



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Analýza možností uživatelských funkcí a knihoven v Mathematice
<b>Student:</b>	Jan ezník
<b>Vedoucí:</b>	Ing. Martin Kohlík, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Po íta ové inženýrství
<b>Katedra:</b>	Katedra íslicového návrhu
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

- 1) Analyzujte možnosti vytvoření referenční stránky pro uživatelsky definovanou funkci a návod, jak tuto referenční stránku modifikovat a zavést do standardní nápovědy SW Mathematica.
- 2) Analyzujte, zda Mathematica podporuje import uživatelské knihovny definované ve více souborech. Import by měl probíhat takovým způsobem, že bude importován pouze hlavní soubor knihovny, ve kterém pak budou obsaženy importy všech ostatních potřebných souborů.
- 3) Výše uvedené funkcionality podporované Mathematicou aplikujte na knihovnu určenou pro výpočty hierarchických spolehlivostních modelů pro složité systémy odolné proti poruchám vytvořené vedoucí práce.
- 4) Otestujte knihovnu po výše uvedených úpravách na spolehlivostním modelu, který dodá vedoucí práce. Vypočítejte závislost intenzity poruchy modelovaného systému na dvou parametrech vybraných vedoucí práce.

### Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
ředitel katedry

V Praze dne 6. února 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA ČÍSLICOVÉHO NÁVRHU



Bakalářská práce

## **Analýza možností uživatelských funkcí a knihoven v Mathematice**

*Jan Řezníček*

Vedoucí práce: Ing. Martin Kohlík, Ph.D.

11. května 2017



---

## Poděkování

Rád bych poděkoval Fakultě informačních technologií za možnost studia informačních technologií, pro které mi tato škola přijde jako nejlepší volba. Také bych chtěl zvláště poděkovat svému vedoucímu bakalářské práce, panu doktoru Kohlíkovi, za vstřícnost a ochotu při práci a konzultacích k této bakalářské práci.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 11. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Jan Řezníček. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Řezníček, Jan. *Analýza možností uživatelských funkcí a knihoven v Mathematice*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

## Abstrakt

Obsahem bakalářské práce je návod na tvorbu knihovny pro výpočty v programu Wolfram Mathematica a její dokumentace. Práce také obsahuje příklad použití takto vytvořené knihovny pro výpočet Markovských modelů spolehlivosti.

**Klíčová slova** Wolfram Mathematica, Wolfram Workbench, knihovny, nápověda, Markovské modely spolehlivosti

---

## Abstract

The main topic of the Bachelor Thesis is a guide to create library for computing in Wolfram Mathematica and sample of work with created library for computing of dependability models based on Markov chains.

**Keywords** Wolfram Mathematica, Wolfram Workbench, libraries, help, dependability models based on Markov chains



---

# Obsah

Úvod	1
<b>1 Nápořěda v programu Wolfram Mathematica</b>	<b>3</b>
1.1 Nápořěda pomocí příkazu „?...“	3
1.2 Nápořěda pomocí klávesy F1	3
<b>2 Co je to Wolfram Workbench</b>	<b>7</b>
<b>3 Markovské modely spolehlivosti</b>	<b>9</b>
<b>4 Redukce Markovských modelů</b>	<b>11</b>
<b>5 Tvorba knihoven ve Wolfram Workbench</b>	<b>13</b>
5.1 Základní instalace	13
5.2 Tvorba jednoduchého projektu	15
5.3 Import více knihovních souborů	15
5.4 Tvorba dokumentace a nápořědy	16
5.5 Sestavení projektu a následný export	17
<b>6 Použití knihovny a výsledky práce</b>	<b>21</b>
6.1 1. část výsledků – Závislost na změně intenzity poruchy	22
6.2 2. část výsledků – Závislost na změně intenzity poruchy nouzové metody	23
6.3 3. část výsledků – Závislost na změnách obou parametrů	24
<b>Závěr</b>	<b>27</b>
<b>Literatura</b>	<b>29</b>
<b>A Obsah přiloženého CD</b>	<b>31</b>



---

## Seznam obrázků

1.1	Ukázka nápovědy pomocí příkazu „?...“ . . . . .	4
1.2	Ukázka nápovědy pomocí klávesy F1. . . . .	5
4.1	Ukázka postupu redukce Markovských modelů. . . . .	11
5.1	Ukázka nástroje DocumentationTools. . . . .	18
6.1	Markovský graf použitý v rámci BP. . . . .	21
6.2	Graf závislosti pro změnu parametru $\lambda$ . . . . .	23
6.3	Graf závislosti pro změnu parametru $\gamma$ . . . . .	24
6.4	3D graf výsledných závislostí. . . . .	25



---

## Seznam tabulek

6.1	Tabulka hodnot závislosti při změně parametru $\lambda$ . . . . .	23
6.2	Tabulka hodnot závislosti při změně parametru $\gamma$ . . . . .	24





---

# Úvod

Během studia na Fakultě informačních technologií Českého vysokého učení technického v Praze se studenti seznámí s řadou programovacích jazyků a také s programy, které tyto jazyky podporují a dokáží kontrolovat správnost syntaxe, požadavků a řadu dalších parametrů. Jedním z takových programů je Wolfram Mathematica. Tento program se během bakalářského studia využívá např. v předmětu Číslicové a analogové obvody (BI-CAO) nebo v předmětu Bezpečnost (BI-BEZ). V rámci těchto předmětů jsme se naučili pracovat s programem Wolfram Mathematica na úrovni tvorby funkcí a výpočtů různých fyzikálních a bezpečnostních příkladů. Základem programu je databáze knihoven pro různé použití (např. matematické výpočty, vykreslování grafů, simulace obvodů, ...). Jednotlivé knihovny jsou vytvářeny již od roku 1988. Je však otázkou, jakým způsobem jsou tyto knihovny vytvořeny a je-li možné nějak získat přístup k tvorbě takových knihoven?

Cílem mé bakalářské práce bylo seznámení se s programem Wolfram Mathematica a s jeho možnostmi, které nabízí pro tvorbu knihoven. Je samozřejmé, že pro normálního uživatele tyto rozšířené možnosti nebudou mít až takový význam a tudíž nemusí být součástí hlavního jádra programu. Bylo tedy od začátku jasné, že na tvorbu knihoven budu potřebovat vedlejší program, který mi dá možnost takové knihovny tvořit.

Před samotným začátkem realizace mé bakalářské práce jsem si ověřil, zda se již na fakultě někdo takovým tématem zabýval. Po prostudování dostupných zdrojů jsem zjistil, že tuto problematiku prozatím nikdo neřešil a tvorba knihoven tedy musí začínat úplně od samých základů. Podle zadání bakalářské práce se však využije jen část možností, které bude tvorba knihovny nabízet, a proto jsou v práci popsány jen základní možnosti, jak takovou knihovnu vytvořit. Dalším úkolem bude popis manipulace s novou knihovnou včetně jejího překladu pro použití v programu Wolfram Mathematica. V případě úspěšného vzniku základní knihovny by mohla moje bakalářská práce posloužit jako manuál pro tvorbu knihoven použitelných ve zmíněných předmětech na FIT ČVUT v Praze.



# Nápověda v programu Wolfram Mathematica

Program Wolfram Mathematica je všestranný program pro matematické výpočty všeho druhu a hodí se pro výpočty jakýchkoliv jednoduchých, složitějších ale i velmi rozsáhlých rovnic, dat a příkladů. Pro uživatele, který se s tímto programem setká úplně poprvé, by bylo zajisté velice náročné orientovat se ve všech funkcích, které program používá a ve správnosti dat, které funkce pro svůj výpočet potřebují.

V programu Wolfram Mathematica proto existují k většině naprogramovaných funkcí nápovědy, které uživateli pomohou rychleji se seznámit s danou funkcí a s jejími možnostmi, a ulehčí tak samotnou práci na výpočtech a tím i ušetří čas. Jak může uživatel tyto nápovědy ale najít? Pokud se jedná o tzv. Symbol, což může být jakákoliv speciální proměnná, která je vytvořená pro nějakou specifickou funkci, ale převážně se jedná o název funkce, která umí určitým způsobem řešit nějaký problém, je možné nápovědu získat dvěma způsoby.

## 1.1 Nápověda pomocí příkazu „?...“

Nápověda „?...“ nám zobrazí nápovědu pro danou funkci přímo v daném notebooku. Je potřeba samozřejmě tento příkaz vyhodnotit jako klasické početní úkony, jinak se nám daná nápověda neukáže. Příklad této nápovědy můžeme vidět na Obrázku 1.1.

## 1.2 Nápověda pomocí klávesy F1

Pro zavolání nápovědy pro vybranou funkci pomocí klávesy F1 je potřeba mít nastaven kurzor na jakoukoliv část názvu funkce a až poté klávesu stisknout. V tuto chvíli program otevře vedlejší soubor, který obsahuje nápovědu k dané

**?MCSolve**

MCSolve solves the system of differential equations.  
MCSolve[dir,name,valuesRules,samplesPerDecade] solves the system of differential equations found in <filename>\_AllEqs.mx, a list of names of all states found in <filename>\_Vars.mx, and a list of names of hazard states found in <filename>\_HazardVars.mx located in the specified <directory> relative to the directory of the current evaluation notebook.  
MCSolve[dir,valuesRules,samplesPerDecade] solves all systems (\*\_AllEqs.mx, \*\_Vars.mx, and \*\_HazardVars.mx files) located in the specified <directory> relative to the directory of the current evaluation notebook.  
The numeric values are taken from <valuesRules> list and the result is sampled with uniform distribution on a logarithmic scale with <samplesPerDecade> values per each decade.  
Parameter <samplesPerDecade> may contain a single list of samples or a list of lists. The sampling is preformed using each list of samples separately.  
MCSolve creates the following files in the specified <directory>:  
– <filename>\_SPD<number>\_SolveDuration.mx – a list containing the duration of the solution of the system and the duration of the sampling using <number> samples per decade,  
– <filename>\_SPD<number>\_FDFTable.mx – the failure distribution function sampled using <number> samples per decade.  
A separate set of files is created for each solved system (<filename>\_AllEqs.mx file).

Obrázek 1.1: Ukázka nápovědy pomocí příkazu „?...“.

funkci. Pokud funkce takovouto nápovědu nemá, vyhledávač ji nenalezne a ohlásí chybu. V rámci této nápovědy se objeví většinou stejný základní popis, jako je v případě nápovědy pomocí znaku „?...“. Dále pak tato nápověda může obsahovat rozsáhlejší popis funkce, příklady volání funkce s různými parametry (i s použitím jiných funkcí, např. pro vykreslení grafu sinusovky – „Plot[Sin[x], x, 0, 2 Pi]“). Pokud existují nějaké funkce, které jsou s touto funkcí příbuzné nebo je tato funkce obsažena v některé referenční manuálové nápovědě, může tato nápověda obsahovat i hypertextové odkazy, které směřují právě na tyto nápovědy. Příklad této nápovědy můžeme vidět na Obrázku 1.2.

V případě hlavní referenční stránky bohužel nelze použít nápovědu znakem „?...“, protože se nejedná o funkci nebo speciální symbol a většinou se název knihovny neshoduje s názvem funkce. Pro referenční stránku lze proto využít pouze nápovědy po stisku tlačítka F1. U referenční stránky nápověda po stisku tlačítka F1 funguje stejně jako v popisu výše.

# Sin

**UPDATED**  
show changes

`Sin[z]`  
gives the sine of  $z$ .

▼ Details

▼ Background & Context

▲ Examples (51)

▲ Basic Examples (4)

The argument is given in radians:

In[1]:= `Sin[Pi / 3]`

Out[1]=  $\frac{\sqrt{3}}{2}$

---

Use `Degree` to specify an argument in degrees:

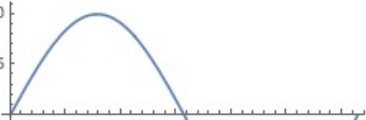
In[1]:= `Sin[60 Degree]`

Out[1]=  $\frac{\sqrt{3}}{2}$

---

Plot over a subset of the reals:

In[1]:= `Plot[Sin[x], {x, 0, 2 Pi}]`

Out[1]= 

Obrázek 1.2: Ukázka nápovědy pomocí klávesy F1.



## Co je to Wolfram Workbench

Program Wolfram Mathematica byl vytvořen a vydán ve verzi 1.0 poprvé v roce 1988. Jedná se o obecný systém pro výpočty matematických úloh. Program lze využít jako klasickou kalkulačku, ale v jeho databázi jsou vytvořeny tzv. „knihovny“, které umějí vypočítat složitější vzorce, vykreslit grafy a jejich využití se v přírodních vědách velmi uplatňuje[1].

Do dnešního dne vývojáři postoupili již velice daleko (nejnovější verze programu Wolfram Mathematica je 11.0) a samozřejmě se rozšířily možnosti výpočtů a také knihoven funkcí, které lze v programu využívat. Samozřejmě i tento program nemůže být naprosto všeobecný, a proto vývojáři postupně upravují a vylepšují samotný program, který ovšem nemusí pořád obsahovat všechny funkce, které uživatel může potřebovat a chce využívat. Proto při práci na programu Wolfram Mathematica dali vývojáři možnost uživatelům, aby si mohli vytvářet své vlastní funkce, které pro práci v programu potřebují. Z tohoto důvodu je vytvořen Wolfram Workbench.

Wolfram Workbench[2] je plugin do programu Eclipse, kde může uživatel vytvářet své vlastní knihovny, které si může libovolně upravovat a poté s nimi pracovat v samotné Mathematice. Samotná práce na tvorbě knihoven je v souladu se syntaxí programu Mathematica, a proto se nový uživatel musí držet striktních pravidel, které používají vývojáři Wolframu, aby mohl danou knihovnu správně vytvořit a následně používat.

Používané knihovny mohou obsahovat více druhů souborů. Vedle základního zdroje knihovny, kde jsou uloženy funkce pro výpočty, jsou zde ještě soubory nápověd a manuálových stránek, které pomáhají ulehčit práci s danou knihovnou. Pokud nově vytvářenou knihovnu bude uživatel používat sám, nemusí se o tyto nápovědy starat, jelikož většinu svého zdrojového kódu zná a ví, co jaká funkce umí a má umět. Za předpokladu, že s danou knihovnou hodlá pracovat i někdo jiný, je téměř nemožné se bez těchto nápověd a manuálů orientovat v knihovně a vědět, jaké funkce obsahuje a co dokáže dané funkce spočítat.

Plugin Wolfram Workbench je velmi kvalitně graficky zpracován a orien-

## 2. CO JE TO WOLFRAM WORKBENCH

---

tace v něm nezabere pro úplného nováčka tolik času. Pro zjednodušení práce samozřejmě pro tento plugin existuje referenční manuál[3], který je celkem přehledný a lze v něm snadno nalézt, co vše má uživatel pro správnou funkčnost udělat.



## Markovské modely spolehlivosti

Kritické systémy s garantovanou úrovní bezpečnosti a spolehlivosti se využívají v mnoha odvětvích (letecký průmysl, medicína, kosmonautika, železniční spojení, ...). Jsou to takové systémy, při jejichž selhání by mohly nastat fatální následky jak pro člověka, tak pro okolní prostředí. Proto se musí při jejich výrobě dbát veliký důraz na kontrolu a testování. Bohužel i v takovýchto situacích nemusí být systém dostatečně odolný, a proto je velice důležité umět odhadnout dobu, po kterou dané systémy budou správně fungovat a také určit v jaké situaci již je potřeba daný systém buďto opravit, nebo ho nahradit novým.

Pro výpočty doby, po kterou bude systém funkční lze využít např. Markovské modely, které slouží k modelování systémů s náhodnými příchody požadavků (lidé ve frontě, půjčovny aut, vozidla v křižovatce), ale i poruchy systému apod. Takovéto modelování systémů má jasně určené stavy (stejně jako u konečného automatu, ve stavech jsou určeny všechny vlastnosti systému) a dále také přechody (známe jejich intenzitu/četnost). Tyto modely mají tu vlastnost, že pravděpodobnost přechodů mezi jednotlivými stavy má exponenciální rozdělení a pravděpodobnost určitého stavu závisí pouze na předchozím stavu a ne na cestě do něj[4].

V rámci Markovských modelů spolehlivosti je možné zavést tzv. Markovský řetězec. Markovský řetězec je speciální náhodná posloupnost, kde pravděpodobnost, že člen  $X_k$  posloupnosti nabude určitou hodnotu, je ovlivněna pouze hodnotou předchozího členu posloupnosti  $X_{k-1}$ . V uvažovaných aplikacích představují hodnoty členů  $X_k$  čísla stavů. Jejich definičním oborem je například množina přirozených čísel  $E = \{1, 2, 3, \dots, n\}$ [5].

Pro sestavení Markovského modelu se nejprve musí definovat vzájemně se vylučující stavy soustavy. Soustava Markovských stavových rovnic popisuje pravděpodobnostní přechody z počátečních stavů do konečných stavů a to za předpokladů:

1. Pravděpodobnost přechodu z jednoho stavu do jiného v časovém inter-

### 3. MARKOVSKÉ MODELY SPOLEHLIVOSTI

---

valu  $\delta t$  je rovna součinu  $\lambda_i(t) * \delta t$ , kde  $\lambda_i(t)$  je intenzita pravděpodobnosti přechodu příslušná ke dvěma stavům, mezi kterými přechod probíhá.

2. Pravděpodobnosti více než jednoho přechodu v časovém intervalu  $\delta t$  jsou řádově menší a lze je tedy zanedbat.

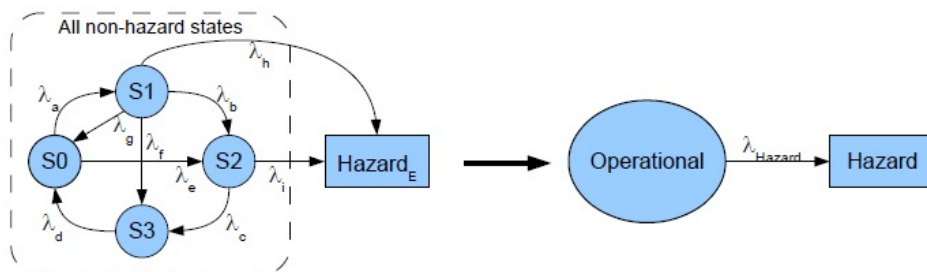
Tyto modely můžeme samozřejmě znázornit jako matici pravděpodobností přechodu, kde řádek značí počáteční stav přechodu, sloupec značí koncový stav přechodu a hodnota v daném řádku a sloupci je pravděpodobnost přechodu mezi dvěma stavy, ale i orientovaným grafem (tzv. Markovský graf), kde uzly grafu představují stavy dané soustavy, orientované hrany grafu označené pravděpodobnostmi přechodu udávají možné přechody[5, 6].

## Redukce Markovských modelů

Zjednodušené modely jsou tvořeny použitím tzv. *redukce*, která vytvoří aproximovaný Markovský model tak, že sloučí všechny nehazardní stavy obecného Markovského modelu do jednoho stavu, který je nazýván v práci jako tzv. „Operational“ (na Obrázku 4.1). Při výpočtu intenzity poruchy je sloučení možné, protože není nutné ve výpočtu rozlišovat mezi nehazardními stavy. Zjednodušený model obsahuje novou  $\lambda_{Hazard}$ , což je míra rizika nahrazující všechny intenzity poruchy v daném modelu[7].

V rámci bakalářské práce se pro tvorbu knihovny a výpočty Markovských modelů poruch použily zdrojové soubory dodané vedoucím bakalářské práce, panem doktorem Kohlíkem. Pro samotné výpočty v notebooku jsou nejdůležitější dvě funkce:

- MCSolve
- MCRReduce



Obrázek 4.1: Ukázka postupu redukce Markovských modelů.

#### 4. REDUKCE MARKOVSKÝCH MODELŮ

---

Redukce modelů probíhá v těchto krocích:

1. Vypočítat přesnou distribuční funkci selhání  $F_E(t)$  pomocí vyřešení systému diferenciálních rovnic korespondujících s daným modelem.  
Tento výpočet provádí funkce `MCSolve`.
2. Najít validní hodnotu pro  $\lambda_{Hazard}$  – nejnižší intenzitu poruchy, u které je funkce  $F_R(t)$  pesimistická, tj. v každém časovém okamžiku je větší, nebo rovna přesné funkci  $F_E(t)$  z prvního bodu.  
Tento výpočet provádí funkce `MCRReduce`.

---

# Tvorba knihoven ve Wolfram Workbench

Program Wolfram Mathematica má většinu základních funkcí, které uživatelé využijí pro práci, již v sobě. Ovšem tvorba složitějších funkcí z menších je velmi zdoluhavá a v případě, že jsou tyto funkce počítány rovnou v rámci notebooku Mathematica, zabírají spoustu místa a tím i znehledňují celý notebook. Samozřejmě pak nastává problém v hledání chyb, když výsledek programu je jiný, než uživatel očekával. Z tohoto důvodu je nejlepším řešením vytvoření vlastních funkcí, které budou mít tyto výpočty v sobě a tím dojde k zjednodušení a zpřehlednění výsledného notebooku nejen pro vývojáře, který funkce vytvořil, ale i pro další uživatele, kteří by se s knihovnou setkali a chtěli ji použít. Je možné takovouto knihovnu vytvořit bez pluginu Wolfram Workbench, ovšem práce je to zdoluhavá a tvůrce nemůže použít tuto knihovnu jinde, než kde si jí sám takto vytvořil.

Plugin Wolfram Workbench umožňuje zjednodušeně vytvořit knihovnu, kterou lze poté exportovat přímo do programu Mathematica a tím pádem může uživatel funkce z nové knihovny použít kdykoliv během své práce.

## 5.1 Základní instalace

První částí pro práci na tvorbě nových knihoven do programu Wolfram Mathematica je samozřejmě instalace samotného programu Wolfram Mathematica. Ten lze stáhnout z oficiálních stránek firmy Wolfram[8].

Pokud se jedná o běžného uživatele, který nemá zajištěnou licenci, bude program možné používat jako tzv. „15-day trial“ verzi. Pro získání licence je možné si zjistit informace na této oficiální stránce. V rámci studia na Fakultě informačních technologií ČVUT v Praze je přístup k licenci zajištěn, a proto se situace s licenci nemusí řešit. Dále je důležitou podmínkou pro správnou funkčnost verze programu Wolfram Mathematica alespoň 6.0 nebo vyšší (pro

nižší verze není zaručena správná funkčnost všech nástrojů).

Dalším programem, který je pro tvorbu knihoven do Mathematicy vyžadován, je program Eclipse Neon. Tento program je tzv. *open source*, takže jeho stažení a funkčnost není nijak limitována licencemi nebo zpřístupňováním plné verze. Tento program je možné stáhnout z oficiálních stránek Eclipse[9], ovšem nutnou podmínkou funkčnosti je verze programu 4.6 nebo vyšší.

Pro správnou funkčnost programu Eclipse je také potřeba mít aktualizovanou verzi Javy (je potřeba verze JDK 8 nebo vyšší).

Samotná instalace pluginu probíhá v těchto jednoduchých krocích:

1. Spustit program Eclipse, v záložkách kliknout na volbu **Help -> Install New Software**.
2. Vedle položky **Work with:** kliknout na tlačítko **Add...**
3. V nově otevřeném okně zadat do položky **Name:** **Wolfram Workbench** a do položky **Location:** zadat **http://workbench.wolfram.com/update/** -> kliknout na tlačítko **OK**.
4. Nyní je možné vybrat si ze dvou položek - **Wolfram Workbench Core** a **Wolfram Workbench webMathematica** (Pro účely bakalářské práce byla zvolena varianta **Wolfram Workbench Core**).
5. Kliknout na tlačítko **Next**, souhlasit s licenčními podmínkami, pak kliknout na **Finish**.

Nyní již probíhá samotná instalace pluginu Wolfram Workbench. V průběhu instalace mohou nastat bezpečnostní varování o neznámém obsahu, ale podle oficiálních informací se nejedná o nic vážného a můžeme tedy kliknout na tlačítko **OK**.

Po dokončení instalace je nutné restartovat program Eclipse. V tuto chvíli je již plugin Wolfram Workbench nainstalován, je nutné jej pouze ověřit. Při otevření okna přes tlačítka **Help -> About Eclipse** je potřeba zvolit znak Wolfram Workbench (znak podobný červenému trojúhelníku) a dále kliknout na tlačítko **OK**.

Pro správnou práci nového pluginu je potřeba ještě Wolfram Workbench nakonfigurovat:

1. Otevřít Eclipse Preferences (Windows/Linux: **Window -> Preferences**, Mac: **Eclipse -> Preferences**).
2. Vybrat v levé liště kategorii **Wolfram**, následně se zobrazí možné verze programu Wolfram Mathematica.
3. Vybrat z preferované verze Mathematicy, pro kterou se budou knihovny vytvářet, nakonec kliknout na tlačítko **OK**.

Detailnější návod, ve kterém jsou například výběry jiných položek, je možné zjistit na stránkách pluginu Wolfram Workbench[2].

## 5.2 Tvorba jednoduchého projektu

1. Kliknout v horním menu na položku `File -> New -> Project`.
2. Rozkliknout složku Wolfram, ve které se nachází tři verze projektů - `Application Project`, `Basic Project` a `JLink Project` (pro účel bakalářské práce byl použit pouze `Application Project`).
3. Vytvořit jméno příslušného projektu (tento název není nikterak důležitý, slouží pouze k orientaci mezi jednotlivými projekty uživatele), popřípadě nastavit, jestli se má projekt vytvořit v základním umístění programu Eclipse, nebo jinde, popřípadě vypsát správnou cestu pro vytvoření.
4. Vybrat projekt/-y, na které se může nově vytvářený projekt odkazovat (pro účel bakalářské práce byl vytvářen nový prázdný projekt bez odkazů).
5. Do kolonky `Application name`: zadat název knihovny, kterou vytváříme (ten je již závazný, po exportu do Mathematicy se tento název bude používat pro zavolání a načtení celé knihovny).
6. Pro možnost vytváření manuálových stránek a nápověd je potřeba zaškrtnout políčko `Create PacletInfo.m` a poté ještě políčko `Create documentation`.
7. Kliknout na tlačítko `Finish` a tím je projekt vytvořen.

## 5.3 Import více knihovních souborů

Při tvorbě projektu může nastat problém, že jedna knihovní složka může obsahovat příliš mnoho funkcí a tím ztížit čitelnost kódu a případné opravy v něm. Je proto vhodné vytvořit více knihovních souborů, které posléze mohou být načteny společně pomocí jednoho inicializačního souboru. Ve stejné složce, kde se nachází první soubor `název_knihovny/název_knihovny.m`, se vytvoří nové knihovní složky, které ovšem nemohou mít stejný název, nejlépe se hodí pojmenování takové, aby již podle názvu bylo zřejmé, o jakou práci se tato knihovna bude starat a jaké funkce v ní jsou obsaženy. Takto vytvořená knihovna je sice součástí projektu, ovšem pokud budeme chtít knihovnu inicializovat v rámci volání celé knihovny, kterou jsme v projektu vytvořili, je potřeba tuto knihovnu zavolat v rámci souboru `init.m`. Soubor `init.m` je inicializační soubor, který při zavolání knihovny metodou

« `název_knihovny` » v programu Wolfram Mathematica spustí tento inicializační soubor, který si načte všechny podpůrné soubory, které pro správnou funkčnost potřebuje. Soubor `init.m` v tuto chvíli obsahuje pouze příkaz `Get["Název_knihovny\Název_knihovního_souboru"]`. Tento příkaz je tedy potřeba nakopírovat a `Název_knihovního_souboru` nahradit názvem nové knihovny, která se do projektu vložila. Takto je možné v projektu mít několik různých knihovnických souborů, které jsou součástí jedné velké knihovny a je možné používat všechny najednou. Může nastat problém inicializace knihoven, jelikož některé knihovny pro své funkce mohou využívat funkce z jiných souborů. Proto je třeba klást důraz na pořadí volajících funkcí `Get` v závislosti na tom, jaké knihovny jsou takto propojeny. V opačném případě se projekt nedokáže v budoucnu zkompileovat a nebude možné ho používat.

### 5.4 Tvorba dokumentace a nápovědy

Při tvorbě knihovny je velmi užitečné vytvořit i dokumentaci a nápovědu a to jak k samotné knihovně, tak i ke všem funkcím, které knihovna obsahuje. Pokud si je uživatel jistý, že dané knihovně bude vždy rozumět, je možné tento krok v tvorbě vynechat a případně ho doplnit později.

Dokumentace zdrojového kódu je možné samozřejmě psát jako komentáře přímo do knihovnického souboru, kde jsou jednotlivé funkce uloženy. V případě, že uživatel chce vytvořit nápovědu, která se po exportu bude volat jako příkaz „?...“, musí do knihovnického souboru, kde je daná funkce vytvořena, přidat příkaz `název_funkce::usage = "obsah_nápovědy"`; . Takto vytvořená nápověda se potom vyvolá jako klasické vyhodnocení nějakého příkladu v notebooku Mathematica.

Pokud se jedná o tvorbu uživatelského manuálu ke knihovně nebo tvorby nápověd po stisku tlačítka F1, je potřeba nejdříve postupovat takto:

1. Rozkliknout v levém horním rohu složku projektu, nalézt a otevřít soubor `PacletInfo.m`.
2. Ve spodní liště rozkliknout záložku `Documentation`.
3. Vybrat si z navrhovaných složek, jakou nápovědu nebo dokumentaci chce uživatel vytvořit.

Obrazovka nyní zobrazuje tři složky - `Tutorials`, `Guides` a `Symbols`

- `Tutorials`

Tato složka může obsahovat manuálovou stránku knihovny a její standardní práci, krok za krokem. Jelikož v rámci zadání bakalářské práce není potřeba vytvářet tutoriál pro samotné knihovní funkce, ale pouze popsat návod, jak takovýto návod je možné vytvořit, zůstala tato složka prázdná.



- **Guides**  
Tato složka může obsahovat manuálovou stránku se základním popisem celé knihovny a v případě nutnosti i s odkazy na jednotlivé funkce, které knihovna obsahuje.
- **Symbols**  
Tato složka obsahuje odkazy na všechny funkce a proměnné ve funkcích, které jsou naimplementovány v dané knihovně a byla by tedy možnost, že budou nápovědy potřeba. Pokud není žádná funkce implementovaná, je tato složka prázdná. V opačném případě se u každého symbolu v této složce zobrazí jedna ze dvou značek:
  - $\times$  značí, že pro danou funkci nebo proměnnou ještě nebyla vytvořena manuálová stránka.
  - Jiný znak (není přesně definován) značí, že pro danou funkci je nápověda již vytvořena.

Pokud uživatel požaduje vytvoření jakékoliv stránky a to v jakékoliv složce, stačí na danou složku kliknout a dále kliknout na tlačítko **Add Page**. Program daný soubor vytvoří a ihned po tvorbě otevře, aby jej mohl uživatel upravit. Následně pak má tvůrce možnost zvolit si hlavní stránku, kterou knihovna otevře, pokud se na ní bude uživatel v programu Mathematica ptát. To se provádí vybráním určitého souboru a kliknutím na tlačítko **Set Main**.

Při práci v samotném notebooku, kde se otevřela nová nápověda nebo referenční stránka, může využívat základní syntaxe jako v programu Mathematica. Vývojáři ale navíc pro tvorbu těchto stránek vytvořili nástroj pro ulehčení - tzv. **DocumentationTools** (viz Obrázek 5.1). Ten se nachází v horní liště pod tlačítkem **Palletes**.

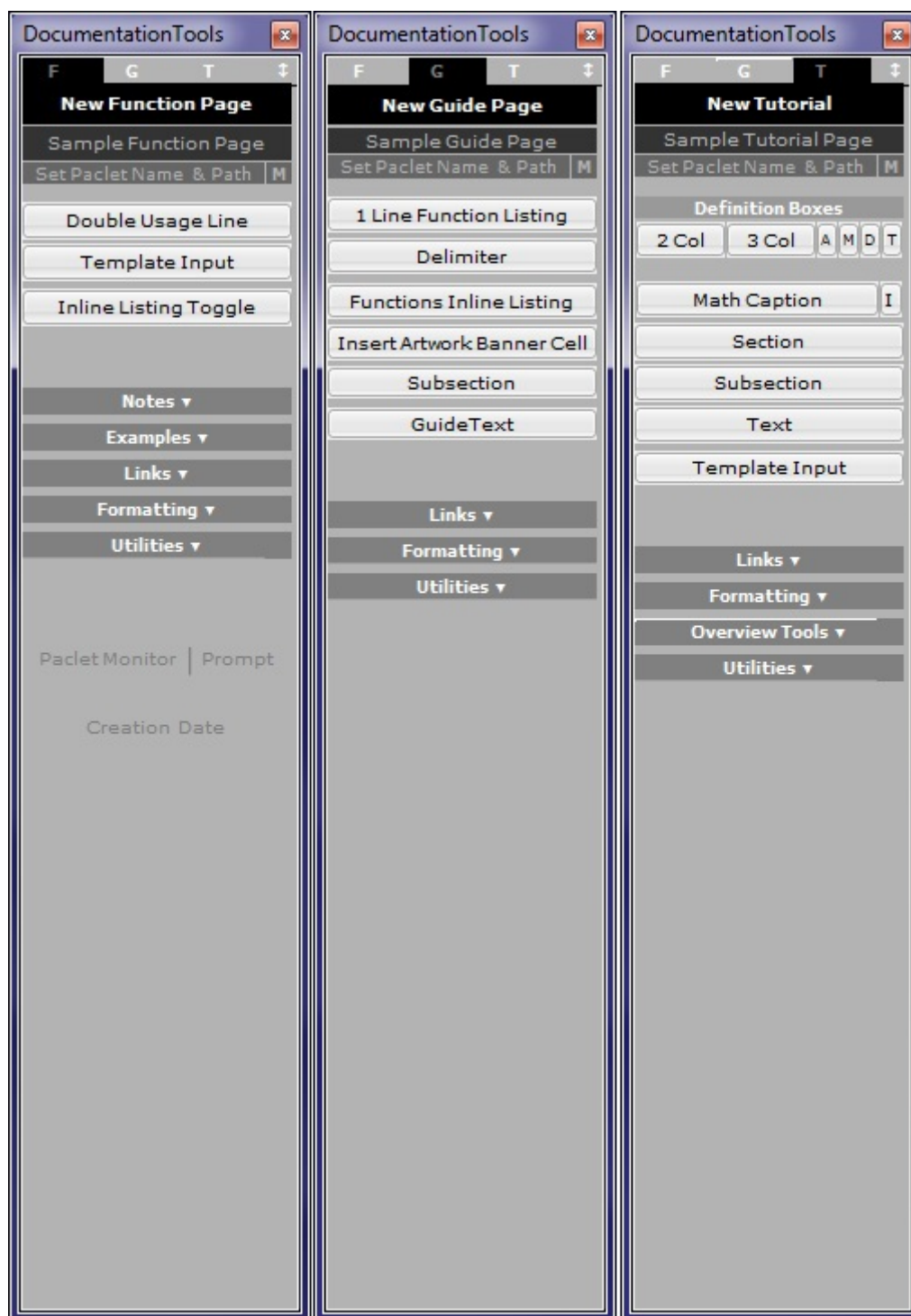
## 5.5 Sestavení projektu a následný export

Ve chvíli, kdy je knihovna dokončena, nastává další část tvorby - sestavení projektu. To se skládá z těchto jednoduchých kroků:

1. Rozkliknutí záložky **Application Tools** v levé dolní části obrazovky, zobrazí se okno sestavování projektu.
2. V políčku **Project** vybrat název projektu, který se bude sestavovat.
3. Kliknutí na tlačítko **Build**.

Program začne vytvářet celkové propojení všech souborů v knihovně, což jsou zdrojové soubory, kde jsou připravené funkce, dále pak referenční stránka knihovny a všechny manuálové stránky a nápovědy, které se pro daný projekt vytvořily. Po úspěšném sestavení projektu se modrým písmem v okně **Console**

## 5. TVORBA KNIHOVEN VE WOLFRAM WORKBENCH



Obrázek 5.1: Ukázka nástroje DocumentationTools.

vypíše celkový čas sestavení projektu. Pokud se v průběhu zpracování vyskytne chyba, program ji červeně vypíše a je tedy potřeba daný projekt upravit před dalším sestavováním.

Samotný export projektu probíhá v následujících krocích:

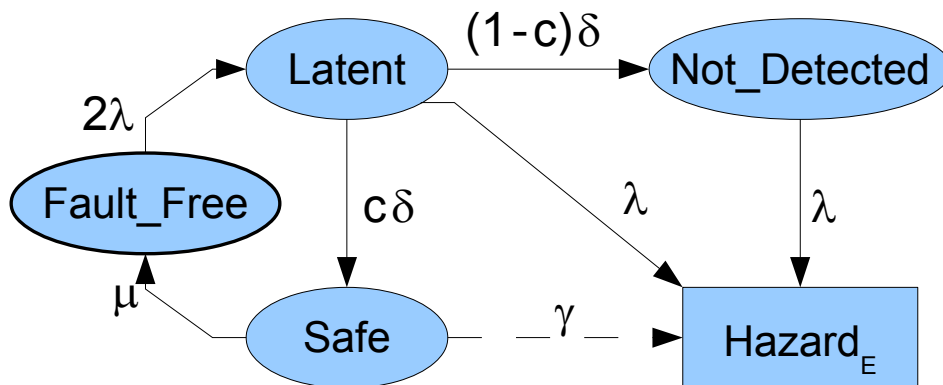
1. Kliknutí na položku **File** -> **Export...**,
2. Rozkliknutí složky **Wolfram** a následná volba mezi **Application Archive** a **Application Folder**
  - **Application Archive**  
Vyexportuje projekt do zkomprimovaného souboru, který se může sdílet velice jednoduše mezi další uživatele, kteří danou knihovnu potřebují využívat.
  - **Application Folder**  
Vyexportuje projekt do složky, kde se nachází knihovny programu Wolfram Mathematica a po exportu tak bude možné knihovny rovnou využívat na daném zařízení.
3. Vybrat z nabídky složku projektu, který se má exportovat, v případě výběru jednotlivých souborů lze použít záložku **Manual select** (pro bakalářskou práci se manuální výběr nepoužil).
4. Vybrat místo, kam se archiv/složka má uložit (v případě archivu je vhodné vybrat umístění, které není pro uživatele příliš těžce dohledatelné, pro složku program vyhledá umístění zdrojových souborů knihoven Mathematicy a toto pole již není tedy potřeba měnit).
5. Zvolit, jestli je požadováno danou knihovnu vyexportovat i s vytvořenou dokumentací, poté již kliknout na tlačítko **Finish**.

Archiv knihovny lze použít jak na platformě MS Windows, tak na dalších platformách (Linux, ...). Dále je důležité při stahování knihovny jiným uživatelem, aby věděl, kde se nachází zdrojové soubory jeho programu Mathematica a správně složku zkopírovat. Jinak knihovna nebude správně fungovat.



## Použití knihovny a výsledky práce

V rámci bakalářské práce je použit Markovský model spolehlivosti vytvořený vedoucím bakalářské práce, panem doktorem Kohlíkem, který je vidět na Obrázku 6.1



Obrázek 6.1: Markovský graf použitý v rámci BP.

Stručný popis Markovského grafu[7]:

- **Fault\_Free** je funkční/bezchybný stav bloku. Míra rizika první poruchy je  $2\lambda$ , protože první porucha může mít vliv na libovolný ze dvou bloků modelu.
- **Latent** stav je aktivní, když blok obsahuje poruchu, která není zatím detekována. Intenzita on-line testu (inverze průměrného zpoždění mezi vznikem poruchy a jejím detekováním) je označena jako  $\delta$ . Pokud test dopadne úspěšně (porucha je detekována), takovýto blok je uzamčen do tzv. **Safe** stavu. Pravděpodobnost úspěchu testu je označena jako  $c$ .

- Pokud test nedopadne úspěšně (porucha nastala, ale nebyla detekována), přejde blok do stavu `Not_Detected`. Bezpečnost bloku v tomto stavu není porušena, ale další chyba (s mírou rizika  $\lambda$ ), která zasáhne dosud nedotčený modul, způsobí selhání zařízení (`Hazard_E` stav). Druhá chyba, která zasáhne již zasažený modul, nemusí být považována za hazard, protože druhý (nedotčený) modul pracuje správně.
- Hrana vedená ze stavu `Latent` do stavu `Hazard_E` vyjadřuje pravděpodobnost, že druhá chyba zasáhne dosud nedotčený modul před tím, než je test dokončen.
- Blok uzamčený ve stavu `Safe` čeká, dokud nebude dokončena oprava (míra opravy je značena jako  $\mu$ ). Blok není v tomto stavu funkční, ale bezpečnost není porušena.
- Funkcionalita bloku bude poskytnuta záložními/nouzovými metodami (např. lidským faktorem), pokud je daný blok ve stavu `Safe`. Intenzita  $\gamma$  vyjadřuje intenzitu poruchy této nouzové metody (např. selhání lidského faktoru).

Výchozí hodnoty pro pravděpodobnost detekce chyby, tzv. intenzitu poruchy, a ostatní parametry použité v této práci:

$$\mu = 24^{-1} [h^{-1}] - \text{intenzita opravy}$$

$$\lambda = 10^{-5} [h^{-1}] - \text{intenzita poruchy}$$

$$\delta = 10^{-1} [h^{-1}] - \text{intenzita on-line testu}$$

$$c = 0.6 - \text{pravděpodobnost úspěchu on-line testu}$$

$$\gamma = 10^{-3} [h^{-1}] - \text{intenzita poruchy nouzové metody}$$

V rámci bakalářské práce byly vypočteny závislosti intenzity selhání celého bloku na změně:

1. Intenzity poruchy  $\lambda$ .
2. Intenzity poruchy nouzové metody  $\gamma$ .
3. Obou parametrů  $\lambda$  i  $\gamma$ .

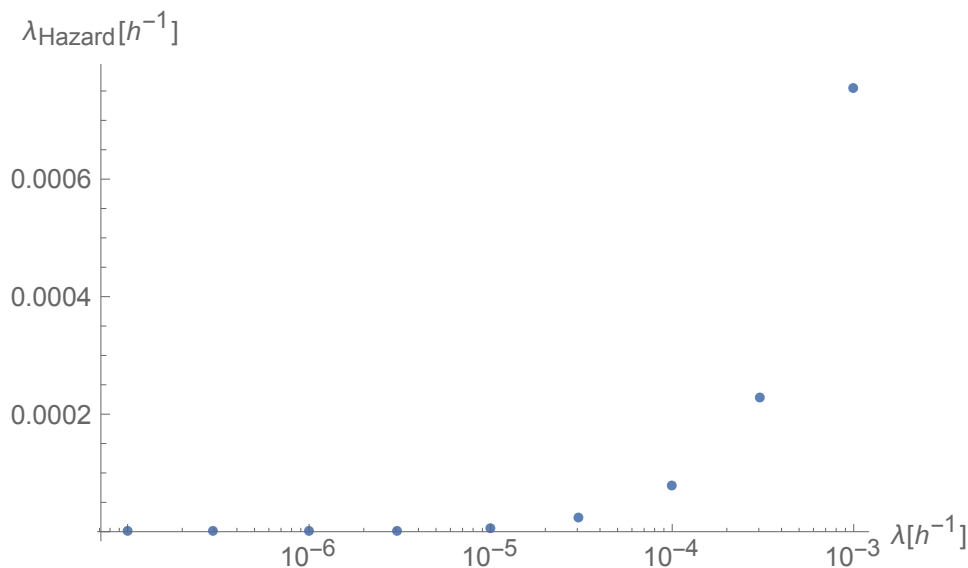
## 6.1 1. část výsledků – Závislost na změně intenzity poruchy

V této části byly použity všechny výchozí hodnoty kromě  $\lambda$ , která se měnila v rozpětí od  $10^{-3}$  do  $10^{-7}$  (Tabulka 6.1 a Obrázek 6.2).

6.2. 2. část výsledků – Závislost na změně intenzity poruchy nouzové metody

$\lambda [h^{-1}]$	$\lambda_{Hazard} [h^{-1}]$
$10^{-3}$	0.000755295
$3 * 10^{-4}$	0.000229894
$10^{-4}$	0.0000768688
$3 * 10^{-5}$	0.0000230837
$10^{-5}$	$7.70162 * 10^{-6}$
$3 * 10^{-6}$	$2.30968 * 10^{-6}$
$10^{-6}$	$7.70305 * 10^{-7}$
$3 * 10^{-7}$	$2.3098 * 10^{-7}$
$10^{-7}$	$7.70326 * 10^{-8}$

Tabulka 6.1: Tabulka hodnot závislosti při změně parametru  $\lambda$ .

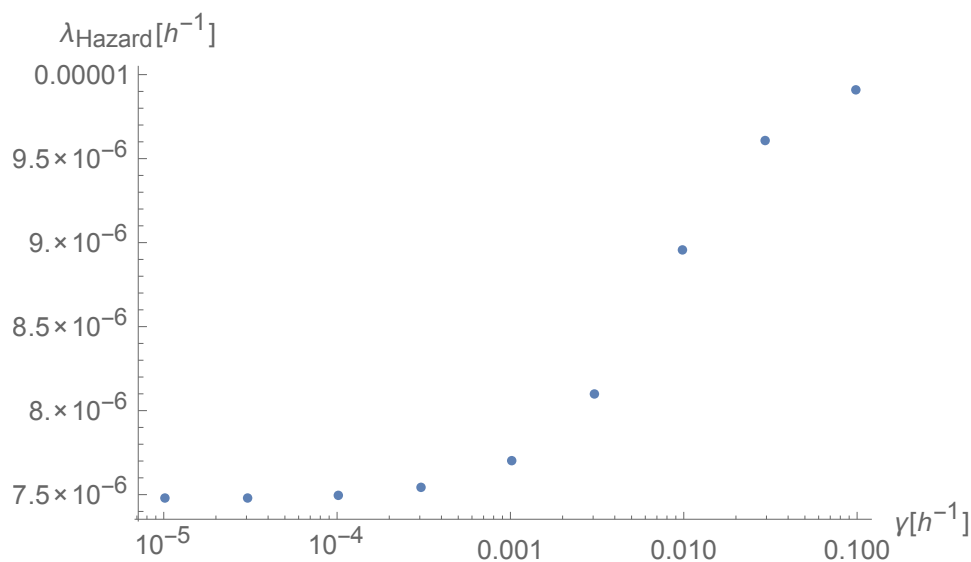


Obrázek 6.2: Graf závislosti pro změnu parametru  $\lambda$ .

## 6.2 2. část výsledků – Závislost na změně intenzity poruchy nouzové metody

V této části byly použity všechny výchozí hodnoty kromě  $\gamma$ , která se měnila v rozpětí od  $10^{-1}$  do  $10^{-5}$  (Tabulka 6.2 a Obrázek 6.3).

$\gamma [h^{-1}]$	$\lambda_{Hazard} [h^{-1}]$
$10^{-1}$	$9.91672 * 10^{-6}$
$3 * 10^{-2}$	$9.60333 * 10^{-6}$
$10^{-2}$	$8.95111 * 10^{-6}$
$3 * 10^{-3}$	$8.0986 * 10^{-6}$
$10^{-3}$	$7.70162 * 10^{-6}$
$3 * 10^{-4}$	$7.5473 * 10^{-6}$
$10^{-4}$	$7.49805 * 10^{-6}$
$3 * 10^{-5}$	$7.48056 * 10^{-6}$
$10^{-5}$	$7.47556 * 10^{-6}$

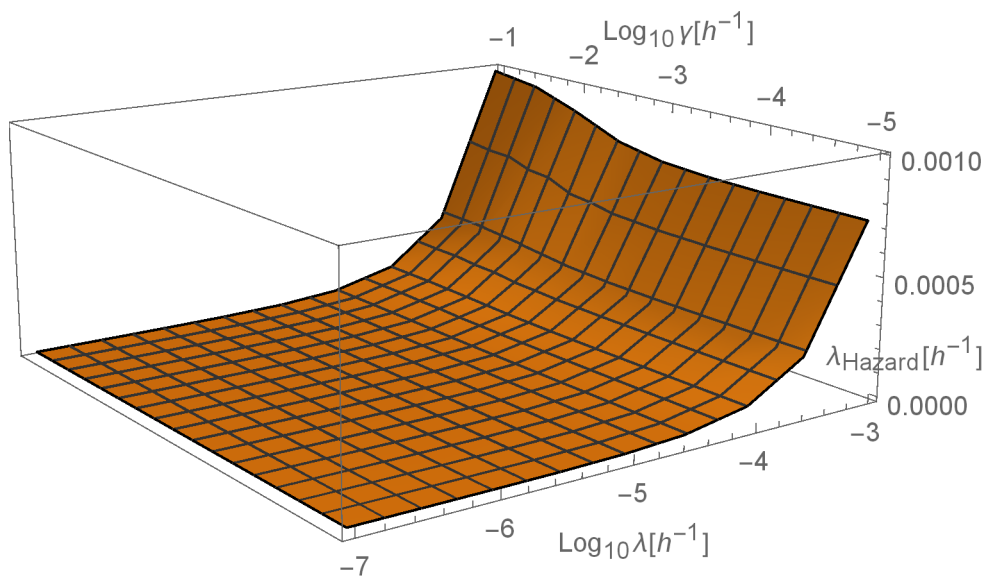
Tabulka 6.2: Tabulka hodnot závislosti při změně parametru  $\gamma$ .Obrázek 6.3: Graf závislosti pro změnu parametru  $\gamma$ .

### 6.3 3. část výsledků – Závislost na změnách obou parametrů

V této části byly použity všechny výchozí hodnoty kromě  $\lambda$  a  $\gamma$ , které se měnily v rozpětích z předchozích bodů (Obrázek 6.4).

Vzhledem k tomu, že program Mathematica zatím neumí vykreslování 3D grafů s logaritmičsky škálovanými stupnicemi, jsou hodnoty na ose  $\lambda$  a  $\gamma$  zlogaritmovány předem (např. hodnota  $10^{-2}$  je v grafu jako hodnota  $-2$ ).





Obrázek 6.4: 3D graf výsledných závislostí.



---

## Závěr

Cílem mé bakalářské práce bylo analyzovat možnosti pro tvorbu jednoduchých knihoven do běžně užívaného programu Wolfram Mathematica. Vytvořená knihovna měla být aplikována na Markovské modely spolehlivosti dodané vedoucím bakalářské práce.

Výsledkem mé bakalářské práce je přehledný manuál pro tvorbu knihovny v pluginu Wolfram Workbench, který běží pod programem Eclipse. Práce obsahuje popis návodu na tvorbu jednoduchého projektu, import více knihovních souborů pomocí jedné velké knihovny a závěrečný export do programu Wolfram Mathematica. Podle tohoto manuálu je v práci zpracována knihovna, kterou jsem následně použil pro výpočty závislostí intenzity poruch pro pravděpodobnosti chyby u Markovského modelu, který mi dodal vedoucí bakalářské práce.

Moje bakalářská práce může mít do budoucna širší využití, a to jako jednoduchý návod na tvorbu jakýchkoliv knihoven pro program Wolfram Mathematica. Další využití práce může být při úpravách výpočtů modelů vedoucího bakalářské práce, pana doktora Kohlíka, a také v uvedených předmětech vyučovaných na FIT ČVUT v Praze. Vzniklý návod umožní tvorbu knihovny pro program Wolfram Mathematica, která by například mohla generovat sadu testů pro předmět Číslicové a analogové obvody včetně výsledků. Vlastní realizace takovéto knihovny může přinést nové otázky, které mohou vyžadovat tvorbu dalších knihoven. Tyto otázky se však objeví až při konkrétní realizaci takového zadání. Manuál obsažený v mé bakalářské práci může v budoucnu sloužit ke vzniku dalších knihoven, které pomohou konkrétní zadání vyřešit.



---

## Literatura

- [1] Wolfram: What Is Mathematica? [online]. Srpen 2016, [Citováno 2017-05-10]. Dostupné z: <https://reference.wolfram.com/legacy/v1/contents/whatis.pdf>
- [2] Wolfram: Wolfram Workbench [online]. Květen 2017, [Citováno 2017-05-10]. Dostupné z: <https://www.wolfram.com/workbench/>
- [3] Wolfram: Wolfram Workbench References [online]. Květen 2017, [Citováno 2017-05-10]. Dostupné z: <http://reference.wolfram.com/workbench/index.jsp>
- [4] Lipka, R.: Analytické pravděpodobnostní modely, Markovské procesy, podklady pro výuku na FEL ČVUT v Praze.
- [5] Hlavička, J. a. k.: Číslíkové systémy odolné proti poruchám. Vydavatelství ČVUT, 1992, s. 47–54.
- [6] Statistika a spolehlivost v lékařství – Markovovy modely, podklady pro výuku na FEL ČVUT v Praze.
- [7] Kohlík, M.: Hierarchical Dependability Models Based on Markov Chains, a dissertation thesis. 2015.
- [8] Wolfram: Wolfram Mathematica [online]. Květen 2017, [Citováno 2017-05-10]. Dostupné z: <http://www.wolfram.com/mathematica/>
- [9] Eclipse: Eclipse IDE [online]. Květen 2017, [Citováno 2017-05-10]. Dostupné z: <http://www.eclipse.org/>



---

## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	manual.txt.....	manuál pro práci se soubory na CD
	src.....	adresář se zdrojovými soubory
	mathematica_library.....	zdrojové soubory knihovny
	mathematica_source.....	notebook a data pro práci s knihovnou
	thesis.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF