V případě Rungovy-Kuttovy-Fehlbergovy metody je nutné explicitně omezit velikost kroku tak, aby nikdy nepřesáhla zbývajících vzdálenost – je nutné explicitně v algoritmu uvést podmínku, při jejímž porušení je velikost kroku nastavena na hodnotu zbývajících vzdálenosti.

4 Implementace

Tato kapitola se krátce zabývá implementací numerického řešení ODR, aplikací na syntetický problém z oblasti testovacích úloh a aplikací na pozměněný Bergmanův model. Závěrem kapitoly je popsána integrace do systému SmartCGMS.

Pro implementaci byl zvolen jazyk C++, a to jednak kvůli samotnému kontextu práce, ve které je centrálním jazykem, a jednak kvůli jeho vlastnostem – rychlost výsledného kódu, vysoká úroveň optimalizací, robustnost, bezpečnost.

4.1 Numerické řešení ODR

Celá část, která se stará o numerické řešení byla implementována jako hlavičková knihovna. Tato nutnost vyvstala vzhledem k implementaci kompletně pomocí šablon, které díky svým vlastnostem mají velmi vysoký optimalizační potenciál.

Řešení se skládá ze dvou souborů:

- `ode_solvers.h` – samotné třídy implementující Rungovy-Kuttovy metody
- `ode_solver_parameters.h` – pomocný hlavičkový soubor obsahující parametrizace Rungových-Kuttových metod (matici, váhové a uzlové vektory)

Centrální třídou je šablonová třída `CRunge_Kutta_ODE_Solver_Base`, která je parametrizována šablonovým parametrem N představujícím stupeň metody. Tento společný předek jednak uchovává matici A (atribut `mRKMatrix`), váhový vektor b (`mWeights`) a vektor uzlů c (`mNodes`). Dále interně definuje metodu pro výpočet k_i dílčích odhadů v průběhu iterace. Nakonec definuje rozhraní `Step`, kterým vnější kód provede další krok (iteraci) výpočtu.

Od této třídy dědí dvě další třídy:

- `CRunge_Kutta_ODE_Solver_NonAdaptive` – varianta pro metody s fixním krokem
- `CRunge_Kutta_ODE_Solver_Adaptive` – varianta pro metody s proměnlivým krokem, navíc obsahuje šablonový parametr se třídou definující adaptivní strategii

Každá z těchto tříd definuje vlastní variantu metody pro zpracování sumy vážených k_i koeficientů v součinu s váhovou maticí b , respektive b^* . Neadaptivní metody provádějí prostou sumu dle rovnice 3.18. Adaptivní metody provádějí tutéž sumu, jen dvakrát – jednou pro metodu řádu N a jednou pro metodu řádu $N - 1$. Rovněž je proveden odhad chyby.

Obě tyto metody jsou agregovány třídou `CRunge_Kutta_ODE_Solver`, která při šablonovém instancování volí podlehlou implementaci a příslušnou variantu třídy.

Dále byly implementovány dvě strategie pro adaptivní délku kroku:

- `CRunge_Kutta_Adaptive_Strategy_Binary_Subdivision` – pro binární dělení délky kroku
- `CRunge_Kutta_Adaptive_Strategy_Optimal_Estimation` – pro suboptimální odhad délky kroku (dle Rungovy-Kuttovy-Fehlbergovy metody)

Z jednotlivých parametrizací byly implementovány varianty:

- Eulerova metoda (jm. prostor `euler`)
- Heunova metoda (jm. prostor `heun`)
- Kuttova metoda (jm. prostor `kutta`)
- metoda pravidla 3/8 (jm. prostor `rule38`)
- Dormand-Prince metoda (jm. prostor `dormandprince`)

Každý takový jmenný prostor dané metody definuje typ `CSolver`, který lze přímo využít. Například pro instancování třídy, obstarávající numerické řešení Eulerovou metodou lze psát:

```
ode::euler::CSolver solver();
```

Nad instancí této třídy lze provádět krokování libovolné diferenciální rovnice. Třída neuchovává vnitřní stav, není tedy omezena na řešení právě jedné rovnice.

4.2 Syntetický problém

Pro ověření implementace na syntetických problémech byla implementována malá aplikace *MPMODESolverEval*. Ta má za úkol pro vybrané diferenciální rovnice spustit různě parametrizované Rungovy-Kuttovy metody, jejich průběh vizualizovat a vypočíst absolutní a relativní chybu od přesného analytického řešení, je-li k dispozici.

Výsledné průběhy jsou zaneseny do grafu a je vygenerována sada souborů ve formátu SVG. Vypočtené chyby jsou zapsány do souborů ve formátu CSV.

Více o výsledcích tohoto testování, konkrétních funkcích a podmínkách v kapitole 5.

4.3 Pozměněný Bergmanův model

Tento model je implementován do systému SmartCGMS v rámci integrace popsané v kapitole 4.4. Z implementačního pohledu to s sebou nese podstatné implikace, nicméně z pohledu řešení diferenciálních rovnic některou z popsaných metod jde pouze o výměnu předpisu funkce.

Rovnice v explicitním tvaru jsou implementovány lambda funkcemi:

```
auto dGsc = [&](double _T, double _Gsc) -> double {
    return (((p*G + cg*G*G + c) / (1.0 + cg * G)) - _Gsc) / dtDiff;
};
auto dG = [&](double _T, double _G) -> double {
    return -(p1 + X)*_G + p1 * Gb + p8 * D1 * VgDLInv;
};
auto dX = [&](double _T, double _X) -> double {
    return -p2 * _X + p3 * (I - Ib);
};
auto dI = [&](double _T, double _I) -> double {
    return -p4 * _I + p7 * Isc;
};
auto dIsc = [&](double _T, double _Isc) -> double {
    return -p7 * _Isc + U.total()*ViInv;
};
auto dD1 = [&](double _T, double _D1) -> double {
    return -p8 * _D1 + p9 * D2;
};
auto dD2 = [&](double _T, double _D2) -> double {
    return -p9 * _D2 + Ag * d_ext.Get_Meal_Disturbance(lastTime);
};
```

};

Ty jsou předány jako parametr instanci třídy, která provádí numerické řešení. Rovnice jsou úmyslně při výpočtu seřazeny tak, aby při výpočtu každé z nich byla uvažována hodnota ostatních veličin z předchozí iterace. Pro korektní simulaci je totiž nutné v každém taktu znát správnou sadu signálů v čase t .

4.4 SmartCGMS

Software SmartCGMS je simulační framework a architektura stavěná tak, aby podporovala tzv. kosimulace. Jako kosimulace se označuje taková simulace, ve které figurují simulovaná, reálná a prototypovaná zařízení v jeden moment. Tato vlastnost je klíčovou pro rapidní vývoj nových technologií – programový kód je sjednocen, a je použit stejný základ a postupy jak pro reálná zařízení při měření u samotného pacienta, tak pro *in-silico* testování v rámci simulace.

Do frameworku byla implementována podpora pro již zmíněný pozměněný Bergmanův model a metody pro numerické řešení obyčejných diferenciálních rovnic.

Implementačně jde o entitu typu *device*, která je synchronizována přes zpětnou vazbu – Bergmanův model vypočte hodnoty v čase t a pošle je do architektury. Každá entita v architektuře může hodnoty přečíst, a na základě nich vypočítat další hodnoty. Jakmile hodnoty projdou všemi entitami, zpětná vazba na konci architektury signalizuje synchronizaci do entity obalující Bergmanův model. Nyní se situace opakuje pro další krok simulace a čas $t + 1$.

Z podstaty je tedy dodržena filozofie architektury SmartCGMS – simulace je řízena událostně. Simulace spojitých veličin je diskretizována do pětiminutových úseků, mezi kterými je vždy provedena synchronizace tak, aby systém zůstal v konzistentním stavu.

5 Testování

Pro otestování vlastností implementace byly použity dvě diferenciální rovnice, každá ve více variantách:

1. $\frac{dy(x)}{dx} = -15y$, poč. podmínka $y(0) = 1$
 - (a) $\langle 0; 1 \rangle$, $h = 0.05$
 - (b) $\langle 0; 1 \rangle$, $h = 0.1$
2. $\frac{dy(x)}{dx} = y^2 - y^3$, poč. podmínka $y(0) = 0.01$
 - (a) $\langle 0; 200 \rangle$, $h = 1$
 - (b) $\langle 0; 200 \rangle$, $h = 2$
 - (c) $\langle 0; 200 \rangle$, $h = 3$

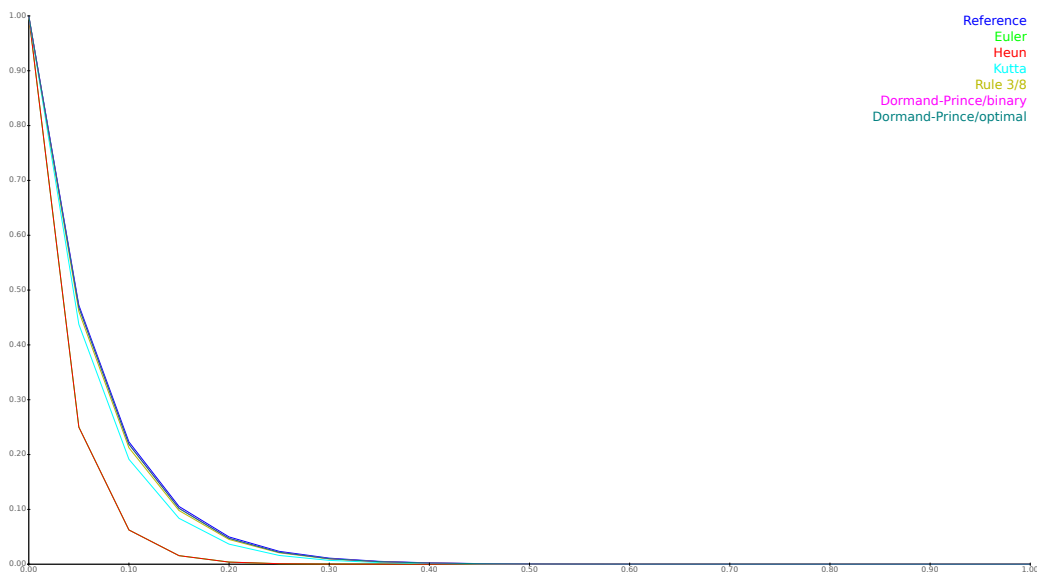
První z těchto rovnic má známé analytické řešení $y(x) = e^{-15x}$ – lze tedy porovnat aproximované hodnoty přímo s referenčními hodnotami. U druhé funkce není známé analytické řešení, ale je známé její chování – funkce je po celou dobu pomalu rostoucí, a v závislosti na zvolené počáteční podmínce se v blízkosti bodu $\frac{1}{y(0)}$ růst rapidně urychlí, než funkce nabyde hodnoty 1. Poté se růst opět zpomalí.

Na obrázku 5.1 je vidět průběh prvního experimentu pro malý krok $h = 0.05$. Při takovém kroku vykazují všechny metody přibližně očekávaný průběh bez větších anomálií. Eulerova a Heunova metoda (na obrázku splývají) je výrazně více vzdálena od referenční křivky. Ostatní metody poskytly uspokojivé výsledky. Oproti tomu druhý experiment, znázorněný na obrázku 5.2, kde byl krok zvětšen na $h = 0.1$, již demonstroval sníženou přesnost metod nižších řádů – Eulerova i Heunova metoda začala kolem přesného řešení oscilovat. Následně se obě ustálily a nakonec zkonvergovaly do stejného bodu, jako metody ostatní ($y(1) \approx 0$).

Experimenty 3 - 5 probíhaly na diferenciální rovnici, jež je považována za tuhou. Byly zvoleny tři velikosti kroků - $h = 1, 2$ a 3 . Experiment s krokem velikosti $h = 1$, znázorněný na obrázku 5.3 je opět pro všechny metody relativně přesný – výsledný průběh připomíná očekávaný výsledek. Ovšem u dvojnásobného kroku, $h = 2$ se situace výrazně mění. Výsledek je znázorněn na obrázku 5.4. Eulerova a Heunova metoda opět oscilují kolem korektní křivky, metody Kutta a pravidla 3/8 se odlonily od očekávaného průběhu.

Metody s adaptivním krokem však zachovávají správný průběh. Obrázek 5.5 pak zobrazuje průběh pro $h = 3$. Zde již začala oscilovat i metoda Kutta, metoda pravidla 3/8 se výrazně odklonila a ustálila se v chybném bodě. Oproti tomu metody s adaptivní velikostí kroku stále zachovávají očekávaný průběh.

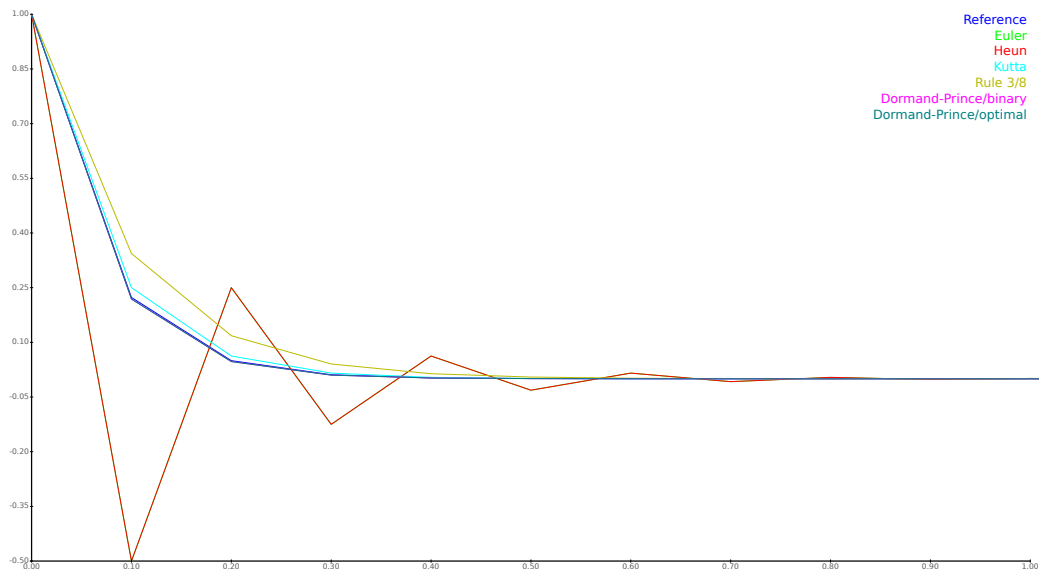
Tabulka 5.1 obsahuje absolutní a relativní chyby. Absolutní chyby vykazují očekávanou tendenci – řád chyby se se zvyšujícím se stupněm metody snižuje. Ovšem relativní chyby vykazují velké anomálie – to z důvodu, že je problém špatně škálovaný pro použití relativní chyby. Velká část hodnot jak exaktního, tak aproximovaného řešení se pohybuje velmi blízko nule. To znamená, že numerická stabilita v těchto částech je velmi malá. Směrodatnou hodnotou je zde tedy absolutní chyba, jelikož je škála hodnot dobře známá.



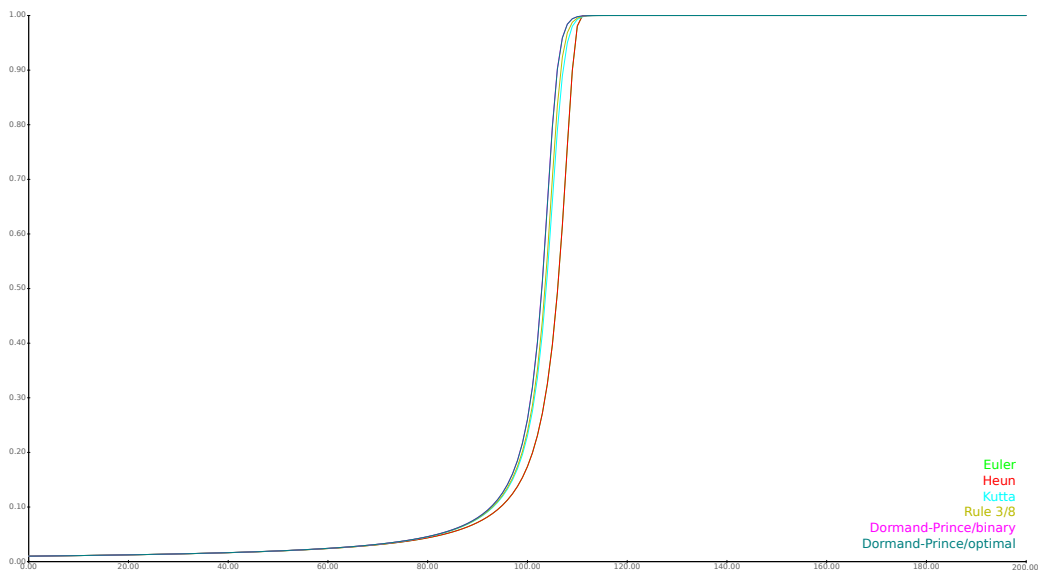
Obrázek 5.1: Rovnice $\frac{dy(x)}{dx} = -15y$, poč. podmínka $y(0) = 1$, interval $\langle 0; 1 \rangle$, $h = 0.05$

Na obrázku 5.6 je dále vidět experiment provedený v prostředí Smart-CGMS s pozměněným Bergmanovo modelem. V prostředí byl zapojen modul pro kalkulaci bolusového inzulínu a příslušné dávky inzulínu bazálního. Byly injektovány 3 časové události v podobě podávaných jídel. Obrázek 5.6 znázorňuje průběh, který odpovídá očekávanému výsledku – různé dávky inzulínu mění průběh glykemie, ovšem kvůli neoptimálnímu dávkování glykemie osciluje. Pokrmu naopak vychylují glykemii směrem do vyšších hodnot. To částečně kompenzuje bolusový inzulín, který je podáván spolu s jídlem.

Rovněž byl implementován prototyp hry, která slouží jako proof-of-concept a bude upravována do finální podoby, ve které bude sloužit pro sběr dat.

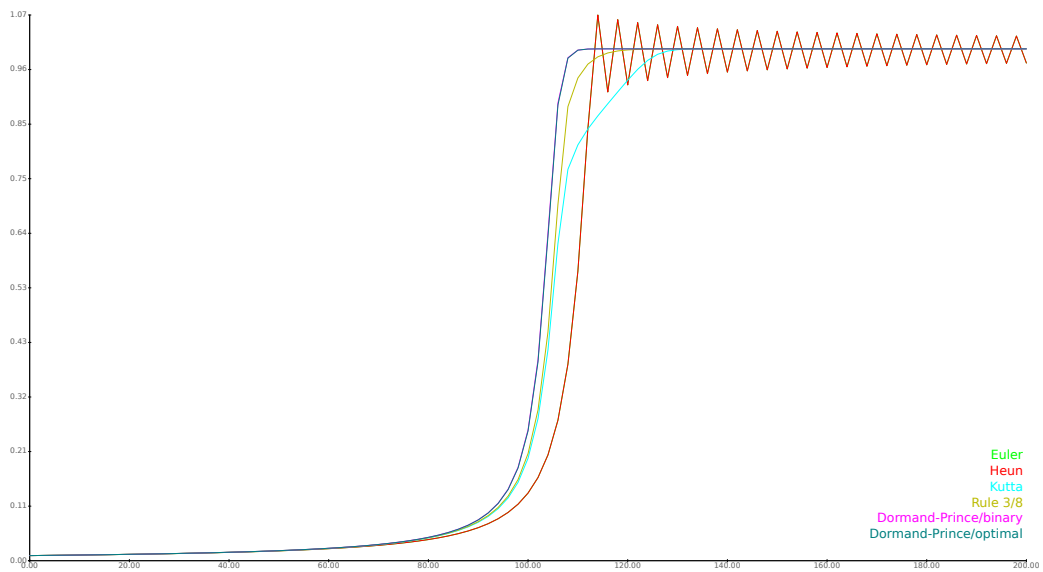


Obrázek 5.2: Rovnice $\frac{dy(x)}{dx} = -15y$, poč. podmínka $y(0) = 1$, interval $\langle 0; 1 \rangle$, $h = 0.1$

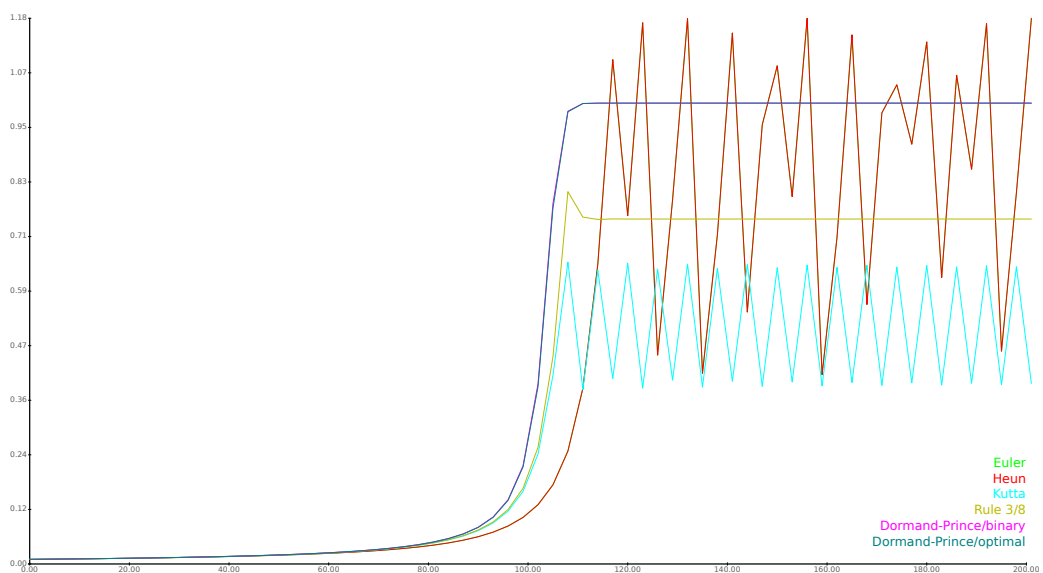


Obrázek 5.3: Rovnice $\frac{dy(x)}{dx} = y^2 - y^3$, poč. podmínka $y(0) = 0.01$, interval $\langle 0; 200 \rangle$, $h = 1$

Motivace byla nastíněna v úvodní kapitole. Obrázek 5.7 znázorňuje převod modelu a simulace do podoby hry.



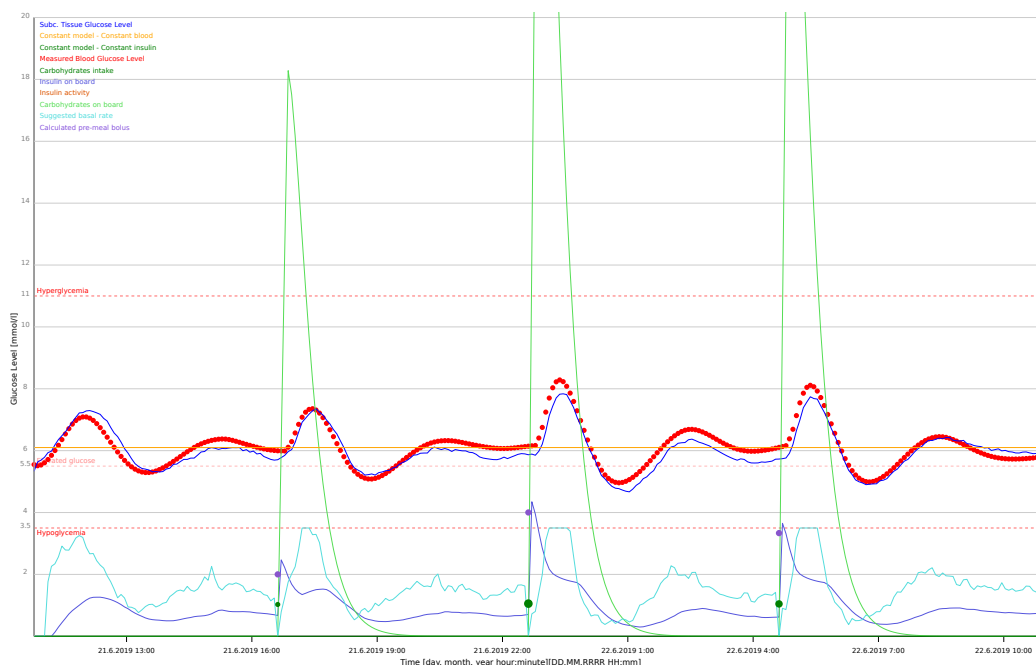
Obrázek 5.4: Rovnice $\frac{dy(x)}{dx} = y^2 - y^3$, poč. podmínka $y(0) = 0.01$, interval $\langle 0; 200 \rangle$, $h = 2$



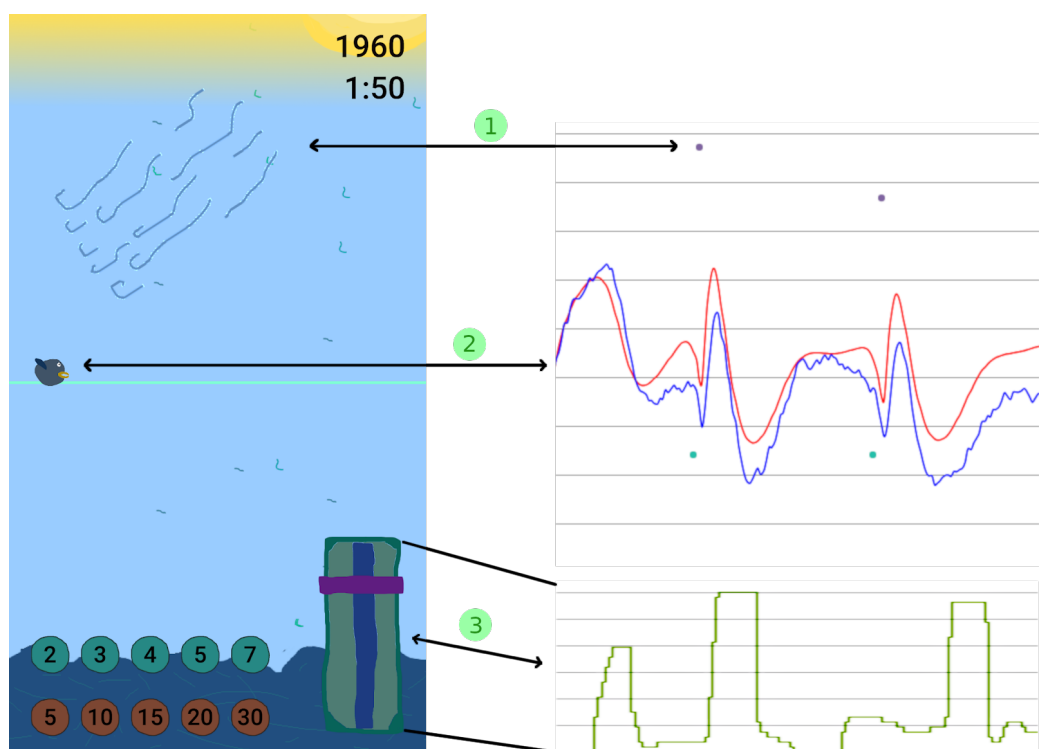
Obrázek 5.5: Rovnice $\frac{dy(x)}{dx} = y^2 - y^3$, poč. podmínka $y(0) = 0.01$, interval $\langle 0; 200 \rangle$, $h = 3$

Exp.	Metoda	Abs. chyba	Rel. chyba	Abs. chyba (koncový bod)	Rel. chyba (koncový bod)
(1)	Euler	0.0268	89.9%	3.0590e-07	99.9%
	Heun	0.0267	89.9%	3.0590e-07	99.9%
	Kutta	0.0055	48.4%	2.3990e-07	78.4%
	Rule 3/8	0.0019	20.9%	1.1847e-07	38.7%
	D.P./bin	0.0008	13.6%	8.1800e-08	26.7%
	D.P./opt	0.0009	13.9%	8.2432e-08	26.9%
(2)	Euler	0.0985	107650%	4.8835e-04	715467%
	Heun	0.0985	107650%	4.8835e-04	715467%
	Kutta	0.0038	101.6%	1.7016e-07	249.3%
	Rule 3/8	0.0197	2639.9%	7.8507e-06	11501.9%
	D.P./bin	0.0007	142.5%	4.0715e-07	596.5%
	D.P./opt	0.0007	143.2%	4.0881e-07	598.9%

Tabulka 5.1: Chyby prvních dvou experimentů (sloupce s abs. a rel. chybou ukazují průměr chyby v celém průběhu)



Obrázek 5.6: Průběh experimentu v prostředí SmartCGMS s implementovaným pozměněným Bergmanovo modelem



Obrázek 5.7: Proof-of-concept mobilní hry pro sběr dat pro odvození modelu optimální inzulinové terapie; (1) převod pokrmů do podoby překážek, (2) převod glykemie do pohybu herní postavy, (3) převod ovládání inzulinové pumpy do podoby ovládání rychlosti mávání křídly herní postavy

6 Závěr

V této práci byly navrženy a provázány dílčí modely, které se podílejí na homeostáze glukózy v lidském těle. Tyto modely zakládají na poznatcích, které představil Bergman a Hovorka. Dále byly implementovány numerické Rungovy-Kuttovy metody pro numerické řešení obyčejných diferenciálních rovnic v pěti variantách. Poslední varianta byla implementována ve dvou provedeních dle strategie určení velikosti kroku a tedy minimalizace výsledné chyby. Celé řešení bylo ověřeno jednak na syntetických problémech dvou diferenciálních rovnic a celkem pěti scénářích, a jednak v prostředí SmartCGMS, které používalo pozměněný Bergmanův model. Rovněž byly modely a příslušné numerické metody využity k vývoji mobilní hry pro sběr dat pro odvození modelu optimální inzulinové terapie.

Dalším pokračováním vývoje bude přizpůsobení parametrů modelu pacienta tak, aby se co nejvíce přiblížil reálným datům, které máme k dispozici od skutečných diabetických pacientů. Takové přizpůsobení implikuje jednak přizpůsobení architektury SmartCGMS tak, aby bylo možné takovou simulaci provádět, a jednak implementaci optimalizačního algoritmu, který bude parametry hledat. Takovým algoritmem může být již implementovaná Meta-diferenciální evoluce (s provedením příslušných úprav jejího kódu), případně jiný algoritmus.

Úpravy do simulátoru SmartCGMS budou k dispozici na portálu diabetes.zcu.cz. Hra získala na Studentské vědecké konferenci 2019 pořádané na Fakultě aplikovaných věd ZČU v Plzni druhé místo a ocenění sponzora IEEE.

Literatura

- [1] International Diabetes Federation. IDF Diabetes Atlas, 8th edn. Brussels, Belgium: International Diabetes Federation, 2017. <http://www.diabetesatlas.org>
- [2] Kasper DL, Fauci AS, Hauser SL, Longo DL, Jameson JL, Loscalzo J. (2015). Harrison's principles of internal medicine (19th edition.). New York: McGraw Hill Education. ISBN: 007174889X
- [3] McAdams BH, Rizvi AA. An Overview of Insulin Pumps and Glucose Sensors for the Generalist. J Clin Med. 2016;5(1):5. Published 2016 Jan 4. doi:10.3390/jcm5010005
- [4] Mastrototaro J. The MiniMed continuous glucose monitoring system (CGMS). J Pediatr Endocrinol Metab, 1999, 12(13), 751-758.
- [5] Steil GM, et al. Feasibility of automating insulin delivery for the treatment of type 1 diabetes. Diabetes, 55 (2006), pp. 3344-3350
- [6] Hovorka R, et al. Closing the loop: the ADICOL experience Diabetes Technol. Ther., 6 (2004), pp. 307-318
- [7] El-Khatib FH, et al. A bihormonal closed-loop artificial pancreas for type 1 diabetes. Sci. Transl. Med., 2 (2010), p. 27ra27
- [8] De Falco I, Cioppa AD, Koutny T, Krcma M, Scafuri U, Tarantino E. Genetic Programming-based induction of a glucose-dynamics model for telemedicine. Journal of Network and Computer Applications, 2018, 119(1), 2018, pp. 1–13. doi: 10.1016/j.jnca.2018.06.007.
- [9] Man CD, Micheletto F, Lv D, Breton M, Kovatchev B, Cobelli C. The UVA/PADOVA Type 1 Diabetes Simulator: New Features. J Diabetes Sci Technol. 2014;8(1):26–34. doi:10.1177/1932296813514502
- [10] Jolanta S, Kosinski W. On Convergence of a Simple Genetic Algorithm. Artificial Intelligence and Soft Computing – ICAISC 2008, 2008, Springer Berlin Heidelberg. pp. 489–498.
- [11] Savic RM, Jonker DM, Kerbusch T, Karlsson MO. Implementation of a transit compartment model for describing drug absorption in pharmacokinetic studies. Journal of Pharmacokinetics and Pharmacodynamics, 2007, 34(5), pp. 711–726. doi: 10.1007/s10928-007-9066-0

- [12] Yamaoka K, Nakagawa T, Uno T. Statistical moments in pharmacokinetics. *Journal of Pharmacokinetics and Biopharmaceutics*, 1978, 6(6), pp. 547–558. doi: 10.1007/BF01062109
- [13] Kemeny JG, Snell JL. *Markov Chains*. Springer-Verlag, New York, 1976. ISBN: 978-0387985091
- [14] Zhao L, Ji P, Li Z, Roy P, Sahajwalla CG. The antibody drug absorption following subcutaneous or intramuscular administration and its mathematical description by coupling physiologically based absorption process with the conventional compartment pharmacokinetic model. *The Journal of Clinical Pharmacology*, 2013, 53(3), 314-325.
- [15] Bergman RN, Ider YZ, Bowden CR, Cobelli C. Quantitative estimation of insulin sensitivity. *The American Journal of Physiology*, 236(6), pp. 667-677, 1979. doi: 10.1152/ajpendo.1979.236.6.E667
- [16] Hovorka R, Canonico V, Chassin LJ, Haueter U, Massi-Benedetti M, Orsini Federici M, Pieber TR, Schaller HC, Schaupp L, Vering T, Wilinska ME. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological measurement*, 25(4), pp. 905-920, 2004. doi: 10.1088/0967-3334/25/4/010
- [17] Bergman RN, Phillips LS, Cobelli C. Physiologic evaluation of factors controlling glucose tolerance in man: measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose. *J Clin Invest.* 1981;68(6):1456–1467. doi:10.1172/jci110398
- [18] Daud NAM, Mahmud F, Jabbar MH. Meal simulation in glucose-insulin reaction analysis using hovorka model towards system-on-chip implementation. *ARPN J. of Eng. and Applied Sci*, 2006, 10(19), 8927-8935.
- [19] Koutny T. Blood glucose level reconstruction as a function of transcapillary glucose transport. *Computers in Biology and Medicine*, 2014, 53, pp. 171 – 178, doi: 10.1016/j.compbiomed.2014.07.017
- [20] Rebrin K, Steil GM. Can interstitial glucose assessment replace blood glucose measurements? *Diabetes Technol. Ther.*, 2 (2010), pp. 461 – 472
- [21] Kobayashi T, Sawano S, Itoh T, Kosaka K, Hirayama H, Kasuya Y. The pharmacokinetics of insulin after continuous subcutaneous infusion or bolus subcutaneous injection in diabetic patients. *Diabetes*, 32 (1983), pp. 331-336

- [22] Wilinska EM, Bodenlenz M, Chassin LJ, Schaller HC, Schaupp LA, Pieber TR, Hovorka R. Interstitial glucose kinetics in subjects with type 1 diabetes under physiologic conditions. *Metabolism*, 2004, 53(11), pp. 1484 – 1491, doi: 10.1016/j.metabol.2004.05.014
- [23] Law AM, Kelton WD. *Simulation modeling and analysis (Vol. 3)*. New York: McGraw-Hill, 2000. ISBN: 978-0073401324
- [24] Lamport L, Melliar-Smith PM. Byzantine clock synchronization. In *Proceedings of the third annual ACM symposium on Principles of distributed computing*, 1984, pp. 68 – 74.
- [25] Fahrland DA. Combined discrete event continuous systems simulation. *Simulation*, 1970, 14(2), 61-72.
- [26] Koutny T, Ubl M. Parallel software architecture for the next generation of glucose monitoring. *Procedia Computer Science*, 2018, 141, pp. 279 – 286, doi: 10.1016/j.procs.2018.10.197
- [27] Verner JH. Explicit Runge–Kutta Methods with Estimates of the Local Truncation Error. *SIAM Journal on Numerical Analysis*, 1978, 15(4), pp. 772 – 790
- [28] Dormand JR, Prince PJ. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 1980, 6(1), pp. 19 – 26, doi: 10.1016/0771-050X(80)90013-3
- [29] Butcher JC. *Numerical Methods for Ordinary Differential Equations*. New York: John Wiley & Sons, 2008, ISBN 978-0-470-72335-7.
- [30] Runge C. *Math. Ann.*, 1895, 46(167). doi: 10.1007/BF01446807
- [31] Atkinson KA. *An Introduction to Numerical Analysis (2nd ed.)*. New York: John Wiley & Sons, 1989. ISBN 978-0-471-50023-0.
- [32] Heun K. Bemerkungen zur Theorie der mehrfach lineär verknüpften Functionen. *Acta Math.*, 1889, 12(1), pp. 103 — 108.
- [33] MathWorks. *Ordinary Differential Equations [online]*. Retrieved from: <https://de.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/odes.pdf>
- [34] Balac S, Fernandez A. Comparison of adaptive step-size control strategies for solving the Generalised Non-Linear Schrodinger Equation in optics by the Interaction Picture method. 2012. hal: hal-00740771f