Technická univerzita v Košiciach Fakulta elektrotechniky a informatiky

Reengineering a ďalší rozvoj výpočtových modelov propagácie častíc kozmického žiarenia v heliosfére na platforme CUDA

Diplomová práca

Bc. Martin Nguyen

2021

Technická univerzita v Košiciach Fakulta elektrotechniky a informatiky

Reengineering a ďalší rozvoj výpočtových modelov propagácie častíc kozmického žiarenia v heliosfére na platforme CUDA

Diplomová práca

Študijný program:InformatikaŠtudijný odbor:9.2.1 InformatikaŠkoliace pracovisko:Katedra počítačov a informatiky (KPI)Školiteľ:RNDr. Pavol Bobík, PhD.Konzultant:doc. Ing. Ján Genči PhD.

Košice 2021

Bc. Martin Nguyen

- Názov práce: Reengineering a ďalší rozvoj výpočtových modelov propagácie častíc kozmického žiarenia v heliosfére na platforme CUDA
- Pracovisko:Katedra počítačov a informatiky, Technická univerzita v Ko-
šiciachAutor:Bc. Martin NguyenŠkoliteľ:RNDr. Pavol Bobík, PhD.Konzultant:doc. Ing. Ján Genči PhD.Dátum:23. 4. 2021Kľúčové slová:heliosféra, kozmické žiarenie, CUDA

Abstrakt: Diplomová práca sa zaoberá reinžinieringom existujúcich modelov a implementáciou nových výpočtových modelov zameraných na určenie modulácie kozmického žiarenia v heliosfére. Vývoj modelov bol cielený na platformu grafických kariet Nvidia s podporou technológie CUDA. Práca sa tiež venuje vývoju podporných nástrojov, ktoré svojimi funkcionalitami umožňujú používateľovi automatizovať beh výpočtových modelov, analyzovať a vizualizovať výsledky, ktoré modely produkujú. V práci sú analyzované známe chyby pri 1D F-p výpočtovom modeli, zadefinované hypotézy a overené ich pravdivosti. Práca obsahuje opis rozšírenia výpočtových modelov o nový 2D F-p model na základe nezávislej publikácie.

- Thesis title: Reengineering and further development of computational models of cosmic ray particle propagation in the heliosphere on the CUDA platform
- Department:Department of Computers and Informatics, Technical University of KošiceAuthor:Bc. Martin NguyenSupervisor:RNDr. Pavol Bobík, PhD.Tutor:doc. Ing. Ján Genči PhD.Date:23. 4. 2021Keywords:heliosphere, cosmic rays, CUDA

Abstract: The diploma thesis deals with the reengineering of existing models and the implementation of new computational models focused on simulations of the cosmic ray modulation in the heliosphere. The development of the models was targeted at the Nvidia graphics card platform with the support of CUDA technology. The content of the diploma thesis includes the development of support tools that, with their functionalities, allow the user to automate the running of computational models, analyze and visualize the results that the models produce. In the diploma thesis, known errors in the 1D F-p computational model are analyzed, hypotheses are defined and their validity is verified. The work contains a description of the extension of computational models with a new 2D F-p model based on the independent publication.

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: Informatika Študijný program: Informatika

Názov práce:

Reengineering a ďalší rozvoj výpočtových modelov propagácie častíc kozmického žiarenia v heliosfére na platforme CUDA Reengineering and further development of computational models of

cosmic ray particle propagation in the heliosphere on the CUDA platform

Študent:

Bc. Martin Nguyen

Školiteľ:

RNDr. Pavol Bobík, PhD.

Školiace pracovisko:

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Študent sa oboznámi so základmi fyziky modulácie častíc kozmického žiarenia v heliosfére.

2. Študent sa oboznámi s CUDA implementáciou 1D modelu modulácie častíc kozmického žiarenia v heliosfére a vykoná analýzu chýb kódu pri malých časových krokoch simulácií.

3. Vytvorí systém automatizácie výpočtov v CUDA 1D modeli pre maticu vstupných parametrov. Systém vykoná zadané simulácie a automatické spracovanie ich výsledkov a ich vizualizáciu. Pripraví prehliadač získaných výsledkov umožňujúci ľahšiu orientáciu vo väčšom množstve simulácií.

4. Implementuje 2D F-p verziu modelu a zahrnie ho do základnej verzie systému automatizácie.

5. Podľa pokynov vedúceho práce vytvorí dokumentáciu.

Jazyk, v ktorom sa práca vypracuje: slovenský Termín pre odovzdanie práce: 23.04.2021 Dátum zadania diplomovej práce: 30.10.2020

prof. Ing. Liberios Vokorokos, PhD. dekan fakulty

Čestné vyhlásenie

Čestne vyhlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 23.4.2021

Vlastnoručný podpis

Poďakovanie

Chcel by som poďakovať každému, kto mi bol, priamo alebo nepriamo, nápomocný pri tvorbe tejto diplomovej práce a mojim blízkym známym, ktorí ma pri písaní diplomovej práce podporovali.

Špeciálne ďakujem patrí vedúcemu práce RNDr. Pavlovi Bobíkovi, PhD. za objasnenie tejto fascinujúcej témy, pravidelné konzultácie, cenné rady, profesionálny prístup a odbornú pomoc. Zároveň sa chcem poďakovať konzultantovi diplomovej práce doc. Ing. Jánovi Genčimu, PhD. za korekciu chýb a cenné pripomienky k formálnej stránke textu.

Obsah

Μ	otivá	cia		1
1	Formulácia úlohy			3
2	Analýza paralelizovaných problémov na GPU			5
	2.1	Parale	lizovateľnosť obrazového spracovania	5
	2.2	Parale	lizovateľnosť simulácii	7
	2.3	Parale	lizované problémy v astrofyzike	8
3	Teó	ria náh	odnosti čísel	10
	3.1	Vyhoo	Inotenie náhodnosti hodov kocky	10
	3.2	Vyhoo	Inotenie stratégii v hre ruleta	13
		3.2.1	Zdvojnásobovanie prehry	15
		3.2.2	Zdvojnásobovanie výhry	16
	3.2.3 Konštantné vkladanie sumy			
		3.2.4	Celkové príjmy kasína pri jednotlivých stratégiách	19
4	Ana	lýza ch	ýb jednorozmerných modelov	21
	4.1	Pulzá	cia	21
		4.1.1	Beh programu bezo zmien s vymedzenou injekciou kinetic-	
			kej energie	24
		4.1.2	Odstránenie optimalizačného prepínača	25

		4.1.3	Zmena jednoduchej presnosti na dvojitú	25		
	4.2	Štatist	ická chyba F-p metódy	28		
		4.2.1	Prvá verzia injekcie častíc	30		
		4.2.2	Druhá verzia injekcie častíc	32		
		4.2.3	Tretia verzia injekcie častíc	33		
		4.2.4	Zhrnutie	33		
5	Rep	rodukc	ia Pei-ovho modelu	35		
	5.1	Prepis	Pei-ovho modelu	35		
	5.2	Vyhoc	Inotenie 2D modelu	40		
6	Ďal	šie rozš	írenie výpočtového systému heliosferických výpočtov	42		
	6.1	Požiac	lavky systému	42		
	6.2	Navrh	ovaný prístup ku spracovaniu parametrov	44		
	6.3	3 Navrhovaný prístup ku spúšťaniu podvýpočtov 4				
	6.4	Navrh	ovaný prístup k optimalizačnému vyhodnoteniu	45		
	6.5	Dopln	enie vyhodnocovacích algoritmov	47		
	6.6	6.6 Export výstupných dát v podobe grafov				
7	Náv	rh soft	vérového riešenia pre systém kontroly výsledkov	49		
	7.1	Analý	za cieľových používateľov	49		
	7.2	Analý	za súčasného pracovného toku	50		
	7.3	Návrh	cieľovej platformy	51		
	7.4	Návrh	potrebných sieťových funkcionalít	53		
	7.5	Návrh	exportovania dát do štandardizovaného formátu	54		
	7.6	Návrh	dodatočných funkcionalít	55		
8	Imp	lement	ácia softvérového riešenia pre systém kontroly výsledkov	56		
	8.1	Imple	mentácia pripojenia na server	57		

	8.2	Implementácia prehľadu výpočtov	58		
	8.3	Implementácia vizualizácie dát	59		
9	Vyh	odnotenie nástroja	60		
	9.1	Spustenie výpočtu	60		
	9.2	Zobrazenie výsledkov a vyhodnotenie chyby	61		
10	Závo	er	66		
Lit	eratú	ira	69		
Zo	znan	nskratiek	72		
Zo	oznam príloh 74				

Zoznam obrázkov

3.1	Tabuľka konvergujúcich pomerov vygenerovaných čísel	12
3.2	Graf strednej kvadratickej odchýlky (RMSE)	14
3.3	Histogramy výsledných súm pri zdvojnásobovaní prehry	16
3.4	Percentuálny podiel profitujúcich hráčov pri zdvojnásobovaní pre-	
	hry	16
3.5	Histogramy výsledných súm pri zdvojnásobovaní výhry	17
3.6	Percentuálny podiel profitujúcich hráčov pri zdvojnásobovaní vý-	
	hry	18
3.7	Histogram početnosti konečných súm pri zdvojnásobovaní výhry .	18
3.8	Percentuálny podiel profitujúcich hráčov pri konštantnom vkladaní	
	1\$	19
3.9	Príjmy kasína v dolároch pri jednotlivých stratégiách	20
4.1	Graf pomeru F-p a B-p pri $dt = 0.5s$ a $K_0 = 5 * 10^{22} cm^2/s$	22
4.2	Spektrá na 1AU vyhodnotené Monte Carlo metódou (červená) a	
	Crank-Nicolson metódou (čierna) pomocou F-p a F-T metódy	23
4.3	Graf pomeru F-p a B-p pri $dt=0.5s,k0=5*10^{22}cm^2/s$ v experi-	
	mente pre TKin <45GeV,57GeV>	24
4.4	Graf pomeru F-p a B-p pri $dt=0.5s,k0=5*10^{22}cm^2/s$ v experi-	
	mente pri odstránení prepínača pre TKin <45GeV,57GeV>	26

4.5	Graf pomeru F-p a B-p pri $dt = 0.5s$, $k0 = 5 * 10^{22} cm^2/s$ v experimente pri použití dvojitej prenosti pre TKin <45GeV,57GeV> na vzorke 30mld	27
4.6	Graf pomeru F-p a B-p pri $dt = 0.5s$, $k0 = 5 * 10^{22} cm^2/s$ v experimente pri použití dvojitej prenosti pre TKin <45GeV,57GeV> na vzorke 300mld	27
4.7	Pomer medzi Monte-Carlo a Crank-Nicholson metódami. Vľavo pre B-p a F-p, vpravo pre B-T a F-T	28
4.8	Graf pomeru F-p a B-p pri dt=50s a $k = 5 * 10^{22} cm^2/s$	29
4.9	Histogram B-p simulácie pre N=10, $dt = 50s$, $k0 = 5 * 10^{22} cm^2/s$ a $V = 400 km/s$ pre TKin <0.9GeV,1.1GeV>	30
4.10	Histogram B-p simulácie pre N=10, $dt = 50s$, $k0 = 5 * 10^{22} cm^2/s$ a $V = 400 km/s$ pre TKin <9GeV,11GeV>	30
4.11	Graf pomeru F-p/B-p pre modifikovanú verziu 1	31
4.12	Graf pomeru F-p/B-p pre modifikovanú verziu 2	32
4.13	Graf pomeru F-p/B-p pre modifikovanú verziu 3	33
4.14	Graf pomerov F-p/B-p v modifikovanej verzii 1, 2 a 3 pre TKin<0.9GeV	,
	1.1GeV>	34
4.15	Graf pomerov F-p/B-p v modifikovanej verzii 1, 2 a 3 pre TKin<9GeV,	
	11GeV>	34
5.1	Graf pomerov Pei CPU/GPU pre dt=0.5s	37
5.2	Graf pomerov Pei CPU/GPU pre dt=2s	37
5.3	Heatmapa pre 50GeV	38
5.4	Heatmapa pre 1GeV	38
5.5	Vizualizované dáta z výpočtového modelu	41
6.1	Princíp fungovania samoadaptačného a samooptimalizačného sys-	
	tému	43
6.2	Automaticky vygenerované grafy	48

7.1	Proces behu programu	50
7.2	Návrh grafického rozhrania	52
8.1	Schéma procesu pripojenia na server	57
8.2	Schéma procesu zobrazenia prehliadača výpočtov	58
9.1	Okno nástroja pre spustenie výpočtu	61
9.2	Prehľad aktuálne vykonávaného výpočtu	61
9.3	Hotový výpočet s určením odchýlky	62
9.4	Tepelná mapa pre odchýlku od databázy 559	63
9.5	Tepelná mapa pre odchýlku od databázy 372	64
10.1	Potvrdenie správnosti modelov pri určení chyby na tepelnej mape z nástroja	68
	,	

Zoznam tabuliek

3.1	Tabuľka počtu vygenerovaných čísel	11
3.2	Tabuľka konvergujúcich pomerov vygenerovaných čísel	12
3.3	Tabuľka strednej kvadratickej odchýlky čísel	13
9.1	Tabuľka parametrov s najmenšou odchýlkou od Force Field spektra s potenciálom 559MV	63
9.2	Tabuľka parametrov s najmenšou odchýlkou od Force Field spektra	
	s potenciálom 372MV	64

Motivácia

Otázka paralelného spracovania úloh sa rieši už niekoľko desaťročí. Klasická paralelizácia bola cielená na delenie úloh na menšie podúlohy, ktoré boli vykonávané v rámci procesov na viacprocesorových systémoch. Až v 21. storočí sme si uvedomili potenciál výpočtovej sily GPU na riešenie niektorých špecifických úloh, ktoré je možné paralelizovať. Spoločnosť Nvidia prišla na trh s technológiou CUDA, ktorá umožňuje ovládať GPU prostredníctvom programového kódu v jazykoch C++ alebo Fortran. Zároveň sa do architektúry GPU zavádza stále väčšie množstvo jadier, čím sa výpočtová kapacita GPU neustále zvyšuje. Na dnešných grafických procesoroch je pomocou technológie CUDA možné riešiť celý rad komplexných úloh. Keďže už je možné na týchto zariadeniach vykonávať aj úlohy, ktoré sa už nevyhnutne nezaoberajú len obrazovým spracovaním, označujeme dnešné grafické procesory ako GPGPU. Predkladaná diplomová práca sa zameriava na paralelné riešenie masívnych výpočtových modelov modulácie kozmického žiarenia v heliosfére.

Spomínané výpočtové modely sú vhodné na paralelizáciu, keďže pri nich nezáleží na ich poradí. Modulácia kozmického žiarenia je proces, pri ktorom častice vchádzajúce do heliosféry strácajú energiu a ich pohyb je ovplyvnený jej magnetickým poľom. Výsledkom je spektrum s nižšími intenzitami častíc vnútri heliosféry v porovnaní s intenzitami mimo heliosféry. Algoritmus výpočtov modulácie kozmického žiarenia v heliosfére tvorí mnohokrát opakovaná úloha trasovania častice s rôznou energiou v heliosfére. Tieto výpočty sú od seba nezávislé a spolu tvoria hľadanú fyzikálnu distribúciu celého modulačného procesu.

Cieľom tejto diplomovej práce je paralelizovanie výpočtových modelov na určenie modulácie kozmického žiarenia v heliosfére. Pomocou technológie CUDA sa snažíme o zrýchlenie simulácie na GPU a dosiahnutie niekoľkonásobne menšieho výpočtového času oproti vykonávaniu na CPU. Súčasná práca nadväzuje na diplomovú prácu, v ktorej sa paralelizovateľné časti programového kódu jednorozmerných a dvojrozmerných výpočtových modelov modulácie častíc kozmického žiarenia prepisovali pomocou CUDA technológie na kódy, ktoré sú vykonávateľné na grafických kartách Nvidia CUDA.

V súčasnosti neexistujú verejne dostupné modely heliosferických výpočtov, ktoré by boli vykonávateľné na GPU. Existujú publikácie, v ktorých sa autori prepisu venovali, avšak ich zdrojové kódy nie sú sprístupnené verejnosti. Motiváciou pre vznik tejto práce bolo vytvorenie a sprístupnenie numerického riešenia modelov heliosferickej modulácie kozmického žiarenia na platforme GPGPU širšej vedeckej komunite.

1 Formulácia úlohy

Náplňou tejto diplomovej práce je analyzovať existujúci systém, ktorý je určeny na paralelné numerické riešenie Parkerových rovníc pre riešenie úlohy modulácie kozmického žiarenia v heliosfére na platforme CUDA. Časťou práce je zadefinovať hypotézy možných riešení problémov v existujúcich modeloch a experimentálne overiť ich pravdivosť.

V predkladanej diplomovej práci navrhujeme doplníť systém o nový dvojdimenzionálny model a vyhodnotiť správnosť jeho implementácie. Zároveň v ňom navrhujeme implementovať podporu spúšťania multivýpočtov na základe zadania viacerých hodnôt parametrov. Implementovanými zmenami pripravíme systém na samoadaptáciu a samooptimalizáciu, pod ktorými rozumieme schopnosť algoritmu určiť pre každý výpočet odchýlku od referenčných výsledkov a na základe získaných dát jednoznačne zvoliť optimálnu kombináciu prepínačov. Pre efektívne spracovanie výsledných dát z modelov bolo naplánované vytvoriť externý overovací nástroj.

Pre úspešné zvládnutie hlavných bodov diplomovej práce analyzujeme existujúce riešenia paralelizovateľných problémov v rôznych technických, medicínskych a fyzikálnych odvetviach (kap.2). Vyvíjané modely používajú vo svojej implementácii Monte-Carlo prístup, ktorý je vo veľkej miere založený na náhodných číslach. Pre zabezpečenie správnej implementácie modelov navrhujeme v práci overiť správnosť generátorov náhodných čísel na grafickej karte Nvidia pozorovaním uniformnosti v procese generovania (kap.3). Pri overovaní je vhodné použiť štatistický problém hracích stratégií v hazardných hrách z oblasti teórie náhodnosti čísel.

Existujú dva základné problémy súvisiace s hardvérovými limitmi grafických kariet v systéme výpočtových modelov propagácie častíc kozmického žiarenia

v heliosfére. Pre riešenie pulzácie a štatistickej chyby v doprednom jednorozmernom F-p modeli navrhujeme zadefinovať hypotézy o ich pôvode a analyzovať ich možné riešenia (kap.4).

Náplňou diplomovej práce je doplniť systém o nový dvojdimenzionálny model, ktorého matematický opis bol predstavený nezávislými doménovými expertmi. Zdrojový kód modelu ani súbory s výstupnými dátami nie sú dostupné. Pri implementácii je nutné vychádzať z kriviek grafu spektier a matematických vzorcov uvedených v publikácií, na základe ktorých je možné model vytvoriť. Správnosť implementovaného modelu je potrebné overiť a porovnať s našimi existujúcimi paralelizovanými modelmi (kap.5).

Na základe požiadaviek, ktoré vznikli počas používania systému, je potrebné rozšíriť systém o ďalšie funkcionality. Systém je nutné doplniť o spracovanie používateľom definovanej matice parametrov, implementovať viacnásobné spúšťanie výpočtov, pridať generovanie výstupných grafov a aktualizovať dodatočné prepínače v CLI verzii systému (kap.6).

V rámci práce bude navrhnutý podporný externý nástroj, ktorý zefektívni prácu so spracovaním dát výskumníkov SAV (kap.7). Navrhujeme, aby implementovaný nástroj umožnil používateľovi bezpečne sa pripojiť na server, stiahnuť dáta zo serveru, umožniť prezeranie detailov hotových výpočtov a vizualizovať dáta v podobe grafov a tepelných máp (kap.8). Funkčnosť nástroja je potrebné otestovať na konkrétnom príklade použitia (kap. 9).

2 Analýza paralelizovaných problémov na GPU

S príchodom GPU sa z hľadiska architektúry zariadení v minulosti uvažovalo aj nad masívne paralelným riešením bežných výpočtových úloh a simulácii, ktoré boli implementované v klasickom programovacom jazyku. Bežne zaužívané paralelizácie boli zamerané na delenie úloh pomocou technológií ako OpenMP v rámci procesov na CPU alebo na paralelne bežiacich CPU klastroch. Príchod paralelnej výpočtovej platformy CUDA na trh zaujal ako odbornú, tak aj amatérsku programátorskú komunitu. Paralelelizácia výpočtových úloh na GPU sa tak stala prístupnejšou. Výhodou niektorých typov výpočtu na GPU oproti CPU je, samozrejme, možnosť niekoľkonásobného zrýchlenia trvania výpočtov a simulácii kvôli ich veľkému počtu jednoduchých jadier, ktoré v danom čase vedia masívne a veľmi rýchlo vykonávať aritmetické operácie.

2.1 Paralelizovateľnosť obrazového spracovania

Bežným výpočtovým problémom, s ktorým GPU pracuje, je spracovanie obrazu. Ako už z názvu GPU vyplýva, ide o grafický procesor, ktorého špecializácia je mierená hlavne na obrazové spracovanie. Príkladom obrazového spracovania sú rôzne oblasti počítačového videnia a analýzy obrazu. Matematické operácie pri obrazovom spracovaní predstavujú vo väčšine prípadov iba jednoduché operácie s celočíselnými dátami. Pri spracovaní obrazu nepotrebujeme získať veľkú presnosť a teda kvôli rýchlosti pracuje mnoho grafických procesorov výhradne s číslami s jednoduchou presnosťou.

Príkladom paralelizovaných problémov z oblasti obrazového spracovania je

využívanie konvolučných neurónových sietí určených predovšetkým pre detekciu a rozpoznanie objektov a osôb v obraze. Jednými z prvých autorov, ktorí experimentálne skúmali zlepšovanie výkonnosti ako aj možnosť škálovania konvolučných sietí pri prepise modelov do CUDA prostredia boli autori v [22]. V článku dobre opisujú proces konvolúcie, ktorý spojením s podvzorkovaním vytvára v jednotlivých krokoch zmenšujúce sa mapy príznakov. Konvolučné vzorce sa dajú na grafickej karte paralelizovať, avšak kvôli ich nepravidelnému pamäťovému prístupu sa ťažko optimalizujú. V článku porovnávajú výkonnosť prepísaných sietí medzi procesorom Intel Core i7 860 a grafickou kartou GTX 275. Autori uvádzajú 2 až 16-násobné zrýchlenie procesu vykonávania na grafickej karte. V dobe písania ich článku mala cena ich počítačovej zostavy hodnotu približne 900 dolárov. Porovnanie ceny a výkonnosti je dobrým ukazovateľom toho, či je investícia do prechodu z prostredia CPU klastrov na grafické karty pre projekt výhodná. V mnohých prípadoch sa ukazuje, že to je výhodnejšie.

Rozšíreným príkladom použitia konvolučných neurónových sietí z reálneho života, kedy je potrebné detegovať nejaký text alebo číslice, je detekcia a rozpoznanie evidenčného čísla vozidla. Prvé algoritmy, ktoré využívali neurónové siete pre rozpoznanie znakov a číslic z obrazu siahajú už do 80-tych rokov 20. storočia. Napríklad autori v [26] využili rozpoznávanie rukou písaných čísel pomocou neurónovej siete. Problematika detekcie objektov ako aj samotný výskum bol v tejto dobe ešte v skorých začiatkoch, pričom neexistovalo veľa metód, ktoré by boli v danej dobe použiteľné. Dnes je však táto oblasť hlboko preskúmaná a existuje hneď niekoľko algoritmov a spôsobov ako túto problematiku riešiť.

Výskumu využitia GPU výpočtov pre zrýchlenie procesu rozpoznania čísel z obrazu sa venovali aj autori v [20], kde implementovanú konvolučnú neurónovú sieť v jazyku C mierne upravili tak, aby bola vykonávateľná v prostredí CUDA. V experimente porovnali rýchlosť vykonávania na procesore Intel Pentium D 925 a na grafickej karte Nvidia 9600GT. Presnosť algoritmu bola zachovaná a výkonnostným meraním dosiahli vo výsledku približne 8-násobne rýchlejší proces rozpoznávania čísel z obrazu. Rovnako sa GPU použilo aj pre trénovanie neurónovej siete, kde sa autorom podarilo zrýchliť tréningový proces 6-násobne.

Ďalším príkladom je publikácia [23], kde sa autori zaoberali implementáciou rýchleho detektora Harris-Hessian bodov na obrázku s nízkym rozlišením. Al-

goritmus je zameraný na detekciu príznakov (angl. features) v obraze, ktorý vo výstupe extrahuje charakteristické body v obraze slúžiace pre ďalšie spracovanie, napríklad kategorizáciu objektov alebo detekciu objektov. Výsledok experimentu s nasadením GPU predstavoval 10 až 20-násobné zrýchlenie oproti výpočtu na CPU, pričom vysoká presnosť algoritmu zostala nezmenená.

2.2 Paralelizovateľnosť simulácii

Príkladom na akcelerované výpočty možno nájsť aj vo viacerých medicínskych výskumoch, napríklad výskumníci v článku [9] využili paralelizovateľnosť problému z medicínskeho prostredia a popísali ich riešenie akcelerácie výpočtovej simulácie rádiofrekvenčnej ablácie pomocou dvoch Nvidia GPU kariet obsahujúcich CUDA jadrá. Vo výsledku dosiahli až 17-násobné zrýchlenie simulácie oproti sekvenčnému riešeniu problému na CPU. Autori prácu rozšírili a v článku [4] využili paralelizáciu pri procese simulácie rádiofrekvenčnej ablácie pre liečenie rakoviny pečene. Reálny zákrok trvá v priemere 5 minút, pričom dĺžka simulácie tohto procesu bez použitia GPU dosahuje až 20 hodín. Autori priebeh simulácie transformovali do CUDA kódu a po testovacom meraní získali 15 až 18-násobné zrýchlenie, pričom presnosť numerických výsledkov ostala nezmenená.

Niekedy sa však GPU nepoužíva ako hlavné výpočtové zariadenie na výpočet celého priebehu simulácie, ale funguje v kolaborácii s CPU. Článok [18] sa zaoberal paralelizáciou simulácie umelej spoločnosti so zameraním na štatistické pozorovanie špecifického správania jednotlivcov v spoločnosti. Pri implementácii problému pracovali v dvoch verziách, pričom v oboch prípadoch GPU neslúžilo ako hlavné výpočtové zariadenie, ale malo funkciu zariadenia, ktoré poskytuje podporný výpočtový výkon k CPU. V prvej verzii použili GPU ako hostiteľský procesor, pričom simulačné výpočty prebiehali v spolupráci so simulačnými jadrami CPU. V druhej verzii sa využívanie GPU optimalizovalo a na funkciu koprocesora sa použilo GPU, pričom dodatočný simulačný rámec fungujúci na báze SPMT (Single process Multi threads) sa využil na integráciu GPU a simulačných jadier CPU. Vyhodnotenie výsledkov preukázalo, že využitie GPU na riešenie konkrétnej simulácie je rýchle, avšak prenosová rýchlosť komunikácie a synchronizácie medzi GPU a CPU vytvárajú tzv. úzke hrdlo (angl. bottleneck) systému.

2.3 Paralelizované problémy v astrofyzike

Z pohľadu paralelných výpočtov a simulácii z prostredia vesmírneho výskumu sa častokrát spomína paralelizácia Monte-Carlo metód, ktoré boli po dlhý čas paralelizované len v prostredí CPU klastrov. S príchodom CUDA vývojárskych nástrojov a rozvoju paralelizácie na grafických kartách sa sparalelizovali aj prístupy pre simuláciu častíc kozmického žiarenia v práci Tautza [16]. Jeho simulácie sa častokrát používajú na pozorovanie problémov, ktoré súvisia so šírením a zrýchľovaním nabitých častíc kozmického žiarenia alebo solárnych častíc pohybujúcich sa v elektromagnetickom poli heliosféry a v medzihviezdnom priestore. V pôvodnej verzii simulácie sa používal Intel Xeon CPU klaster pozostávajúci zo 63 procesorových jadier. Simulačný experiment autor prepísal do prostredia CUDA a otestoval jeho rýchlosť na grafickej karte Nvidia Tesla, ktorá obsahovala 448 CUDA jadier. Výpočet na procesore trval približne 8 hodín a 30 minút, pričom pri použití rovnakých parametrov a simulačného modelu sa dosiahlo na grafickej karte 3,5-násobné zrýchlenie.

Paralelnému riešeniu Fokker-Planckovej rovnice sa venovali autori v [17]. Táto téma je hlavnou témou aj našej diplomovej práce ako aj predchádzajúcich prác Solanika v [11] a [5]. Dopredná simulácia častíc kozmického žiarenia oproti spätnej simulácii je omnoho výpočtovo náročnejšia, keďže pri F-p verziách modelu dostaneme len veľmi málo častíc, ktoré vyhovujú našim požiadavkám. Napríklad, pri injekcii častíc z hranice heliopauzy pri 100AU sa snažíme o zachytávanie častíc kozmického žiarenia pri 1AU, ktorá zodpovedá vzdialenosti Zeme od Slnka. Castica prechádza heliosférou a mnohokrát mení svoj smer na iregularitách magnetického poľa heliosféry. Simulácia pohybu častice sa odvíja od Monte-Carlo prístupu. Keďže sa častica pohybuje difúzne v heliosfére a pri ceste do vnútornej heliosféry sa pohybuje proti smeru slnečného vetra, tak je veľmi málo simulácii, keď sa častica injektovaná na 100AU dostane až na 1AU. Naopak, pri spätnej B-p simulácii injektujeme časticu na 1AU a simulujeme pohyb spätne v čase tak, aby častica skončila mimo heliosféry na úrovni heliopauzy. Pri B-p metóde získame skoro 100% úspešnosť, pričom pri F-p metóde to závisí aj od ďalších parametrov. Je preto výhodnejšie používať B-p metódu aj keď pre výskum pohybu častíc je vhodné simulovať pohyb aj prostredníctvom F-p modelu.

Stochastické diferenciálne rovnice paralelizované v práci [17] boli prepísané v rámci práce aj do CUDA prostredia a podobne ako pri predchádzajúcich prácach bolo porovnané zrýchlenie simulácie oproti CPU simuláciám bežiacim na CPU klastroch. Autori v CPU verzii používali OpenMP technológiu pre paralelné spracovanie výpočtov na 8 jadrách v dvoch testovacích výpočtoch, najprv bez použitia optimalizačných prepínačov prekladača GCC na zrýchlenie matematických operácii a v druhom teste aj s ich použitím. V paralelnej GPU verzii autori využili grafickú kartu Nvidia GeForce GTX 660 s počtom 192 CUDA jadier a podobne ako pri CPU verzii v prvom teste nepoužívali optimalizačný prepínač, ktorý ponúka prekladač NVCC a v druhom teste bol výpočet spustený aj s optimalizáciou fast-math. Pri výpočtoch sa vo veľkej miere používajú náhodne vygenerované čísla, ktoré bolo nutné v CUDA verzii kódu správne implementovať. Pri prepise modelov využívali generátor pseudonáhodných čísel XORWOW. Správnosť prepísaných modelov bola skontrolovaná na základe porovnania výstupných dát z [19]. Z hľadiska rýchlosti bolo vykonávanie F-p metódy na GPU značne rýchlejšie. Autori uvádzajú 10 až 60-násobné zrýchlenie oproti výpočtom na CPU.

3 Teória náhodnosti čísel

Generovanie náhodných čísel pri fyzikálnych simuláciách prechodu častíc kozmického žiarenia heliosférou, od jej hraníc po orbitu Zeme, tvorí jeden zo základných stavebných prvkov tejto diplomovej práce.

Aby bolo generovanie náhodných čísel skutočne uniformné, teda aby bola pravdepodobnosť vygenerovania každého čísla rovnaká, bola pre tento účel vytvorená nasledujúca kapitola, ktorá sa zaoberá úlohami s náhodnými číslami. Na týchto úlohách sme preverili vhodnosť implementácie generovania náhodných čísel v CUDA prostredí pre fyzikálne simulácie. Naprogramované úlohy úzko súvisia s teóriou náhodných čísel. Pre efektívne spracovanie boli úlohy implementované paralelne a experimentálne otestované na počítači s grafickou kartou Nvidia GTX 1070Ti, ktorá obsahuje 2432 CUDA jadier.

Prvá podkapitola sa zaoberá simuláciou náhodných hodov kocky. Úlohou tejto podkapitoly je zistiť, či generovanie čísel na grafickej karte je uniformné. Distribúcia číslic možných hodnôt vygenerovanej číslice musí byť približne rovná matematickej pravdepodobnosti hodu konkrétneho čísla pri hode kockou. Druhá podkapitola sa zaoberá postupnosťami náhodných čísel pri hazardnej hre ruleta. V nej sa táto práca venuje vyhodnoteniu základných stratégii pri hraní hier, konkrétne pravdepodobnosti zisku a prehry pri každej stratégii, pravdepodobnosti nadobudnutia miliónového zisku a zisteniu predpokladaných zárobkov kasín pri hráčoch využívajúcich príslušné stratégie.

3.1 Vyhodnotenie náhodnosti hodov kocky

Existuje hneď niekoľko využití hracích kociek – od klasických spoločenských hier typu "Človeče nehnevaj sa", až po hazardné hry, v ktorých sa hrá o peniaze. Rovnako

ako je ich využitie rôzne, existuje aj niekoľko typov kociek. Niektoré hracie kocky majú ihlanový tvar, iné majú diamantový tvar a podobne. V našom experimente sa zaoberáme klasickou hracou kockou v tvare kocky – so šiestimi pravidelnými stenami v tvare štvorca a numerickým ohodnotením stien číslami od 1 po 6.

Za predpokladu, že hracia kocka nie je modifikovaná a hod kockou sa nedá zmanipulovať, je pravdepodobnosť hodu každého čísla rovná práve 1/6. Túto skutočnosť sme experimentálne otestovali programovým riešením generovania čísel 1 až 6 na grafickej karte Nvidia. Testovanie prebehlo postupným generovaním 10^{12} náhodných čísel, pričom pri približne každom 10^N vygenerovanom čísle bola zapísaná štatistická informácia a sledovaná stredná kvadratická odchýlka chybovosti pre každú číslicu od 1 po 6.

V nasledujúcej tabuľke 3.1 je výber z konkrétneho počtu vygenerovaných čísel pri generovaní sekvencie $10^N(2, ..., 12)$ náhodných čísel.

	10^{2}	10^{3}	10^{8}	10^{12}
1	16	216	16659732	166667163468
2	14	201	16668868	166666349278
3	23	202	16667091	166666701836
4	21	194	16667759	166666490630
5	21	204	16670499	166666646896
6	25	183	16673987	166666647892
Počet	120	1200	100007936	1000000000000

Tabuľka 3.1: Tabuľka počtu vygenerovaných čísel

Teoreticky by mal pomer počtu hodov konkrétnej číslice k celkovému počtu vygenerovaných čísel so zväčšujúcim sa počtom vygenerovaných čísel konvergovať práve k 1/6. Pre jednotlivé počty vygenerovaných čísel z tabuľky 3.1 sme vytvorili pomer k celkovému počtu vygenerovaných čísel v nasledujúcej tabuľke 3.2 a sledovali sme, či hodnota konverguje k 1/6.

Z pozorovania je zrejmé, že s väčším počtom vygenerovaných čísel získavame uniformnosť pri generovaní, čo môžeme ešte vizualizovať vo forme grafu. Z nameraných údajov sme zostrojili graf, ktorého krivky zobrazujú jednotlivé pomery pre čísla od 1 do 6. Os X je logaritmická a zobrazuje hodnoty zodpovedajúce počtu vygenerovaných čísel (10^2 , 10^3 ... 10^{12}). Os Y predstavuje hodnotu pomerov z

	10^{2}	10^{3}	10^{8}	10^{12}	
1	0,133333	0,18	0,166584	0,166666	
2	0,116667	1,1675	0,166675	0,166666	
3	0,191667	0,168333	0,166657	0,166666	
4	0,175	0,161667	0,166664	0,166666	
5	0,175	0,17	0,166691	0,166666	
6	0,208333	0,1525	0,166726	0,166666	
Skutočná	0.16666667				
pravdepodobnosť					

Tabuľka 3.2: Tabuľka konvergujúcich pomerov vygenerovaných čísel

tabuľky 3.2. Ako je z grafu možné vyčítať, už pri 10^6 vygenerovaných čísel je pomer každej číslice veľmi blízky matematickej pravdepodobnosti 1/6 a je len ťažko vidieť pokrok pri ďalšom zvyšovaní počtu generovaných čísel.



Obr. 3.1: Tabuľka konvergujúcich pomerov vygenerovaných čísel

Ďalšie zlepšenie možno pozorovať iným druhom grafu – strednou kvadratickou odchýlkou (angl. root mean squared error). Ide o rozptyl, ktorý sa vypočíta vzťahom 3.1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$
(3.1)

V tabuľke 3.3 sú zapísané hodnoty strednej kvadratickej odchýlky pre každú zodpovedajúcu číslicu od 1 po 6 pri zodpovedajúcom celkovom počte vygenerovaných čísel 10^N , kde 2 \leq N \leq 12. Z tabuľkových dát vieme povedať, že so zväčšujúcim sa počtom vygenerovaných čísel klesá hodnota strednej kvadratickej odchýlky. Pri počte 10^{12} vygenerovaných čísel je už táto hodnota zanedbateľná.

	10^{2}	10^{3}	10^{8}	10^{12}
1	0,001111	0,000178	6,81* 10 ⁻⁹	2,4* 10 ⁻¹³
2	0,0025	6,98* 10 ⁻⁷	7,7* 10 ⁻¹¹	1,01*10 ⁻¹³
3	0,000625	0,000057	8,1* 10 ⁻¹¹	1, 2* 10 ⁻¹⁵
4	0,000064	0,000027	5,3* 10 ⁻¹²	3,1* 10 ⁻¹⁴
5	0,000064	1,78* 10 ⁻⁶	6,2* 10 ⁻¹⁰	3,9* 10 ⁻¹⁶
6	0,001736	2,5 *10 ⁻⁷	3,59* 10 ⁻⁹	3,5* 10 ⁻¹⁶
RMSE	0,031914	0,008347	4,32* 10 ⁻⁵	2,5 *10 ⁻⁷

Tabuľka 3.3: Tabuľka strednej kvadratickej odchýlky čísel

Podobne ako v predošlých pozorovaniach môžeme z tabuľky 3.3 zostrojiť samotný graf 3.2 zobrazujúci krivku meniacej sa hodnoty strednej kvadratickej odchýlky, v ktorom je jasne vidieť znižujúcu sa chybovosť s narastajúcim počtom vygenerovaných čísel. Preto môžeme definitívne usúdiť, že so zväčšujúcim sa počtom generovaných čísel pomer skutočne konverguje k pravdepodobnostnej hodnote 1/6, čím sme potvrdili uniformnosť generovania čísel na grafickej karte prostredníctvom CUDA rámca.

3.2 Vyhodnotenie stratégii v hre ruleta

História hazardnej hry ruleta siaha až do 18-teho storočia, keď bola ruleta známa medzi francúzskou šľachtou. Úspech tejto hry možno pozorovať aj dnes. Ruleta je obľúbená medzi hazardnými hráčmi prevažne vo veľkých kasínach a v dnešnej



Obr. 3.2: Graf strednej kvadratickej odchýlky (RMSE)

dobe aj v online priestore. Princípom tejto hry je roztočenie hracieho poľa s guličkou. Hracie pole pozostáva z 37 čísel, pričom každé číslo má priradenú svoju farbu. V implementácii bola použitá európska verzia rulety, kde číslo 0 má zelenú farbu a ostatné čísla majú priradenú červenú alebo čiernu farbu nasledovne: **Čierna farba**: 2,4,6,8,10,11,13,15,17,20,22,24,26,28,29,31,33,35 **Červená farba**: 1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36

Ruleta je hazardná hra, v ktorej sa hrá o peniaze. V hre existujú rôzne typy stávok, ktoré sú aj rôzne peňažne ohodnotené. Napríklad stávka na jedno číslo vráti hráčovi v prípade výhry 36-násobok vkladu. Staviť sa môže aj na hranu, napríklad stávka na roh štyroch susediacich čísel vráti hráčovi 9-násobok vkladu. Výška výhry súvisí aj s pravdepodobnosťou úspechu tohto vkladu. Je zrejmé, že pravdepodobnosť, že padne práve 1 číslo z 37-mich možných je relatívne malá.

V rulete existujú aj takzvané vonkajšie stávky, v ktorých je pravdepodobnosť úspechu omnoho vyššia. Ide o najpopulárnejší spôsob vkladania peňazí v hre. Výška výhry je však omnoho nižšia. Napríklad, staviť sa môže na párnosť hodeného čísla – buď padne párne alebo nepárne číslo. Inou možnosťou je stávka na farbu – buď padne červená alebo čierna. Výška výhry je v oboch prípadoch dvojnásobná, keďže ide o skoro 50 percentnú šancu na výhru.

Pri implementácii sme sa zamerali na stávkovanie na jednu farbu – čiernu a riešenie sme otestovali na sto miliónoch stávok. Každý hráč začína s 1000\$ a hrá

maximálne 100 zatočení rulety. Prvotne začína s vkladom 1\$, ďalší vklad závisí od výhry/prehry a stratégie. Pre predstavu momentálne existuje na svete okolo 4 000 kasín [1]. Našim experimentom sme preto simulovali približne všetky zatočenia rulety na svete za 1 mesiac.

Otestovali sme nasledujúce stratégie:

- 1. Zdvojnásobenie prehry.
 - (a) Nemám na vklad, vložím všetko.
 - (b) Nemám na vklad, končím prehral som.
- 2. Zdvojnásobenie výhry.
 - (a) Greedy metóda vkladám dvojnásobok až do konca.
 - (b) Non-greedy metóda I. ak dosiahnem 100% zisk, začnem vkladať 1\$.
 - (c) Non-greedy metóda II. ak dosiahnem 50% zisk, začnem vkladať 1\$.
- 3. Konštantné vkladanie sumy 1\$.

3.2.1 Zdvojnásobovanie prehry

Očakávaným výsledkom pri zdvojnásobovaní prehry je zisk maximálne 1100\$, keďže hráč po výhre vkladá stále 1\$ a má 100 pokusov. Pokiaľ hráč prehrá, bude sa snažiť svoju prehranú sumu vyrovnať. Výhodou môže byť to, že prehranú sumu vo väčšine prípadov naozaj vyrovná. Nevýhodou je risk, že stratí celú svoju štartovaciu sumu a že môže získať len malú sumu.

Na nasledujúcom histograme 3.3 je možné vidieť počet koncových súm pri zdvojnásobovaní prehry. Z histogramu je možné vyčítať, že vo väčšine prípadov skončí hráč s profitom, ale je mnoho prípadov, kedy stratí celú svoju počiatočnú sumu. Maximálne dosiahol hráč sumu 1070 dolárov.

V prípade riskantnej stratégie nemám – vlož všetko sa podarilo niektorým hráčom vyšplhať na menej stratovú sumu – okolie súm 700\$-800\$, avšak niektorí stratili všetko, čo zobrazuje aj vyšší počet konečných súm 0.

Lepšie je to vidieť na histograme 3.4, kde je uvedené percentuálne porovnanie. Väčšina hráčov odchádza domov s výherným pocitom.



Obr. 3.3: Histogramy výsledných súm pri zdvojnásobovaní prehry



Obr. 3.4: Percentuálny podiel profitujúcich hráčov pri zdvojnásobovaní prehry

Výhra je však v tomto prípade malá. Vyšší zisk možno získať riskantnejšou stratégiou – vkladaním dvojnásobne vyhranej sumy.

3.2.2 Zdvojnásobovanie výhry

Očakávaným výsledkom pri zdvojnásobovaní výhry je strata maximálne 100\$, keďže hráč po prehre vkladá stále 1\$ a má 100 pokusov. Pokiaľ hráč vyhrá, bude sa snažiť svoju vyhranú sumu zdvojnásobiť. Výhodou môže byť to, že sa vyhraná suma naozaj zdvojnásobí. V extrémnych prípadoch sa môže niektorý z účastníkov stať milionárom. Nevýhodou je risk, že svoju vysokú výhru stratí a začne znova tam, kde začal.

Na nasledujúcom histograme 3.5 je možné vidieť počet koncových súm pri zdvojnásobovaní výhry. Z histogramu je možné vyčítať, že vo väčšine prípadov skončí hráč s malou stratou peňazí. Na druhej strane sa objavuje aj mnoho prípadov, kedy hráč svojú výhru x-násobne zúročí. Pri Greedy stratégii dosiahol hráč maximálnu sumu 67 109 820 dolárov. Pri 50% non-greedy prístupe dosiahol hráč



maximálne 2010 dolárov a pri 100% non-greedy prístupe maximálne 3020 dolárov.

Obr. 3.5: Histogramy výsledných súm pri zdvojnásobovaní výhry

Úspešnosť jednotlivých stratégii je lepšie možné vidieť na nasledujúcom obrázku 3.6, ktorý zobrazuje percentuálny pomer hráčov s profitom a stratových hráčov. V porovnaní so stratégiami *Nemám - končím* a *Nemám - vlož všetko*, je percentuálny podiel stratových hráčov veľmi citeľný. Vyplýva to z rizikovosti danej stratégie. Z hľadiska možného zisku sú tieto stratégie lepšie.

Pri greedy stratégii je šanca neúspechu až 98 percentná, ale je s ňou možné dosiahnúť nemalé zisky. V niektorých prípadoch je možné získať z vloženej sumy 1000\$ až 10000-násobok. Samozrejme, takáto situácia sa vyskytne len ojedinele. Pri Non-greedy prístupoch je risk menší, avšak počet stratových hráčov predstavuje rozmedzie medzi 94% až 96%. Dá sa povedať, že pri aplikovaní tejto stratégie v reálnom živote bude hráč skoro určite stratový a pre dosiahnutie zisku by musel mať šťastie.

Na histograme početnosti konečných súm 3.7 môžeme vidieť, že tzv. milionársky pocit zažije jeden človek zo 100 miliónov, čo zodpovedá skoro 0% pravdepodobnosti. Pre lepší prehľad je na nasledujúcom obrázku zobrazený celkový počet výhercov súm 10\$ až 10^7 \$.



Obr. 3.6: Percentuálny podiel profitujúcich hráčov pri zdvojnásobovaní výhry



Obr. 3.7: Histogram početnosti konečných súm pri zdvojnásobovaní výhry

3.2.3 Konštantné vkladanie sumy

Pre otestovanie klasickej stratégie, kedy sa chce hráč iba zabaviť, sme zvolili vkladanie konštantnej sumy 1\$. Teoreticky je možné touto stratégiou získať a zároveň aj prehrať pri 100 zatočeniach maximálne 100\$, čím je táto stratégia papierovo najmenej riziková, ale aj najmenej zisková.

Na obrázku 3.8 sa nachádza vizuálne zobrazenie percentuálneho počtu profitových a stratových hráčov, respektíve hráčov, ktorí majú na konci hry profit do 10\$ a nad 10\$.

Ako je možné z obrázku vyčítať, pomer stratových a profitových hráčov tu nie je až tak extrémne veľký, než pri ostatných rizikovejších stratégiách.



Obr. 3.8: Percentuálny podiel profitujúcich hráčov pri konštantnom vkladaní 1\$

3.2.4 Celkové príjmy kasína pri jednotlivých stratégiách

Na obrázku 3.9 sme zhrnuli možné príjmy kasína pri predpokladanej návštevnosti sto miliónov hráčov. Ak by mal každý hráč pred vstupom do kasína na konte 1000\$, môžu, v závislosti od stratégie, dosahovať príjmy kasína maximálne 10¹¹\$. Informáciu o tom, ako sú jednotlivé stratégie výhodné pre hráča a kasíno sme štatisticky overili a zobrazili v nasledujúcom histograme konečných príjmov kasína 3.9. Ako je možné z grafu vyčítať, kasíno dosiahlo profit pri každej jednej hráčskej stratégii. Veľkosť konečného zisku sa, samozrejme, odvíja od rizikovosti stratégie.

Z nášho experimentu môžeme skonštatovať, že najväčší úžitok dosiahlo kasíno pri stratégii zdvojnásobovania výhry a to aj napriek tomu, že jeden človek zo sto miliónov získal miliónový profit. Najmenší, aj keď stále pomerne vysoký zisk,



dosiahlo kasíno pri stratégii konštantného vkladania sumy 1\$.



4 Analýza chýb jednorozmerných modelov

Práca [11] sa zaoberala prepisom heliosferických modelov z klastrového prostredia, využivajúcich výpočtovu silu CPU, do prostredia CUDA s využitím grafickej karty Nvidia. Vo finálnej verzii bola táto implementácia úspešná a získala niekoľkonásobné zrýchlenie oproti dovtedajším CPU výpočtom. V prípade F-p modelu sa dosiahlo približne 7,5-násobné zrýchlenie oproti CPU a v prípade B-p algoritmu išlo až o skoro 87-násobné zrýchlenie. Následne sa začal klásť dôraz na prepis ďalších modelov do prostredia grafických kariet.

Postupným testovaním a overovacími procedúrami boli pozorované v konečných výstupných dátach nedostatky, ktoré budeme v tejto diplomovej práci analyzovať a ak to bude z implementačného hľadiska možné, tak aj experimentálne riešiť. V prípade nenájdenia riešenia týchto problémov budú vylúčené niektoré ich možné príčiny a kruh s pravdepodobnými príčinami sa zúži.

Spomedzi známych chýb prepísaného 1-dimenzionálneho modelu do prostredia Nvidia CUDA sa zameriame na nasledujúce dve pozorovania:

- 1. Pulzácia viditeľná v grafe pomeru F-p a B-p modelov pri malých časových krokoch výpočtu [5].
- 2. Štatistická chyba F-p metódy pri vyšších časových krokoch výpočtu [14].

4.1 Pulzácia

V práci [5] boli predstavené štyri možné príčiny pulzácie. Za prvú možnú príčinu bolo označené podozrenie na nedostatočne presné generovanie náhodných čísel

pri výpočtoch. Diplomová práca sa zaoberala pokusom o riešenie problému pomocou zmeny algoritmov generácie náhodných čísel. Vo výsledku experimentu bolo preukázané, že distribúcia náhodných čísel nemala za následok príznačnú pulzáciu.

Za ďalšie možné príčiny boli označené použitia optimalizačných prepínačov pri kompilácii programu, rozdiely medzi jednoduchou a dvojitou presnosťou a distribúcia simulácii jednotlivých energií. V našej diplomovej práci sa budeme zaoberať dvoma predpokladanými príčinami pulzačného javu. Prvou je optimalizácia využívania hardvérových prostriedkov GPU a druhou je použitie premenných s dvojitou presnosťou. Ako prvú a najjednoduchšiu zmenu si zvolíme odstránenie optimalizačného prepínača "use_fast_math", ktorým sa výpočtové úlohy programu optimalizujú na využívanie hardvérových prostriedkov grafickej karty. Druhým experimentom bude overenie vplyvu využívania premenných s jednoduchou presnosťou a implementačná zmena programu na využívanie premenných s dvojitou presnosťou.



Obr. 4.1: Graf pomeru F-p a B-p pri dt = 0.5s a $K_0 = 5 * 10^{22} cm^2/s$

Testovanie sa uskutoční vo výpočtovom systéme, v ktorom je použitá grafická karta Nvidia GeForce 1080Ti. V prvom rade sa odmeria doba behu programu bez zmeny prepínačov a typu premenných. Experiment sa uskutoční s dĺžkou kroku dt=0.5s, rýchlosťou slnečného vetra V = 400 km/s, difúznym koeficientom $K_0 = 5 * 10^{22} cm^2/s$ v počte N=30 miliárd simulácii. Pre získanie početnejšej mno-
žiny dát a zníženie štatistickej chyby (odstránenie šumu) obmedzíme hodnoty rozsahu injektovanej kinetickej energie častíc do rozmedzia medzi 45GeV a 57GeV. Je nutné poznamenať, že v našom numerickom experimente očakávame enormné spomalenie vykonávania výpočtu, v dôsledku čoho by sme bez obmedzenia rozsahu injektovaných energií nevedeli náš experiment vykonať v rozumnom čase a úspešne vyhodnotiť na výstupnej vzorke dát bez šumu. Následne bude odmeraná dĺžka behu programu s odstráneným optimalizačným prepínačom. Zároveň sa vygeneruje a porovná graf pomeru dát z F-p a B-p metódy. Na základe grafu budeme sledovať prítomnosť pulzácie vyskytujúcej sa približne pri hodnote kinetickej energie 50GeV. Rovnaký postup uskutočníme v experimente so zmenou typu premenných z jednoduchej presnosti do premenných s dvojitou presnosťou.



Obr. 4.2: Spektrá na 1AU vyhodnotené Monte Carlo metódou (červená) a Crank-Nicolson metódou (čierna) pomocou F-p a F-T metódy Obrázok založený na: [14].

Testovanie pre uvedený rozsah energií pri F-p metóde je potrebné, pretože častice kozmického žiarenia pri prechode heliosférou strácajú svoju energiu. Častice nainjektované do heliosféry s energiou T_{inj} majú na 1AU nižšie energie s distribúciou, ktorej šírka je menšia a bližšia T_{inj} pri vyšších energiách. Túto skutočnosť možno pozorovať na obrázku 4.2. Pre testovanie výsledku vo vymedzenom rozsahu energií spektra sú preto vhodnejšie vyššie energie, kde je výpočet aj rýchlejší. Vypočítaný rozsah prinesie výsledok na 1AU, ktorý pri jeho počiatočných a konečných energiách nebude úplný. Budú v ňom chýbať častice injektované s energiami mimo injekčného rozsahu. To sa prejaví efektom odrezania pri najnižších a najvyšších energiách testovaného rozsahu. Porovnaním výsledku s B-p metódou sa určí, v akom rozsahu energií z injektovaného rozsahu je výsledok F-p metódy korektný bez odrezaní.

4.1.1 Beh programu bezo zmien s vymedzenou injekciou kinetickej energie

V zdrojovom kóde pôvodného 1-Dimenzionálneho F-p programu sme modifikovali metódu generovania injektovanej kinetickej energie častice, v ktorej sme obmedzili počiatočnú hodnotu na rozmedzie od 45GeV do 57GeV. Okrem tejto zmeny nebol program menený. Ide teda o verziu programu s optimalizačným prepínačom a s premennými s jednoduchou presnosťou. Spomínaný výpočet, ktorý sme spustili na počte 30 miliárd simulácii, bude východiskovým modelom pri porovnávaní ďalších experimentov. Rozsah energií 45GeV až 57GeV bol vybraný, pretože simulácie trvajú pri vyšších energiách kratšie a v tomto rozsahu sa vyskytuje pulzácia pri energii približne 50GeV.

Čas vykonávania programu bol odmeraný a určený na 15 minút. Zároveň sme vytvorili graf pomeru F-p a B-p dát. Výsledný graf je možné vidieť na obrázku 4.3. Z obrázku možno usúdiť, že v našich experimentoch sa stačí zamerať priamo na pulzáciu vyskytujúcu sa v rozmedzí energií 50GeV a 52GeV.



Obr. 4.3: Graf pomeru F-p a B-p pri $dt = 0.5s, k0 = 5*10^{22} cm^2/s$ v experimente pre TKin <45GeV,57GeV>

4.1.2 Odstránenie optimalizačného prepínača

Implementácia tejto zmeny pozostáva z odstránenia prepínača use-fast-math z kompilačného príkazu. Súčasná implementácia využíva tento prepínač pre zrýchlenie výpočtu aproximáciou medzivýsledkov pri výpočtovom procese. Pri pokuse s kompiláciou CUDA programu bez prepínača očakávame spomalenie behu programu. Zároveň môžeme predpokladať, že pulzácia sa týmto krokom môže odstrániť z dôvodu nahradenia aproximácie výsledkov na zaokrúhľovanie podľa normy IEEE 754.

Vo výsledku sa dĺžka behu programu predĺžila skoro 100-násobne z 15 minút na 1 deň a 59 minút. V tomto prípade je možné vidieť veľký vplyv prepínačov a optimalizácie na rýchlosť vykonávania úloh v grafických procesoroch. Pri väčšine grafických operácii sa využívajú celé čísla, ktoré grafická karta dokáže spracovať na svojich jadrách veľmi rýchlo. Pre priblíženie rýchlosti so spracovaním aj s formátmi s pohyblivou rádovou čiarkou sa snaží kompilátor s použitím spomínaného prepínača nahradzovať matematické funkcie použité v zdrojovom kóde s hardvérovými prostriedkami grafickej karty. Nanešťastie sa vykonávanie kódu bez optimalizácie musí vypočítavať softvérovo a pri ohromnom množstve simulácii, ktoré sa v heliosferických výpočtoch využívajú, sa beh takéhoto programu adekvátne spomalí.

Odhliadnuc od rýchlosti vykonávania sme na obrázku 4.4 zobrazili graf pomeru F-p a B-p dát v rozmedzí injektovanej kinetickej energie od 45GeV do 57GeV. Z grafu je možné vidieť nepriaznivý výsledok, keďže sa približne pri 50GeV výskyt pulzácie touto jednoduchou zmenou neodstránil.

Numerickým experimentom sme zistili, že vplyv optimalizačného prepínača –use-fast-math nemá vplyv na problém s pulzáciou, naopak jeho vynechaním sa hlavná myšlienka rýchlejšieho vykonávania programu na GPU oproti vykonávania na CPU nenaplňuje.

4.1.3 Zmena jednoduchej presnosti na dvojitú

Ako druhú implementáciu volíme zmenu premenných F-p metódy programu z pôvodnej jednoduchej presnosti float na premenné s dvojitou presnosťou double.



Obr. 4.4: Graf pomeru F-p a B-p pri dt = 0.5s, $k0 = 5 * 10^{22} cm^2/s$ v experimente pri odstránení prepínača pre TKin <45GeV,57GeV>

Zmena typu bude prevedená na všetkých premenných, nezávisle na ich použití. Vychádzajúc z architektúry použitej grafickej karty Nvidia 1080Ti očakávame pri tomto experimente ešte väčšie spomalenie behu programu než to bolo pri odstránení optimalizačného prepínača. Zatiaľ čo výrobca grafickej karty uvádza výkon karty pri používaní 32-bitovej jednoduchej presnosti v hodnote 11.34 TFLOPS, pri 64-bitovej dvojitej presnosti je to iba 354.4 GFLOPS.

Znenie vyšetrovanej hypotézy môže byť nasledujúce: Príčina pulzácie pri nízkej hodnote časového kroku vyplýva z nedostatočnej presnosti a je spôsobená použitím jednoduchej presnosti premenných pri simuláciách.

Vo výsledku sa dĺžka behu programu predĺžila skoro trojnásobne než to bolo pri experimente s odstránením prepínača. Z vykonávacej doby 15 minút bola doba behu programu 3 dni a 8 hodín. Na obrázku 4.5 je uvedený pomer F-p a B-p dát, kde si je možné všimnúť zmiznutie pulzu pri hodnote 51GeV.

Pri testovacej vzorke 30 miliárd simulácii je však vzorka zašumená, preto sme vykonali testovanie na vzorke 300 miliárd injektovaných častíc o celkovej dobe vykonávania 33 dní. Experiment bol vykonaný v 10 iteráciách po 30 miliárd injektovaných častíc. Vstupnými parametrami boli hodnoty slnečného vetra V =400km/s, difúzny koeficient $K0 = 5 * 10^{22} cm^2/s$ a časový krok dt = 0.5s. Vý-



Obr. 4.5: Graf pomeru F-p a B-p pri dt = 0.5s, $k0 = 5 * 10^{22} cm^2/s$ v experimente pri použití dvojitej prenosti pre TKin <45GeV,57GeV> na vzorke 30mld

sledný súbor z experimentu dosiahol veľkosť 1.78GB. Výsledok z experimentu je predstavený na nasledujúcom obrázku 4.6.



Experiment Graf pomeru FP/BP na datach dt=0.5, k=5e22 medzi TKin<45,57> (N300/N10) optimalizovane, double

Obr. 4.6: Graf pomeru F-p a B-p pri dt = 0.5s, $k0 = 5 * 10^{22} cm^2/s$ v experimente pri použití dvojitej prenosti pre TKin <45GeV,57GeV> na vzorke 300mld

Z grafu je možné vidieť, že hypotéza, ktorá predpokladala, že príčinu pulzácie spôsobuje jednoduchá presnosť pri výpočte, sa potvrdila.

4.2 Štatistická chyba F-p metódy

Vychádzajúc z obrázku 4.7 JGR článku [14] pozorujeme systematickú chybu pri F-p metóde v krivke pomeru spektier na 1AU získaných F-p a Crank-Nicholson (CN) metódami pri hodnotách časového kroku dt väčších ako 2 sekundy. V prípade pomeru metódy B-p a CN môžeme sledovať takmer konštantný priebeh krivky grafu, zatiaľ čo pri metóde F-p sa krivka v značnej miere zakrivuje a sledujeme pri nej chybu +-5%. Na obrázku 4.8 môžeme vidieť vykreslenú krivku pomeru výsledných spektier zo súčasnej F-p metódy so štatistikou 400 miliárd simulácii injektovaných častíc s výsledkami referenčnej B-p metódy so štatistikou o veľkosti 10 miliárd simulácii. Z priebehu krivky dostávame spomínanú chybu z článku [14], ktorú sa budeme našimi testami snažiť vyriešiť.



Obr. 4.7: Pomer medzi Monte-Carlo a Crank-Nicholson metódami. Vľavo pre B-p a F-p, vpravo pre B-T a F-T

Obrázok založený na: [14].

Na obrázku je výsledok simulácii pre vstupné parametre rýchlosti slnečného vetra V = 400 km/s a difúzneho koeficientu $K0 = 5 * 10^{22} cm^2/s$.

V našich experimentoch sa budeme snažiť tento rozdiel medzi F-p a B-p metódou vyriešiť pomocou zmeny pozície injektovanej častice v simuláciách. Skúmanou hypotézou je, že injekcia častíc sa nenachádza v počiatočnej vzdialenosti rovnej presne hodnote r_0 =100AU, ale má svoju distribúciu. Jej distribúcia je podobná tej, akú má metóda B-p. Chceme preveriť, či takáto rozdielna injekcia častíc do heliosféry nepovedie k výsledkom bez chyby tzv. základnej vlnovky z článku JGR [14]. Chyba základnej vlnovky je rozdielom medzi F-p a B-p metódami vý-



Obr. 4.8: Graf pomeru F-p a B-p pri dt=50s a $k = 5 * 10^{22} cm^2/s$

počtu, kde pomer ich spektier má tvar podobný vlnovke.

Ako referenčné hodnoty distribúcie častíc za hranicou heliosféry budeme brať výsledky z B-p metódy o veľkosti 10 miliárd simulácii. V prípade F-p testovacích metód budeme používať vstupné testovacie parametre V = 400 km/s, dt = 50s a $k = 5 * 10^{22} cm^2/s$.

V prvom rade si analyzujeme výsledky simulácie z B-p metódy. Na nasledujúcich obrázkoch 4.9 a 4.10 je možné vidieť histogramy početností injektovaných vzdialeností r_0 pri hodnotách kinetickej energie v rozmedzí 0.9GeV až 1.1GeV a 9GeV až 11GeV. Histogramy sme doplnili aj o krivku Gaussovej funkcie, ktorej parametre sú uvedené na pravej časti obrázkov. Gaussova funkcia nám slúži ako fit pre distribúciu vzdialeností injektovaných častíc. Poznamenajme, že v B-p metóde, ktorá prebieha späť v čase, ide o distribúciu častíc po opustení heliosféry. Táto distribúcia realistickejšie opisuje injekciu častíc do heliosféry, ktorá je v základnom F-p modeli daná jednou hodnotou, injekciou na hranici heliosféry.

V oboch prípadoch vidíme klesajúcu krivku počiatočných hodnôt vzdialeností injektovaných častíc. Pri testovaniach s F-p modelmi budeme sledovať správnosť distribúcie častíc v podobe pomeru výskytov počiatočných vzdialeností častíc z výsledkov testovaných metód s referenčnými početnosťami uvedených dát z B-p simulácie.



Obr. 4.9: Histogram B-p simulácie pre N=10, dt = 50s, $k0 = 5 * 10^{22} cm^2/s$ a V = 400 km/s pre TKin <0.9GeV,1.1GeV>



Histogram B-p data N=10, dt=50, k=5e22, V=400 pre Tkin <9.0,11.0>

Obr. 4.10: Histogram B-p simulácie pre N=10, dt = 50s, $k0 = 5 * 10^{22} cm^2/s$ a V = 400 km/s pre TKin <9GeV,11GeV>

4.2.1 Prvá verzia injekcie častíc

Tento test sme pomenovali názvom injection-modif-v1. Zmena v zdrojovom kóde spočíva vo vygenerovaní náhodnej počiatočnej hodnoty r_0 s Gaussovou distribúciou za hranicou 100AU. Pre generovanie počiatočnej pozície častice sme použili

vzorec 4.1

$$r_0 = r_0 + (\sqrt{2 * k * dt} * R) \tag{4.1}$$

kde $r_0 = 100AU$, k je difúzny koeficient k = beta*k0*P, kde beta je rýchlosť častice v jednotkách rýchlosti svetla a P je rigidita častice v GV. Rigidita je úmerná hybnosti častice na jednotku náboja P = (p*c)/Ze, kde p je hybnosť častice a e je elementárny náboj, Z je protónové číslo. R je náhodné číslo s Gaussovou distribúciou.

Uvedeným vzorcom pre modifikáciu a prispôsobením hraničných hodnôt sme zmenili injektovanie častice z pôvodnej vzdialenosti 100.01AU do intervalu v rozmedzí 100AU a väčšími vzdialenosťami. Našim experimentom sledujeme overenie, či príde k zmene v krivke pomeru spektier F-p a B-p na 1AU. Pre test použijeme výpočet so štatistikou 400 miliárd injektovaných častíc s časovým krokom dt = 50. V prípade vygenerovania hodnoty počiatočnej vzdialenosti menšej ako 100AU výpočet zahadzujeme/nerealizujeme.

Uvedenou modifikáciou sme získali výstupný súbor zaznamenaných hodnôt o veľkosti 1.7GB za vykonávací čas 4.5 hodiny. Na obrázku 4.11 je zobrazený graf pomeru hodnôt F-p a B-p metódy.



Obr. 4.11: Graf pomeru F-p/B-p pre modifikovanú verziu 1

Tvar získaného pomeru výsledkov F-p a B-p metódy, zobrazeného na obrázku 4.11, je veľmi podobný referenčnému grafu uvedeného v JGR článku. Injekcia 1

neovplyvnila výskyt chyby s vlnovkou. To nás viedlo k úprave injekcie na tvar, ktorý zahŕňa aj konvekciu slnečného vetra.

4.2.2 Druhá verzia injekcie častíc

Druhý pokus sme pomenovali názvom injection-modif-v2. V tomto prípade spočíva zmena programu v pridaní hodnoty rýchlosti slnečného vetra V vynásobenej časovým krokom dt. Použitý výsledný injekčný vzorec je uvedený v 4.2.

$$r_0 = r_0 + (\sqrt{2 * k * dt} * R) + (V * dt)$$
(4.2)

Rovnako ako pri prvej verzii je injektovanie vymedzené do intervalu od 100AU, pričom výpočty s nižšími hodnotami zahadzujeme. Výpočet s rovnakými parametrami, ako boli použité pri prvej verzii, trval približne 4.5 hodiny a výsledný súbor mal veľkosť 1.7GB. Na obrázku 4.12 je prezentovaný graf pomeru modifikovanej F-p metódy a referenčných výsledkov z B-p metódy.



Obr. 4.12: Graf pomeru F-p/B-p pre modifikovanú verziu 2

Z uvedeného grafu je možné vidieť, že prítomnosť chyby je rovnaká ako pri prvej verzii programu, tak aj pri pôvodnej verzii použitej v článku JGR.

4.2.3 Tretia verzia injekcie častíc

Ako posledný test s názvom injection-modif-v3 sme doplnili injekčný vzorec z druhej verzie o konštantnú hodnotu. Plné znenie vzorca je uvedené v 4.3.

$$r_0 = r_0 + \left(\sqrt{2 * k * dt} * R\right) + \left(V * dt\right) + \left(\frac{2 * k * dt}{100}\right)$$
(4.3)

Výsledný súbor so zapísanými hodnotami trval podobne ako pri predošlých dvoch pokusoch 4.5 hodín a mal veľkosť 1.7GB. Pomer výsledkov tretej modifikovanej verzie s referenčnými hodnotami B-p metódy je uvedený na nasledujúcom obrázku 4.13.



Obr. 4.13: Graf pomeru F-p/B-p pre modifikovanú verziu 3

4.2.4 Zhrnutie

Pre finálne zhodnotenie výsledkov sa najprv pozrime na odchýlku početnosti distribuovaných častíc z hľadiska ich počiatočnej vzdialenosti od Slnka pri všetkých troch vykonaných testoch oproti distribúcii pri referenčnej B-p metóde. Na obrázku 4.15 je možné vidieť grafy pomerov pri energiách od 9GeV do 11GeV a na obrázku 4.14 sú uvedené krivky pomerov pri energiách od 0.9GeV po 1.1GeV. Rozdiel v distribúcii počiatočných hodnôt je hlavne pri menších vzdialenostiach malý a preto môžeme považovať naše cieľové testovacie vzorky za správne.



Obr. 4.14: Graf pomerov F-p/B-p v modifikovanej verzii 1, 2 a 3 pre TKin<0.9GeV, 1.1GeV>



Obr. 4.15: Graf pomerov F-p/B-p v modifikovanej verzii 1, 2 a 3 pre TKin<9GeV, 11GeV>

Berúc v úvahu nemeniace sa krivky pomerov hodnôt F-p a B-p metód pri všetkých troch testovaniach teda môžeme v závere dedukovať, že táto jednoduchá zmena v podobe zmeny hodnoty počiatočnej vzdialenosti injektovanej častice nemá vplyv na riešenie tohto problému. Naša hypotéza sa nepotvrdila ako platná.

5 Reprodukcia Pei-ovho modelu

Pre overenie správnosti implementácie 2-Dimenzionálnych modelov do CUDA sme sa rozhodli reprodukovať 2D model a stochastický prístup pre riešenie Parkerovej rovnice z článku [21]. V článku sú uvedené vzorce, ktoré autori použili pre definovanie modelu, ktoré sme použili pre vytvorenie nášho kódu tohto modelu. Cieľom tejto práce je snaha o verifikáciu, respektíve, o prípadné opravenie chýb nášho 2D F-p modelu, ktoré by v prípade modelu z Pei-ovho článku mohli zmiznúť. V nasledujúcich podkapitolách si ukážeme rozdiely medzi súčasným 2D F-T modelom, ktorý bol opísaný v práci [5], ako aj problémy, ktoré sa v priebehu nášho experimentu objavili a boli opravené.

5.1 Prepis Pei-ovho modelu

Pri prepise Pei-ovho modelu sme zvolili iteratívny prístup k zmenám výpočtového kódu. V prvej iterácii sme zaviedli hneď niekoľko zmien v kóde. Prvou zmenou bolo prepísanie kódu z F-T do F-p verzie, pričom v kóde boli odstránené všetky výskyty mu a LBT. Naopak, v implementácii sme zaviedli heliosferickú šírku Θ s nulovou hodnotou na póloch heliosféry a 90 stupňami na rovníku v ekliptike. Heliosferická šírka Θ je generovaná metódou Monte-Carlo v rozmedzí hodnôt od 0 po $\pi/2$, obdobne ako to bolo pri generovaní mu.

V modeli nám pribudla Γ podľa vzorca 5.1 a zmenilo sa aj magnetické pole Bfactor podľa vzorca 5.2.

$$\Gamma = \frac{r * \Omega}{V} * \sin \Theta \tag{5.1}$$

$$Bfactor = \frac{B_e}{B} = \frac{1}{r^2} * \frac{\sqrt{1+\Gamma^2}}{\sqrt{1+\frac{\Omega}{V}}}$$
(5.2)

Zmena nastala aj K_{rr} (5.3), K_{par} (5.4), K_{per} (5.5) a v $K_{\Theta\Theta}$ (5.6).

$$K_{rr} = K_{perp} + (K_{per} - K_{perp}) * \frac{1}{1 + \Gamma^2}$$
(5.3)

$$K_{par} = K_0 * \beta * P * \frac{B_e}{B}$$
(5.4)

$$K_{per} = 0.1 * K_{par}$$
 (5.5)

$$K_{\Theta\Theta} = K_{perp} \tag{5.6}$$

V prvej verzii nepočítame so zmenou K_{rr} a preto $dK_{rr} = 0$. Zmeny v Δr a $\Delta \Theta$ sú uvedené na nasledujúcich vzorcoch 5.7 a 5.8.

$$\Delta r = \frac{2 * K_{rr}}{r} + V + R_1 * \sqrt{2 * K_{rr}}$$
(5.7)

$$\Delta\Theta = \frac{K_{\Theta\Theta} * \cos\Theta}{r^2 * \sin\Theta} + \frac{1}{r} * R_2 * \sqrt{2 * K_{\Theta\Theta} * \Delta t}$$
(5.8)

Správnosť implementácie v prostredí CUDA sme porovnali s výsledkami výpočtov na CPU, ktorých pomery pre časový krok 0,5s a 2s je možné vidieť na nasledujúcich obrázkoch 5.1 a 5.2. Krivka pomeru hovorí o odlišnosti výpočtov na CPU a GPU. Z grafov vidíme, že pomer sa pohybuje v blízkosti jednotky. Môžeme teda povedať, že implementácia na GPU je správna a mohli sme tak spustiť výpočty na veľkom počte dát.

Pre sledovanie správnosti modelu sme v CUDA implementácii pridali sledovanie, na základe ktorého sme na konci behu programu vygenerovali tepelnú mapu. Táto funkcionalita sa ujala ako úspešná a predstavuje v našom systéme automatizované generovanie máp pre 2D modely prostredníctvom prepínača v kóde.

Vytváranie mapy spočíva v zaznamenávaní výskytu častice na pozíciách v rozsahu parametrov r = <0, 100>, Θ = <0, 180> a Tinj = <0, 100>. Pri výskyte



Obr. 5.1: Graf pomerov Pei CPU/GPU pre dt=0.5s



Obr. 5.2: Graf pomerov Pei CPU/GPU pre dt=2s

inkrementujeme prítomnosť na pozícii $index = MAX_{\Theta} * MAX_R * map_{Tinj} + MAX_R * map_{\Theta} + map_r$. Výsledná tepelná mapa z výpočtu pre kinetickú energiu $T_{Kin} = 1 GeV$ je zobrazená na obrázku 5.4 a pre kinetickú energiu $T_{Kin} = 50 GeV$ na obrázku 5.3.

Pri prvej verzii programu sme vykonávali zápis výstupných dát s lokálnym medzihviezdným spektrom LIS podľa Yamadu. V Pei-ovom článku [21] sa však používa iné LIS – Webber Higbie. Pre jej vykonanie potrebujeme ukladať údaje ako o výslednej heliošírke Θ na 1AU, tak aj o injektovanej heliošírke Θ na 100AU.



Obr. 5.3: Heatmapa pre 50GeV



Obr. 5.4: Heatmapa pre 1GeV

Vyhodnocovanie s LIS spektrom Webber Higbie sa uskutočňuje nasledujúcim vzorcom 5.9:

$$J_{lis} = \frac{21.1 * e^{-2.8 * \log(Tkin_{inj})}}{1 + (5.85 * e^{-1.22 * \log(Tkin_{inj})}) + (1.18 * e^{-2.54 * \log(Tkin_{inj})})}$$
(5.9)

Vzorec udáva intenzitu častíc v jednotkách počet častíc na m^2 s sr MeV. Pri získaných výsledkoch sme zistili, že prvá verzia programu nebola úplne rovnaká ako verzia v publikovanom článku [21] a snažili sme sa ďalej modifikovať model v druhej verzii programu. Ako prvé sme zmenili hodnotu uhlovej rýchlosti rotácie Slnka omega z pôvodných 2.693e-6 na hodnotu 2.863e-6, keďže sa v článku používa rotácia slnka za 27 dní v radiánoch za sekundu a my sme používali hodnotu 25 dní. 27 dní predstavuje strednú hodnotu diferenciálnej rotácie Slnka a 25 dní je hodnota na Slnečnom rovníku.

Ako ďalšie sme v druhej verzii zaviedli injekciu theta do rozmedzia od 0 až π (oproti predošlej 0 až $\pi/2$). Väčšou zmenou je výpočet BFactoru, ktorý sa zmenil na vzorec 5.10.

$$Bfactor = \frac{B_e}{B} = \frac{r^2}{\sqrt{1+\Gamma^2}}$$
(5.10)

Poslednou zmenou bola oprava chyby v prvom sčítanci $\Delta \Theta$ doplnením o chýbajúci násobok časového kroku Δt , ktorého tvar je uvedený v 5.11.

$$\Delta\Theta = \frac{K_{\Theta\Theta} * \cos(\Theta)}{r^2 * \sin(\Theta)} * \Delta t + \left(\frac{R_2 * \sqrt{2 * K_{\Theta\Theta} * \Delta t}}{r}\right)$$
(5.11)

Ukázalo sa, že druhá verzia sa bližšie podobá k Pei-ovmu modelu, avšak ani zďaleka nedosahuje úplne rovnaké výsledky. V tretej verzii programu sme zaviedli aj dK_{rr} , keďže sme vo verziách 1 a 2 pracovali s $dK_{rr} = 0$. Keďže sme testovali možnosť, že výsledky v Pei-ovom článku boli počítané s nulovou hodnotou dkrr, čo sme vyvodili z jeho citácie na tvar SDE rovníc v článku [27]. Vzorce pre výpočet dKrr sme odvodili v nasledujúcej forme 5.13. Hodnota ν je substitučná premenná pre vzťah 5.12.

$$\nu = \frac{2r\Omega^2(\sin\theta)^2}{V^2} \tag{5.12}$$

$$dK_{rr} = \frac{0.1K_0\beta Rg((2r\sqrt{1+\gamma^2}) - \frac{r^2\nu}{2\sqrt{1+\gamma^2}})}{1+\gamma^2} + \frac{0.9K_0\beta Rg((2r(1+\gamma^2)^{\frac{3}{2}})) - (\frac{r^2\nu^3\sqrt{1+\gamma^2}}{2})}{(1+\gamma^2)^3}$$
(5.13)

Zmena dK_{rr} sa prejaví aj vo vzorci pre výpočet $\Delta\Theta$ 5.14:

$$\Delta\Theta = \left(\frac{K_{\Theta\Theta} * \cos(\Theta)}{r^2 * \sin(\Theta)} * \Delta t\right) + \left(\Delta K_{\Theta\Theta} * \frac{\Delta t}{r^2}\right) + \left(\frac{R_2 * \sqrt{2 * K_{\Theta\Theta} * \Delta t}}{r}\right)$$
(5.14)

V rámci kontroly odvodených vzťahov sme ukázali, že tieto vzorce sú ekvivalentné vzorcom v [15] vo verzii standard 2D modelu. Po implementovaní zmien sme sa snažili priblížiť vstupné parametre a nastavenia modelu čo najviac Peiovmu modelu. Viaceré vstupné parametre však Pei vo svojom článku neuvádza. Príkladom je rýchlosť slnečného vetra. Tú sme preto skúsili nastaviť na najpravdepodobnejšiu hodnotu 400 km/s. Pri rýchlosti 400 km/s sme sa veľmi nepriblížili Pei-ovej krivke. Teoreticky sme sa domnievali, že s vyššou hodnotou rýchlosti slnečného vetra sa priblížime k Pei-ovej krivke bližšie. To sa na výpočte na GPU nepotvrdilo, čo vzbudzovalo dojem, že prepis heliosférických výpočtov do GPU prostredia bol nesprávny. Overiť sme to chceli sledovaním častice v priebehu výpočtov. Sledovať sme začali hodnoty koordinácií r, Θ, ID procesu a derivácii difúzneho koeficientu dK_{rr} a $dK_{\Theta\Theta}$. Pri sledovaní častíc a zápise na disk bolo potrebné si dávať veľký pozor na veľkosť súboru. Ľahko sa môže stať, že sa disk a pamäť preplní, keďže výpočet prebieha v miliardách behov. Vybrali sme preto iba prvých 1 milión krokov výpočtu trajektórie častíc a vymedzili 5 stupňové rozmedzie Θ medzi 85 až 90 stupňov a r medzi 10AU a 11AU. Výsledný súbor pre 1 milión stavov dosahoval veľkosť 600MB.

Rovnaký model sme otestovali aj na CPU. Samozrejme, rýchlosť výpočtu je v porovnaní s GPU prepísanými modelmi značne pomalšia a pri výpočte na jednom jadre sme boli nútení skrátiť počet injektovaných častíc pri testovacom výpočte na 1 miliardu. Ukázalo sa však, že pomer CPU a GPU dát bol veľmi blízky jednotke, čo svedčilo o zanedbateľnom rozdiele medzi výsledkami z paralelného kódu a sekvenčného kódu.

5.2 Vyhodnotenie 2D modelu

Uvedené dáta z článku [21] sme zobrazili v grafe na obrázku 5.5 vo forme bodov. Podľa informácií uvedených v článku bola pri výpočtoch použitá hodnota difúzneho koeficientu $K_0 = 2 * 10^{23} cm^2/s$. Táto je v porovnaní s bežným rozsahom hodnôt difúzneho koeficientu nezvyčajne vysoká. Prepísaný model na základe uvedených matematických rovníc sme experimentálne overili na výpočte s parametrami $K_0 = 2*10^{23} cm^2/s$, dt = 5s, V = 400 km/s a s počtom injektovaných častíc 10 miliárd. Výsledné dáta pre $K_0 = 2*10^{23} cm^2/s$ sme v obrázku 5.5 vizualizovali modrou farbou. Porovnaním s uvedenými hodnotami v článku vidíme, že sme sa nepriblížili k želanej krivke.



Multigraf WH 1e3bin 2D Pei modelu pre rozne parametre na CPU/GPU (v12)

Obr. 5.5: Vizualizované dáta z výpočtového modelu

Vytvorili sme hypotézu, v ktorej sme sa domnievali, že pri písaní článku nastala chyba a hodnota difúzneho koeficientu K_0 by mala byť nižšia. Experiment sme otestovali s parametrami $K_0 = 2 * 10^{22} cm^2/s$, dt = 5s, V = 400 km/s a s počtom injektovaných častíc 100 miliárd. Dáta sme vizualizovali červenou farbou na obrázku 5.5. Ako je možné vidieť, naša hypotéza sa potvrdila a so znížením hodnoty difúzneho koeficientu sme získali veľmi podobnú krivku, ktorá je blízka uvedenej krivke v článku [21]. Uvedenými skutočnosťami sme teda overili správnosť prepísaného modelu ako aj poukázali na možnú chybu v uvedenom článku.

6 Ďalšie rozšírenie výpočtového systému heliosferických výpočtov

Existujúce riešenie pre paralelné spracovanie heliosferických výpočtov podporuje implementácie jednorozmerných a dvojrozmerných F-p a B-p modelov. Implementáciu dvojrozmerného F-p modelu podľa článku [21] sme rozoberali v kapitole 5. Model sme pridali do aktuálnej verzie systému a napojili sme potrebné komponenty do konzolového prostredia pre spúšťanie výpočtových modelov. Pri zlučovaní zmien sme kládli dôraz na zachovanie existujúcej štruktúry projektu ako aj na zachovanie menných konvencií. Okrem toho sme pri pridávaní modelu aktualizovali zoznam vzájomne sa vylučujúcich kombinácii prepínačov, ktoré by v prípade súčasného použitia mohli spôsobovať problémy.

Ďalším z kľúčových bodov tejto diplomovej práce je vytvorenie systému pre automatizáciu výpočtov pre jednorozmerné a dvojrozmerné modely a príprava systému na implementovanie samooptimalizačného rámca. Automatizácia výpočtov má hlavne slúžiť na zjednodušenie práce so spúšťaním heliosferických výpočtov pre výskumníkov ústavu experimentálnej fyziky SAV, ale neskôr aj pre širšiu odbornú verejnosť, keďže sa systém plánuje zverejniť a v obmedzenom režime aj sprístupniť verejnosti.

6.1 Požiadavky systému

Princípom softvérového riešenia pre samoadaptačný a samooptimalizačný systém je idea spustenia množiny malých výpočtov s rôznymi parametrami, na základe ktorých sa vyberie najoptimálnejšia kombinácia parametrov. Výsledná kombinácia parametrov bude následne použitá pre spustenie jedného veľkého výpočtu s veľkým počtom injektovaných častíc, ktorý trvá niekoľkonásobne dlhšiu dobu, než kontrolné výpočty. Výber správnych parametrov sa bude vykonávať na základe vopred definovaných vzorcov, ktoré budú implementované do matematickej logiky súčasného systému. Zhrnutie tohto princípu je znázornené na nasledujúcom obrázku 6.1.



Obr. 6.1: Princíp fungovania samoadaptačného a samooptimalizačného systému

Architektúra systému pre samoadaptačný systém zahŕňa podporu spúštania viacnásobných výpočtov s rôznymi parametrami, ktoré získa systém na vstupe od používateľa. Pre zabezpečenie správnej funkčnosti systému je nutné navrhnúť a implementovať CLI parser pre jednotlivé používateľské vstupy. Parser musí vstupy kategorizovať na základe príznaku pre multivýpočet. V prípade, že budú na vstupe zadané parametre iba pre jeden výpočet, musí systém túto skutočnosť rozpoznať a vykonať originálnu funkcionalitu systému.

Program na základe parametrov bude spúšťať jednotlivé podvýpočty, pričom výsledky a informácie o behu programu sa budú zapisovať do samostatne pripravených podadresárov, ktoré sú vytvorené systémom heliosférických výpočtov pred spustením každého podvýpočtu.

Po dokončení všetkých kontrolných výpočtov prebehne samooptimalizácia na základe vyexportovaných dát. Dáta budú vyhodnotené na základe vopred definovaného vyhodnocovacieho vzorca, ktorým sa určí najoptimálnejšia kombinácia parametrov. Tieto parametre budú použité pre jeden veľký výpočet. Samotná samooptimalizácia však nie je náplňou našej diplomovej práce. V našej práci poskytneme návrh spôsobu vyhodnocovania a systém rozšírime o exportovanie výstupných dát, ktoré budú môcť byť použíté v ďalších verziách systému pre riešenie otázky samooptimalizácie.

6.2 Navrhovaný prístup ku spracovaniu parametrov

Systém bude podporovať multiparametre pre rýchlosť slnečného vetra V a difúzny koeficient K_0 . Podporovať musí vymenovanie parametrov pomocou oddeľovania čiarkami, automatický výpočet parametrov na základe dokončenia postupnosti pomocou bodiek, ale aj udaním hodnoty počiatočného parametra, konečného parametra a kroku medzi jednotlivými hodnotami.

Správnymi vstupmi teda môžu byť všetky 3 nasledujúce prípady:

- 1. Vymenovaná postupnosť: 1e22, 3e22, 5e22, 7e22, 9e22.
- 2. Postupnosť s dopočítaním: 1e22, 3e22, .., 9e22.
- Postupnosť určená počiatočnou, konečnou hodnotou a krokom: start=1e22, stop=9e22, step=2e22.

Vytvorený parser musí vstupné dáta ošetriť od neočakávaných vstupov a nesprávnych hodnôt pre zabezpečenie systému pred jeho nežiadaným zlyhaním počas behu. Všeobecným pravidlom správnych vstupných hodnôt sú čísla s jednoduchou presnosťou, ktoré môžu obsahovať aj hodnoty zapísané pomocou desatinnej čiarky, ale aj zapísané vedeckou notáciou. Po konverzii reťazca na čísla a potrebnom overení vstupu sa zadané číselné hodnoty skonvertujú na hodnoty so správnymi jednotkami, s ktorými modely pracujú. Napríklad rýchlosť slnečného vetra sa na vstupe zadáva v jednotkách km/s, avšak model pracuje v AU/s.

V existujúcom riešení systému boli akceptovanými hodnotami difúzneho koeficientu K_0 čísla v intervale od $1 * 10^{22} cm^2/s$ po $1 * 10^{23} cm^2/s$. V prípade 2D modelu podľa Peia sme však pracovali s vyššími hodnotami K_0 , preto rozšírime rozsah povolených hodnôt do intervalu od $1*10^{22} cm^2/s$ po $3*10^{23} cm^2/s$. Pre rýchlosť slnečného vetra určíme rozsah povolených hodnôt od 100 km/s do 1500 km/s.

6.3 Navrhovaný prístup ku spúšťaniu podvýpočtov

Vzhľadom na to, že proces optimalizácie vyžaduje separátne informácie o každom podvýpočte, je potrebné si tieto výpočty ukladať do samostatných adresárov. Aby sa zachovala existujúca adresárová štruktúra výsledkov pre výpočty s jednou kombináciou prepínačov, navrhneme nový adresár pre multiparametrické výpočty. V ňom budú umiestnené globálne adresáre multiparametrických výpočtov odlíšené ich názvom, ktorý bude predstavovať časovú známku začatia výpočtu. V týchto adresároch budú vnorené príslušné podadresáre s výsledkami pre konkrétne parametre, spoločný informačný súbor obsahujúci informácie o všetkých spustených výpočtoch a spoločné grafové obrázky. V podadresároch sa bude nachádzať log súbor zaznamenávajúci zachytené častice, štatistické výstupné súbory pre príslušné LIS spektrá Yamada alebo Webber-Higbie a grafové obrázky pre konkrétny podvýpočet.

Samotné spúšťanie podvýpočtov bude manažovať hlavný proces programu. Distribúcia podvýpočtov bude sekvenčná, pričom každý výpočet bude vykonávaný paralelne na grafickej karte CUDA. Po skončení výpočtu a jeho vyhodnotení sa alokované pamäte na grafických kartách uvoľnia, inicializujú sa nanovo spolu s novým náhodným stavom a hlavný proces spustí nasledujúci podvýpočet s novými parametrami. Po prejdení všetkých možných kombinácií požadovaných parametrov nasleduje optimalizačné vyhodnotenie.

6.4 Navrhovaný prístup k optimalizačnému vyhodnoteniu

Jedným z možných porovnávacích prístupov pri vyhodnocovaní optimálnych parametrov môže byť výpočet strednej kvadratickej odchýlky. Porovnávanie bolo použité aj v článku JGR [14] pri stanovovaní neistoty meraní (angl. Method uncertainty) spektier na 1AU zo stochastickej integrácie Monte Carlo metódy a Crank-Nicolson metódy. Pri porovnávaní sa vychádzalo z nasledujúcich vzorcov 6.1 pre η_i a 6.2 pre η_{RMS} :

$$\eta_{i} = \frac{J_{comp}(T_{i}) - J_{ref}(T_{i})}{J_{ref}(T_{i})}$$
(6.1)

$$\eta_{RMS} = \sqrt{\frac{\sum_{i} \left(\frac{\eta_i}{\sigma_{n,i}}\right)^2}{\sum_{i} \frac{1}{\sigma_{n,i}^2}}}$$
(6.2)

V uvedených vzorcoch T_i predstavuje i-tú hodnotu kinetickej energie, $J_{comp}(T_i)$

predstavovalo i-tú hodnotu spektra Monte Carlo metódy, $J_{ref}(T_i)$ je i-tá hodnota spektra Crank-Nicolson metódy, η_{RMS} je hodnota odchýlky a $\sigma_{n,i}$ je i-tá hodnota štandardnej odchýlky.

V našej implementácii môžeme výpočtové vzorce použiť pre určenie odchýlky referenčných spektier (J_{ref}) voči predpočítanému spektru (J_{comp}), ktoré bude zadané používateľom na vstupe. Po úspešnom skončení behu multivýpočtu si z každého log súboru vypočítame štandardnú odchýlku $\sigma_{n,i}$ pomocou intenzity w_i a strednej hodnoty i_{AVG} pre každý bin pomocou vzorca 6.3.

$$\sigma_{n,i} = \frac{(w_i - i_{AVG})^2}{N}$$
(6.3)

Uvedenú hodnotu zapíšeme aj do výstupných súborov ako štvrtý sťĺpec. Samozrejme, v prípade uchovania log súboru si je možné tieto hodnoty vypočítať dodatočne. Častokrát sa však stáva, že log súbory si kvôli ich veľkosti neuchovávame a preto je vhodné si túto informáciu zapísať do výstupného súboru, z ktorého by sme ju v prípade potreby mohli využiť.

Príkladom reálneho využitia tejto funkcie je určenie správnosti Monte Carlo metódy voči meraniami získaným dátam. Môžeme hľadať najpodobnejšiu kombináciu prepínačov v našej implementácii Monte Carlo metódy voči reálnym dátam z meracích prístrojov umiestnených na povrchu Zeme a v kozme. Napríklad Force field aproximácia spektier odvodených z meraní neutrónových monitorov [24] a meraní PAMELA experimentu [13] pokrýva obdobie od roku 1951 do roku 2016.

PAMELA experiment bol aktívny na orbite Zeme od roku 2006 do roku 2010 a poskytol sériu veľmi presných meraní spektier kozmického žiarenia. Merania prebiehali v období klesajúcej slnečnej aktivity. Spektrá protónovej zložky kozmického žiarenia kolaborácia PAMELA zverejnila s časovým krokom približne jeden mesiac, presnejšie pre jednotlivé Carringtonove rotácie rokov 2006 až 2010. Pre účely testovania modelu sme vybrali prvé zverejnené spektrum z roku 2006 a posledné zverejnené spektrum z roku 2010. Vyhodnoteniu týchto modelov sa venujeme v kapitole 9.

Výstupné súbory v databáze však pracovali s reálnym LIS. V našich implementáciách sme používali model LIS Yamada, ktorý je veľmi užitočný v počiatočných a testovacích fázach vývoja modelov.

6.5 Doplnenie vyhodnocovacích algoritmov

Nové funkcionality systému, ktoré sú používateľom voliteľné, bolo nutné zapínať a vypínať na základe prepínačov. Z hľadiska vyhodnocovacích algoritmov sme systém rozšírili z pôvodnej podpory vyhodnotenia s použitím Yamada LIS spektra o dva nové modely LIS spektra. Prvým prístupom je implementácia LIS modelu Webber-Higbie [21] pri dvojrozmernom Pei-ovom modeli, ktorý narozdiel od LIS modelu Yamada nepracuje s distribučnou funkciou f, ale s intenzitou v jednotkách počet častíc na (m^2 s sr MeV). Na vyhodnotenie používa vzorec pre modifikovaný J_{LIS} , ktorého matematické vyjadrenie je uvedené vo vzorci 5.9.

K dispozícii je viacero modelov LIS spektra [25] [12]. Pre porovnanie výstupných dát z našich algoritmov s inými dátami, napríklad s dátami z prístroja z experimentu PAMELA [13], potrebujeme aj my použiť iný druh vyhodnotenia. Z dôvodov porovnania s Force Field aproximáciou PAMELA dát si zvolíme v našom systéme LIS použité autormi aproximácie.

Spôsoby LIS vyhodnotenia sme v procese vývoja zahrnuli v našom systéme. Na základe zvoleného typu vyhodnotenia sa modifikuje aj obsah výstupných súborov. Systém je možné vylepšiť pridaním prepínačov, ktorými by sa dynamicky menili vyhodnocovacie algoritmy na základe používateľového želania. Pri behu programu bez použitia prepínača by sa automaticky použila predvolená vyhodnocovacia metóda Yamada, ktorá bola zaužívaná aj v predchádzajúcich verziách systému.

6.6 Export výstupných dát v podobe grafov

Pre porozumenie výstupných dát sa veľmi často vytvárajú rovnaké typy grafov, napríklad graf pre zobrazenie intenzity zaznamenaných častíc pre príslušnú hodnotu kinetickej energie na 1AU alebo graf zobrazujúci počet registrovaných častíc v príslušnom bine. V našom systéme sme doplnili funkcionalitu pre automatické vytváranie týchto typov grafov za pomoci externej knižnice GNUPlot. Táto grafická knižnica umožňuje jednoduché vytváranie grafov z prostredia príkazového riadku alebo pri použití externých dynamických knižníc aj v rámci programového kódu.



Obr. 6.2: Automaticky vygenerované grafy

Pri generovaní grafov sa vygenerujú dočasné súbory, v ktorých sú zapísané súradnice x a y pre krivku grafu. Tento súbor sa pošle do knižnice GNUPlot lib spolu s parametrami, ktoré modifikujú vzhľad výsledného grafu. Podobne ako pri voliteľných funkcionalitách pre zmenu vyhodnocovacieho algoritmu je možné aj tu pridať do systému prepínač, ktorým je možné generovanie grafov zapnúť z prostredia príkazového riadku. Príklad automaticky vygenerovaného grafu intenzity častíc aj pre počet registrovaných častíc na 1000 binoch je možné vidieť na obrázku 6.2.

7 Návrh softvérového riešenia pre systém kontroly výsledkov

Jedným z cieľov tejto diplomovej práce je vytvorenie samostatného nástroja, ktorý bude slúžiť na sprehľadnenie a uľahčenie kontroly výsledkov heliosférických výpočtov z dedikovaného vzdialeného počítača. V tejto kapitole sa budeme zaoberať návrhom softvérového riešenia, ktoré pozostáva z analýzy cieľových používateľov, ich požiadaviek na systém a na základe nich navrhneme štruktúru multifunkčnej aplikácie.

7.1 Analýza cieľových používateľov

Hlavnými používateľmi systému sú vedeckí pracovníci Slovenskej akadémie vied. S výhľadom do budúcna sa však plánujú rezervovať výpočtové prostriedky aj pre externých záujemcov, ktorým budú prístupné verejné kontá s vlastnými výpočtovými prostriedkami, ktoré budú oddelené od kľúčových serverov. Hovoríme teda o dvoch skupinách používateľov a viacerých serveroch. Je nutné špecifikovať a autentifikovať používateľa priamo v nástroji prostredníctvom prístupových dát, ktoré budú používatelia zadávať manuálne pri pripájaní na server.

Systém pre kontrolu výsledkov však nemusí byť nutne online nástrojom. Mnohí pracovníci si výsledné dáta sťahujú na svoje externé zariadenia a zdieľajú si ich medzi sebou. Z toho dôvodu navrhneme systém tak, aby podporoval prehliadanie aj offline dát priamo z pamäte počítača alebo externých zariadení. V oboch prípadoch bude postačujúce vytvorenie jedného prehliadača, ktorý na vstupe príjme adresárovú cestu k súborom. V prípade online prehliadania sa použije adresa dočasného priečinku, do ktorého sa potrebné súbory stiahnú zo vzdialaného servera.

7.2 Analýza súčasného pracovného toku

Spúšťanie súčasného programového riešenia, spolu s modifikovanými zmenami uvedenými v predchádzajúcich kapitolách tejto diplomovej práce, sa uskutočňuje prostredníctvom rozhrania príkazového riadku. Hrubá schéma výpočtového procesu je uvedená na obrázku 7.1. Pomocou dodatočných argumentov programu si používateľ zvolí typ heliosferického výpočtu, modifikuje vstupné parametre a zadáva želaný počet injektovaných častíc výpočtu. Program následne inicializuje výpočtové prostredie, alokuje potrebnú pamäť na grafických kartách, ale aj mimo nich, a spustí výpočet pre zadané parametre s podporou CUDA paralelizácie. Počas behu programu sa v závislosti od zvolenej metódy výpočtu priebežne zapisujú do výstupných súborov informácie o časticiach, ktoré sú použité pre vytvorenie konečných výstupných súborov a grafov. Po skončení vykonávania sa pamäte uvoľnia, vygenerujú sa grafy a program končí.



Obr. 7.1: Proces behu programu

Systém bol v kapitole 6 obohatený aj o spúšťanie multivýpočtov, kde používateľ zadefinuje viacnásobné parametre výpočtu. Pri takomto type sa vytvorí jeden hlavný priečinok, v ktorom sú hierarchicky povytvárané ďalšie podadresáre. Jednotlivé podadresáre obsahujú konkrétne výstupné súbory a obrázky grafov jedného výpočtu. Je zrejmé, že prehliadanie takýchto súborov môže byť chaotické a ich spracovanie je náročné. V hlavnom priečinku sa nachádza prehľadový súbor obsahujúci všetky zadefinované parametre, informácie o podvýpočtoch a stave, v akom sa aktuálne bežiaci výpočet nachádza. Súbor je svojim textovým obsahom čitateľný a zrozumiteľný aj pre ľudí a štruktúrovanie textových informácii je navrhnuté tak, aby bol ľahko spracovateľný pre ľubovoľné externé programy vytvorené nezávisle na tejto diplomovej práci.

Hlavná myšlienka nášho dodatočného systému pre kontrolu výpočtov je vytvorenie samostatnej prenositeľnej aplikácie. Znamená to, že proces spúšťania heliosferických výpočtov ostane neporušený a oddelený vo forme CLI programu bez grafického rozhrania umiestneného na serveri. Týmto sa zabezpečí, že nie je nutné meniť súčasný pracovný tok, rýchlosť spracovania výpočtov ostane zachovaná a nebude obmedzovaná ďalšími pamäťovými a výpočtovými nárokmi. Naše riešenie však využije všetky dostupné výstupné súbory a zjednoduší spracovanie dát používateľovi systému.

7.3 Návrh cieľovej platformy

V rámci návrhu softvérového riešenia uvažujeme medzi desktopovou a webovou aplikáciou. V dnešnom svete je veľmi rozšírený trend mať všetko vo forme webových aplikácii a stránok. Niet sa čo čudovať, keďže podľa publikácie [2] je dnes v rozvinutých krajínách pripojených k internetu viac než 90% ľudskej populácie. Veľký dosah majú sociálne siete a online hry, ku ktorým sa často pristupuje prostredníctvom webových prehliadačov bez nutnosti inštalovania dodatočných aplikácií. Ďalšou výhodou webových aplikácií je častá nenáročnosť na operačný systém a platformu. Webové prehliadače sú dostupné ako na desktopových zariadeniach, tak aj na mobilných telefónoch a tabletoch. Nevýhodou je však nutnosť pripojenia na internet a v prípade slabého signálu alebo nízkych prenosových rýchlostí je práca s webovou aplikáciou nepríjemná a problematická.

Výhodou desktopovej aplikácie je možnosť práce s offline dátami, ktoré sú uložené na externom pamäťovom úložisku alebo priamo v pamäti počítača, z ktorého používateľ pristupuje k aplikácii. V rámci cieľového operačného systému sme zvolili platformu Windows, keďže ide o jeden z najrozšírenejších a najpoužívanejších operačných systémov vo svete. Znamená to, že našu desktopovú aplikáciu budú môcť používať na vedecké účely ako pracovníci SAV, tak aj externí záujemcovia, ktorým budú prístupné verejné výpočtové prostriedky. Jednou z ďalších výhod grafického rozhrania je určite prívetivý používateľský zážitok u ľudí, ktorí s CLI rozhraním nepracujú často. Ide hlavne o bežných používateľov, ktorí sú zvyknutí na operačné systémy s podporou grafického rozhrania ako aj rôzne aplikácie, ktoré sa v dnešnej dobe nezaobídu bez GUI.

Samozrejme, práca v CLI je všestrannejšia. Ponúka možnosť vytvárania vlastných skriptov a neobmedzuje používateľa len svojimi graficky sprístupnenými funkcionalitami. Pre prácu so základnými funkcionalitami, ktoré náš nástroj bude ponúkať, je pre používateľa prirodzenejšia práca v grafickom rozhraní. Pri návrhu grafického rozhrania sa môžeme inšpirovať známymi aplikáciami, ktoré podporujú exportovanie grafov, prehliadanie súborov, tabuľkovými editormi pre zobrazovanie dát alebo sieťovými aplikáciami na vzdialenú prácu so súbormi. Všetky aplikácie spája panel nástrojov, v ktorom sa je možné za pomoci jednoduchého klikania myšou prepínať medzi podporovanými nástrojmi. Vytvoriť kvalitné UI nie je len o dizajnovaní rozhrania. Zahrnúť je potrebné aj UX princípy, ktoré používateľovi spríjemnia používanie aplikácie. Autori v [7] pri dizajnovacích princípoch hovoria o základných bodoch, ktoré by malo rozhranie spĺňať. Potrebujeme, aby obsah bol konzistentný a jednoduchý na vnímanie, teda by mal priamo zobrazovať len to, čo je potrebné. Hlavná stránka nemusí zobrazovať všetky funkcionality, ktoré aplikácia ponúka. Zároveň hovorí o výbere farieb rôznych tlačidiel a textu tak, aby ich vzhľad napovedal používateľovi o ich funkcionalite. Základné body z publikácie sme zahrnuli v našom návrhu používateľského rozhrania, ktoré je predstavené na obrázku 7.2.



Obr. 7.2: Návrh grafického rozhrania

7.4 Návrh potrebných sieťových funkcionalít

Vzhľadom na to, že nástroj pre prácu s heliosferickými výpočtami má byť skôr či neskôr dostupný nielen interným výskumníkom Slovenskej akadémie vied, ale aj širšej verejnosti, je potrebné navrhnúť a implementovať nástroj, ktorý prostredníctvom prívetivého grafického rozhrania umožní používateľovi spúšťať výpočty, pozrieť sa na zoznam aktuálne bežiacich výpočtov a zobraziť výsledky hotových výpočtov prostredníctvom grafových obrázkov zo vzdialeného serveru.

Navrhované softvérové riešenie musí podporovať online prístup na server, ktorý bude bezpečný a v závislosti od používateľovho pripojenia rýchly. Znamená to, že používateľ pre možnosť online práce s výpočtami bude musieť byť autorizovaný. V rámci používania výpočtovej sily pre spúšťanie výpočtov bude zriadené aj verejné konto, ktoré môžu využívať aj externí používatelia systému v obmedzenom režime. V súvislosti s prístupom na server bude nástroj podporovať ako čítanie hotových výpočtov zo súborového systému servera, tak aj spúšťanie nových výpočtov na základe zadaného vstupu v grafickom rozhraní nástroja. Burns sa vo svojej knihe [6] o sieťovom programovaní v programovacom jazyku C# venoval viacerým možnostiam pre vytvorenie bezpečného pripojenia. Opieral sa o knižnicu SSH.NET, v ktorej sú preddefinované rozhrania pre SSH a SFTP pripojenie. Knižnica je voľne dostupná a je ju možné využiť aj v našej aplikácii.

Ďalšou užitočnou funkcionalitou je prehľad o hotových a prebiehajúcich výpočtoch. Vďaka priebežnému zapisovaniu údajov máme počas behu výpočtu prehľad o stave, v akom sa nachádza. Náš nástroj si tieto informácie môže extrahovať a na základe informácií bude podporovať aj real-time zobrazovanie zoznamu výpočtov spolu so stavom, v ktorom sa aktuálny výpočet nachádza. To znamená, že používateľ bude mať prehľad o všetkých výpočtoch, ktoré sa na serveri vykonávali a sú hotové, ale aj o aktuálne bežiacich výpočtoch spolu s percentuálnym identifikátorom určujúcim ich úroveň dokončenia. Používateľ si bude môcť priamo v prehliadači zobraziť detaily výpočtu a vykonávať komparatívne operácie nad hotovými výpočtami.

7.5 Návrh exportovania dát do štandardizovaného formátu

Užitočnou funkcionalitou je export stavu a informácii o výpočte do štandardizovaného formátu. V navrhovanej aplikácii sa informácie budú ukladať v podobe objektov, ktorých obsah je možné serializovať do súboru. Aktuálne sú populárne dva typy štandardizovaných formátov určených pre prenos údajov. Hovoríme o formáte XML a JSON. Značkovací jazyk XML pozostáva z pomenovaných elementov, ktoré môžu mať svoje atribúty a do ktorých vieme vkladať informácie. Pri deserializácii si musíme súbor stále rozobrať (angl. parsing), čo vyžaduje dodatočnú funkcionalitu na spracovanie takéhoto súboru.

Pri formáte JSON hovoríme o objektovej notácii programovacieho jazyka JavaScript, ktorý sa používa hlavne na prenos údajov medzi webovými servermi a webovým klientom, napríklad prehliadačom. Oproti XML je lepšie čitateľný a v prípade použitia programovacieho jazyka JavaScript sa nevyžaduje dodatočná funkcionalita na spracovanie dát v tomto formáte, keďže jazyk obsahuje vlastnú funkciu na spracovanie tohto formátu. Tento typ formátu sa kontinuálne tlačí do popredia a tvorí nosný formát pre rôzne typy API rozhraní vo webovej sfére.

Z tohto hľadiska sa v našej aplikácii bude využívať JSON formát pre export výpočtových údajov. Vo verzii .NET 3.0 sa pridala funkcionalita zápisu a čítania JSON súborov do vstavanej funkcionality [3] v rámci menného priestoru v prostredí .NET s názvom "System.Text". V predošlých verziách bola veľmi využívaná externá knižnica Newtonson.JSON, ktorá je obľúbená medzi vývojármi aj v súčasnosti. Samozrejme, vývojári .NET rámca tlačia na používanie vstavaného balíčka pre čítanie a zapisovanie vo formáte JSON, v ktorom sľubujú vysoko efektívne a pamäťovo nenáročné spracovanie dát. Mnoho existujúcich riešení a systémov však stále využíva knižnicu Newtonson.JSON [10], ktorá je pre väčšinu systémov postačujúca a prechod na vstavanú knižnicu sa neplánuje. Komunita vývojárov, ktorí využívajú knižnicu Newtonson.JSON, ako aj počet existujúcich riešení a návodov pre spracovanie dát vo formáte JSON so spomínanou knižnicou je značne vyššia a preto ju budeme v našej aplikácii používať. Samotné nastavenie je veľmi jednoduché a návod je dobre opísaný vo viacerých referenčných príručkách a bibliografických zdrojoch, napríklad v knihe [10].

7.6 Návrh dodatočných funkcionalít

Bežnou praktikou používateľov, ktorí pracovali s CLI verziou heliosferických výpočtov, je vytváranie grafov. Na základe nich získa výskumník prehľad o energiách častíc na registračnej hranici (zvyčajne 1AU), počte registrovaných častíc, ale rovnako vie aj zhodnotiť, či program zbehol bez zjavných štatistických chýb. Implementáciu grafového výstupu z CUDA programu sme opísali v kapitole 6. Náš navrhovaný nástroj by mal poskytnúť jednoduchý a rýchly prístup k týmto grafom priamo zo serveru alebo v offline verzii z príslušného priečinku s výstupnými súbormi.

Ďalším bežným prístupom používateľov je hľadanie spojitosti medzi výstupnými súbormi rôznych modelov. Úroveň odchýlky medzi výstupmi môže signalizovať mieru správnosti implementovaného modelu. Náš nástroj bude podporovať vytváranie grafov a teplotných máp (angl. heatmap) priamo prostredníctvom použivateľského rozhrania. Táto funkcionalita nielenže zefektívni pracovný tok, ale v prípade nášho gridového výpočtu aj dopomôže k nájdeniu správnych parametrov a korelácii medzi nástrojom podporovanými výpočtovými modelmi a želaným výsledkom z externého modelu či reálne nameraných dát. Vytvorené grafy sa následne načítajú do používateľského rozhrania s interaktívnymi prvkami.

8 Implementácia softvérového riešenia pre systém kontroly výsledkov

Na základe navrhovaného konceptu systému pre kontrolu výsledkov sme za cieľovú platformu zvolili desktopovú aplikáciu. Pri implementácii sme použili programovací jazyk C# a využili vývojovú platformu WPF, ktorá pri dizajnovaní grafického prostredia využíva značkovací jazyk XAML. Vybraná kombinácia programovacích jazykov poskytuje možnosť vytvorenia modernej desktopovej aplikácie založenej na aplikačnom rámci .NET Framework.

Pri implementácii sme využili vývojové prostredie Visual Studio 2017, ktoré podporuje efektívne vyvíjanie webových, desktopových, ale aj mobilných aplikácií [8] písaných hlavne v programovacích jazykoch C++, C# a JavaScript. Zároveň sa tu nachádza aj vstavaný vyhľadávač NuGet balíčkov, v ktorom si môžeme rýchlo a jednoducho stiahnuť rôzne overené knižnice a komponenty a vložiť ich do vyvíjaného projektu. V našej implementácii sme využili nasledujúce knižnice:

- SSH.NET práca so zabezpečeným pripojením na server prostredníctvom SSH a SFTP protokolov.
- Json.NET práca so štandardizovaným formátom JSON. Využitie pre export a import dát v príslušnom formáte.
- ScottPlot.WPF nástroj na vytváranie responzívnych a interaktívnych grafov.

8.1 Implementácia pripojenia na server

Pre umožnenie prístupu k dátam uloženým na serveri a pre funkcionalitu spúšťania výpočtov je potrebné vytvoriť zabezpečené spojenie medzi používateľskou aplikáciou a cieľovým serverom. V súčasnej dobe je pre účely vzdialeného pripojenia k serveru po sieti najrozšírenejším sieťovým protokolom SSH, ktorý poskytuje zabezpečenú šifrovanú komunikáciu. Pre bezpečný prenos údajov a súborov medzi serverom a aplikáciou sme použili zabezpečený protokol SFTP, ktorý je z hľadiska bezpečnosti založený na podobnom princípe ako SSH. Schéma pripojenia na server je predstavená na obrázku 8.1.



Obr. 8.1: Schéma procesu pripojenia na server

V rámci našej aplikácie poskytujeme používateľovi možnosť zvoliť si cieľový server na základe IP adresy. Pri pripájaní na server je používateľ požiadaný o zadanie používateľského mena a hesla. Poskytovanie pripojenia je pokryté v knižnici SSH.NET, ktoré poskytuje komplexné rozhranie pre odosielanie príkazov na server, prácu so súborovým podsystémom a prenos súborov po sieti. Po úspešnom pripojení na server sa v aplikácii odomknú funkcionality, ktoré sú prístupné len v online režime. Pri neúspešnom pokuse o prihlásenie je používateľ o tejto skutočnosti oboznámený a online funkcionality zostanú nedostupné.

Používateľ si vie vybrať pracovný priečinok umiestnený na serveri priamo prostredníctvom rozhrania aplikácie, kde sme implementovali samostatný prehliadač súborového systému. Prehliadač získava informácie o adresárovej štruktúre v reálnom čase prostredníctvom SFTP dopytov na server a zobrazuje aktuálne dostupné podadresáre v rámci priečinku, kde sa momentálne používateľ nachádza. Na základe preddefinovanej štruktúry výpočtových priečinkov dokáže aplikácia rozoznať také adresáre, v ktorých sa nachádzajú výsledky hotových výpočtov, ale aj výstupné súbory práve bežiacich výpočtov. Používateľ je počas prehliadania prostredníctvom textovej anotácie v používateľskom rozhraní oboznamovaný o tom, či sa práve nachádza v štruktúrovo podporovanom priečinku alebo nie. V prípade zvolenia podporovaného priečinku sa automaticky stiahnu základné dáta o výpočtoch zo všetkých podadresárov a používateľovi je zobrazený zoznam výpočtov spolu s informáciami o stave a parametroch výpočtu.

Počas behu programu sa program v pravidelných časových intervaloch uisťuje, či je pripojenie na server stále aktuálne. Pokiaľ sa používateľ odhlási zo serveru alebo vyprší časový limit pripojenia, aplikácia aktualizuje svoj stav, signalizuje ho v používateľskom rozhraní a aktualizuje aktuálne dostupné funkcionality. Zároveň sú všetky online funkcionality ošetrené blokom pre zachytenie a spracovanie výnimiek pre zaistenie správnej funkcionality programu.

8.2 Implementácia prehľadu výpočtov

Ak sa používateľ za pomoci svojich prihlasovacích údajov úspešne pripojil na server a zvolil správny pracovný priečinok, systém dokáže za pomoci SFTP dopytov na server zisťovať stav o aktuálne bežiacich výpočtoch. V rámci modernizácie výpočtového programového kódu je v procese výpočtového behu zaznamenávaný progres od inicializovania cuRAND generátora až po posledný zápis údajov do výstupného súboru. V prípade viacnásobného vykonávania grid-výpočtu sa informácie ukladajú do spoločného globálneho súboru, ktorý obsahuje informácie o každom podvýpočte zvlášť. Na základe logovacieho súboru je program schopný extrahovať informácie o percentuálnom pokroku výpočtu a zobraziť príslušný ukazovateľ priebehu výpočtu priamo v používateľskom rozhraní aplikácie. Schéma procesu pre stiahnutie a zobrazenie výpočtových dát je zobrazená na obrázku 8.2.



Obr. 8.2: Schéma procesu zobrazenia prehliadača výpočtov
Prehľadový prieskumník sa skladá z troch vrstiev. Prvá vrstva je všeobecná a obsahuje informácie o všetkých parametroch výpočtu. Po kliknutí na detail výpočtu sa zobrazí druhá vrstva, ktorá predstavuje zoznam všetkých podvýpočtov vybraného záznamu. Po rozkliknutí sa zobrazí posledná, tretia vrstva, v ktorej sa nachádzajú už konkrétne údaje o výpočte spolu s miniatúrou spektra výpočtu, ktoré je zvolené.

8.3 Implementácia vizualizácie dát

Z logovacieho súboru, do ktorého sa zapisujú zachytávané častice, sa pravidelne aktualizuje počas celého priebehu výpočtu. Pri ukončovaní výpočtového cyklu sa z logovacieho súboru vygenerujú štyri výstupné súbory, ktoré predstavujú hodnoty spektier spolu s počtom zachytených častíc v rámci vopred definovaného intervalu kinetickej energie. Tieto údaje vieme dodatočne vizualizovať vo forme grafov, ktoré slúžia na grafické vyhodnotenie priebehu výpočtu. Zároveň sú tieto údaje užitočné pre dodatočné porovnanie správnosti údajov s referenčnými hodnotami z iných výpočtov. Napríklad, vyhodnotenie správnosti F-p metódy vizualizovaním pomeru hodnôt z F-p výpočtu s referenčnými hodnotami B-p metódy.

Aplikácia umožňuje používateľovi prostredníctvom súborového prehliadača vyberať súbory a načítať ich do systému. Podporovanými formátmi súborov sú súbory obsahujúce spektrá v binoch štruktúrované do dvoch alebo troch stĺpcov. V prípade dvoch stĺpcov bude prvý predstavovať hodnotu kinetickej energie v jednotkách GeV a druhý stĺpec hodnotu spektra pre danú kinetickú energiu. Takýto formát je zaužívaný v Crank-Nicholsonovej implementácii výpočtového programu. V prípade troch stĺpcov bude v prvom uvedená hodnota kinetickej energie v GeV, v druhom počet častíc zachytených v rámci konkrétneho binu a v treťom hodnota spektra pre daný bin.

Používateľ má možnosť zvoliť typ zobrazeného grafu, teda či chce dáta vizualizovať spojenou kontúrou grafu alebo zobraziť iba jednotlivé body v grafe. Zároveň si vie manuálne zvoliť spomedzi načítaných dát iba tie záznamy, ktoré chce zobraziť. Druhou funkcionalitou je vizualizácia pomeru dát. Používateľ si vyberie požadovanú dvojicu záznamov, z ktorých sa vypočíta ich pomer a výsledné dáta sa vykreslia v podobe grafu.

9 Vyhodnotenie nástroja

Všetky doplnené funkcionality systému pre spúšťanie modelov ako aj samotný externý nástroj sme otestovali v praxi pri porovnávaní force-field dát s výstupnými dátami systému pre paralelizáciu výpočtových modelov.

9.1 Spustenie výpočtu

Pre zhodnotenie presnosti naprogramovaných modelov sme zvolili výpočet B-p metódy s mapou parametrov pre rýchlosť slnečného vetra od 300km/s po 700km/s s krokom 20km/s, difúznymi koeficientami od $1.5 * 10^{22}cm^2/s$ po $1 * 10^{23}cm^2/s$ s krokom $0.5 * 10^{22}cm^2/s$, časovým krokom 50s a počtom injektovaných častíc 160 miliónov. Vo výsledku by sme tak mali získať celkový počet 372 výpočtov. Pre spustenie sme využili grafické rozhranie nášho externého nástroja. V nástroji sme sa pripojili na vzdialený počítač prostredníctvom nášho používateľského mena, hesla a IP adresy počítača. Po pripojení sme zvolili pracovný priečinok, v ktorom sa nachádza aktuálna verzia modelov. Po úspešnom načítaní dát sa nám odomkla funkcionalita pre spúšťanie výpočtov v paneli nástrojov. Grafické rozhranie spúšťania výpočtu je možné vidieť na obrázku 9.1.

V okne sme vyplnili požadované parametre výpočtu a odoslali požiadavku na server. Keďže sa vytvoril nový proces pre spúšťanie výpočtov, naše grafické rozhranie nástroja ostalo funkčné bez zamrznutia a nástroj môžeme využívať aj naďalej. V pozadí sa však odoslala požiadavka na vzdialený počítač, v ktorom sa začali vykonávať výpočty. V našom nástroji môžeme sledovať percentuálny progres výpočtu, v akom sa práve nachádza. Ako môžeme na obrázku 9.2 vidieť, v momente vytvorenia snímky obrazovky sa aktuálne vykonával podvýpočet s parametrami $K0 = 2 * 10^{22} cm^2/s$ a V = 300 km/s, pričom je hotový na 60%.



Obr. 9.1: Okno nástroja pre spustenie výpočtu

Status:		Working Dir: Change									
Connected		//home/nguyen/cuda-helio-cli-DP-final									
(Calculation detail										X
? List executions	Status	ко	dt	v	N	Percentage	Error	MethodType	OnlineUrl	LocalUrl	ImageUrl
	100%	1.5E+22	50	300	10	100	NaN	BP_1D	//home/nguyen/c	C:\Users\marti\so	C:\Users\marti\sc
	60%	2E+22	50	300	10	60	NaN	BP_1D	//home/nguyen/c	C:\Users\marti\so	C:\Users\marti\sc

Obr. 9.2: Prehľad aktuálne vykonávaného výpočtu

9.2 Zobrazenie výsledkov a vyhodnotenie chyby

Celkové trvanie všetkých 372 podvýpočtov trvalo 3 dni a 10 hodín. Výpočet nebol prerušený a bol správne ukončený. Logovací súbor obsahoval všetky potrebné informácie, ktoré boli v rámci prehliadača načítané v našom nástroji. V rámci prehľadu máme možnosť vypočítať percentuálnu odchýlku od očakávaného vstupu, ktorým dostaneme prehľad o množine kombinácii parametrov najpodobnejších a najmenej podobných výpočtov. Náhľad prehľadového okna spolu s vypočítanými hodnotami odchýlky je možné vidieť na obrázku 9.3.

Pre overenie sme použili dáta, spektrum protónov, z experimentu PAMELA, z obdobia od 07.07.2006 do 26.07.2006, čo je prvé zverejnené spektrum experimentu. To je označované ako S1 a v článku [13]. Dáta sú aproximované Force field spektrom s modulačným potenciálom 559MV. Z meraní očakávame, že najmenšia chyba, teda najmenší rozdiel medzi meraním a jedným zo spektier z B-p modelu,



Obr. 9.3: Hotový výpočet s určením odchýlky

bude pri kombinácii parametrov $K0 = 3.6 * 10^{22} cm^2/s$ a V = 429 km/s. Po načítaní spektra si v nástroji vieme upraviť poradie stĺpcov súboru a zvoliť ich typ. Po kontrole sa dopočítajú percentuálne RMS chyby, ktorých hodnoty sú zobrazené priamo v prehliadači. Ako chybu v tomto prípade označujeme RMS chybu η_{RMS} , ktorá udáva rozdiel medzi spektrom z B-p metódy a importovaným spektrom, ktoré zvyčajne pochádza z meraní. RMS chybu určujeme podľa nasledujúceho vzťahu 9.1.

$$\eta_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \eta_i^2} \tag{9.1}$$

Vo vzťahu platí pre η_i nasledujúci vzťah 9.2.

$$\eta_i = \frac{J_{B-p}(T_i) - J_{experiment}(T_i)}{J_{experiment}(T_i)}$$
(9.2)

kde T_i je kinetická energia binu, $J_{B-p}(T_i)$ je intenzita pre T_i získaná simuláciou v B-p modeli a $J_{experiment}(T_i)$ je intenzita z experimentu. Keďže modulácia kozmického žiarenia je najväčšia na nízkych energiách, tak RMS chybu počítame v rozsahu energií 0,1GeV až 2GeV. Na základe hodnoty si ich vieme usporiadať vzostupne ale aj zostupne. Okrem toho máme implementovanú funkcionalitu tepelnej mapy, ktorá nám vizuálne odlíši mieru odchýlky pomocou farby. Výstup je predstavený na obrázku 9.4.

Ako môžeme z obrázku 9.4 vidieť, najmenšie odchýlky predstavujú modrú





Obr. 9.4: Tepelná mapa pre odchýlku od databázy 559

farbu a najbližšie výpočty sú uvedené v tabuľke 9.1. Naša očakávaná kombinácia prepínačov spĺňa kritériá pre parametre $K0 = 3,5*10^{22} cm^2/s$ a V = 400 km/s, čím sme overili, že naša implementácia B-p metódy na grafickej karte je v porovnaní s experimentálnymi údajmi korektná.

K0	V	Odchýlka		
3e22	360	0,81%		
5e22	600	0,82%		
4e22	480	0,83%		
3,5e22	420	0,84%		
2,5e22	300	0,84%		

Tabuľka 9.1: Tabuľka parametrov s najmenšou odchýlkou od Force Field spektra s potenciálom 559MV

Pre kontrolu si overíme správnosť aj s PAMELA dátami z obdobia od 02.01.2010 do 23.01.2010, so spektrom označeným ako S47, čo bolo spektrum získlané počas posledného Carringtonovho cyklu počas ktorého experiment fungoval. Toto spektrum je aproximované Force field spektrom s modulačným potenciálom 372MV. Najmenšiu odchýlku by sme mali dostať s parametrami $K0 = 5,3 * 10^{22} cm^2/s$ a V = 405 km/s. Po načítaní dát do nástroja a výpočte percentuálnej chyby si v nástroji môžeme zostaviť tepelnú mapu, ktorá je zobrazená na obrázku 9.5. Z vytvorenej mapy vidíme, že najbližšie parametre $K0 = 5 * 10^{22} cm^2/s$ a V = 400 km/ssa nachádzajú v tmavomodrej farbe a z tabuľky 9.2 môžeme vyčítať, že odchýlka je iba 0,7%. Môžeme teda aj v tomto prípade skonštatovať, že B-p model je implementovaný správne.

K0	V	Odchýlka	
9e22	700	0,63%	
8,5e22	660	0,66%	
5,5e22	440	0,67%	
8e22	620	0,67%	
8,5e22	680	0,68%	
5e22	400	0,70%	

Tabuľka 9.2: Tabuľka parametrov s najmenšou odchýlkou od Force Field spektra s potenciálom 372MV



372

Obr. 9.5: Tepelná mapa pre odchýlku od databázy 372

Uvedené výsledky ukazujú dôležitú vlastnosť nielen jednorozmerných riešení

B-p a F-p metódy. Mapa ukazuje, že je viacero spektier z B-p metódy podobných spektru z meraní. Pre viaceré kombinácie parametrov rýchlosti slnečného vetra a difúzneho koeficientu získame veľmi podobné spektrum. Všetky spektrá z modrej oblasti sú podobné meraniu. Keďže z meraní rýchlosti slnečného vetra vieme, aká bola jeho hodnota, vieme vybrať z modrej oblasti hodnotu K0, ktorá prislúcha danému meraniu. To okrem iného fyzikálne hovorí, že ak sa spektrum z B-p či F-p modelu zhoduje v rámci malej chyby (napríklad menej než 1 percentný rozdiel intenzít medzi 0,1GeV až 2GeV) s meraním, tak model nutne neopisuje parametre prítomné počas merania. Z hľadiska jednopercentnej chyby, rovnaké spektrum získame použitím viacerých kombinácií vstupných parametrov V a K_0 . Toto je dôležitý poznatok pre dvojrozmerné modely s viacerými vstupnými parametrami, ktoré nie sú odvodené priamo z meraní. Pri dvojrozmerných modeloch je preto zložitejšie len na základe zhody simulovaných spektier so spektrami experimentálnymi určiť, či sme vstupné parametre určili správne.

10 Záver

Predložená diplomová práca sa venuje reinžinieringu existujúcich modelov a implementácii nových výpočtových modelov zameraných na určenie modulácie kozmického žiarenia v heliosfére. Vývoj modelov bol cielený na platformu grafických kariet Nvidia s technológiou CUDA. Práca sa venovala aj ďalšiemu vývoju podporných nástrojov, ktoré svojimi funkcionalitami umožňujú používateľovi automatizovať beh výpočtových modelov, analyzovať a vizualizovať výsledky, ktoré modely produkujú.

V rámci práce boli analyzované vybrané publikácie, ktoré sa venovali paralelizácii modelov a simulácii z pôvodného programového kódu určeného na CPU vykonávanie do prostredia CUDA. Projekty pochádzali z oblasti medicíny a astrofyziky, pričom poukazovali na výhody a nevýhody použitia grafických kariet pre konkrétne výpočtové problémy. Vo všetkých prípadoch bola porovnávaná rýchlosť spracovania výpočtov medzi CPU a GPU. Bolo preukázané, že po prepise kódu na platformu CUDA a spracovaní výpočtov na GPU sa výpočty zrýchlili.

V práci bolo dokázané, že distribúcia číslic pri generovaní náhodných čísel na grafickej karte je uniformná. Tvrdenie sme dokazovali generovaním čísel hracej kocky, pričom distribúcia číslic bola približne rovná matematickej pravdepodobnosti hodu konkrétneho čísla na hracej kocke. V práci bola analyzovaná uniformnosť generovania postupnosti náhodných čísel na GPU pri simulovaní rôznych stratégií v hazardnej hre ruleta. Úlohou bola potvrdená správnosť prístupu pri generovaní náhodných čísel prostredníctvom nástrojov platformy CUDA a v rámci práce boli vyhodnotené úspešnosti jednotlivých vybraných stratégií v hre ruleta.

V predloženej práci boli analyzované dve známe chyby 1D výpočtových modelov, ktoré sú zamerané na určenie modulácie kozmického žiarenia v heliosfére. Modely sú implementované v systéme pre paralelné vykonávanie výpočtových modelov na GPU. Prvou analyzovanou chybou bola pulzácia, ktorá sa objavuje pri malých časových krokoch. V práci sme potvrdili dôvod pulzácie v grafe pomeru F-p a B-p dát pri malých časových krokoch výpočtu pomocou zmeny typu premenných z jednoduchej presnosti na dvojitú. Na základe simulačných experimentov bola vyvrátená hypotéza o vplyve optimalizačného prepínača na vplyv pulzácie. Následne sme sa v práci venovali druhej štatistickej chybe, ktorá je viditeľná v grafe pomeru F-p a B-p modelov pri vyšších časových krokoch výpočtu. Predpokladaný vplyv počiatočnej vzdialenosti častice od hranice heliosféry na výskyt tejto chyby sa v troch iteráciách pokusu nepotvrdil. Pokus prebiehal pomocou zmeny počiatočnej hodnoty vzdialenosti injektovanej častice a sledovaním zmeny v krivke grafu.

Existujúce paralelné modely, ktoré boli vstupom tejto práce, boli rozšírené o nový 2D model podľa Pei-ovho referenčného článku. Zámerom implementácie nového modelu bolo overiť správnosť už implementovaných modelov. Existujúci 2D model má oproti referenčnému modelu viacero odlišností, napríklad v tvare závislosti rýchlosti slnečného vetra na heliošírke. Pri snahe o prepis referenčného modelu do nášho systému sme sa v niekoľkých iteráciách pomocou ladiacich experimentov snažili vstupné parametre a nastavenia implementovaného modelu čo najviac priblížiť Pei-ovmu modelu, keďže informácie o viacerých vstupných parametroch autori v publikácií neuvádzajú. V Pei-ovej publikácií je uvedená mimoriadne veľká hodnota difúzneho koeficientu, použitím ktorého sa výsledky modelov implementovaných v práci nepribližovali k spektru v referenčnom článku. Táto hodnota mohla byť v článku uvedená chybne, keďže s jej znižovaním výsledky referenčného modelu konvergovali k výsledkom implementovanému modelu.

Funkcionalita systému bola rozšírená o možnosť zadefinovania matice parametrov, ktorou sa pomocou jedného príkazu v prostredí príkazového riadku spustí multivýpočet. So spomínanou funkciou je systém pripravený pre budúce samoadaptačné a samooptimalizačné vylepšenia. Na základe zadaných parametrov sa v systéme začnú vykonávať výpočty s jednotlivými kombináciami viacerých parametrov. Používateľ môže vygenerované výstupné dáta ďalej analyzovať a vizualizovať v podobe tepelných máp a grafov.

V práci bol navrhnutý a implementovaný nástroj pre zefektívnenie práce vý-

skumníkov pri analyzovaní a vizualizovaní multivýpočtových výsledkov z 1D modelov. Vytvorený nástroj ponúka v grafickom rozhraní možnosť pripojenia na vzdialený server, spúšťanie multivýpočtov, prezeranie výsledných dát a prácu s nimi. Užitočnosť systému sa preukázala pri overovaní výpočtových modelov určovaním RMS chyby voči aproximovaným Force field spektrám, ktoré boli získané z meraní neutrónových monitorov a meraní PAMELA experimentu. Nástroj umožnil spustiť kontrolný multivýpočet, načítať dáta, určením RMS chýb nájsť najpodobnejšie spektrá medzi množinou simulovaných spektier a importovaným spektrom z experimentu. Určené chyby boli spracované nástrojom a bola z nich vygenerovaná tepelná mapa, ktorá je uvedená na obrázku 10.1.



559

Obr. 10.1: Potvrdenie správnosti modelov pri určení chyby na tepelnej mape z nástroja

Vzniknutý nástroj a doplnené modely budú v budúcnosti rozšírené o samooptimalizačný adaptívny algoritmus, ktorý bude použitý pri analýze dát na ústave experimentálnej fyziky SAV z experimentov na meranie kozmického žiarenia, napríklad dáta z experimentu AMS-02.

Literatúra

- [1] World Casino Directory. *Countries worldwide ranked by number of casinos*. URL: https://www.worldcasinodirectory.com/countries (cit. 17. 10. 2020).
- [2] International Telecommunication Union (ITU). *Measuring digital development Facts and figures 2020.* 2020.
- [3] Joseph Albahari a Eric Johannsen. *C# 8.0 in a nutshell: the definitive reference*. O'Reilly Media, 2020.
- [4] Claudio Schepke a Marcelo Cogo Miletto. "Acceleration of Radiofrequency Ablation Process for Liver Cancer Using GPU". In: 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP) (2020). DOI: 10.1109/pdp50117.2020.00065.
- [5] Michal Solanik. "Paralelizácia dvojrozmerných modelov modulácie kozmického žiarenia v heliosfére". 2020.
- [6] Sean Burns. *Hands-on network programming with C# and .Net Core: build robust network applications with C# and .Net Core.* Packt Publishing, 2019.
- [7] Westley Knight. UX for developers: how to integrate user-centered design principles into your day-to-day development work. Apress, 2019.
- [8] Bruce Johnson. *Professional visual studio*, 2017. John Wiley a Sons, Inc., 2018.
- [9] Marcelo Miletto a Claudio Schepke. "Acceleration of a Computational Simulation Application for Radiofrequency Ablation Procedure Using GPU". In: 2018 Symposium on High Performance Computing Systems (WSCAD) (2018). DOI: 10.1109/wscad.2018.00054.
- [10] Mark J. Price a Khan Ovais Mehboob Ahmed. C# 7 and .NET: designing modern cross-platform applications: the open source revolution of .NET Core. Packt Publishing, 2018.

- [11] Michal Solanik. "Prepis modelov distribúcie kozmického žiarenia v heliosfére do CUDA jazyka". 2018.
- [12] Carmelo Evoli et al. "Cosmic-ray propagation with DRAGON2: I. numerical solver and astrophysical ingredients". In: *Journal of Cosmology and Astroparticle Physics* 2017.02 (2017), s. 015–015. DOI: 10.1088/1475-7516/2017/ 02/015.
- [13] Ilya G. Usoskin et al. "Heliospheric modulation of cosmic rays during the neutron monitor era: Calibration using PAMELA data for 2006-2010". In: *Journal of Geophysical Research: Space Physics* 122.4 (2017), s. 3875–3887. DOI: 10.1002/2016ja023819.
- P. Bobik et al. "On the forward-backward-in-time approach for Monte Carlo solution of Parkers transport equation: One-dimensional case". In: *Journal of Geophysical Research: Space Physics* 121.5 (2016), s. 3920–3930. DOI: 10.1002/2015ja022237.
- [15] Rolf Kappl. "SOLARPROP: Charge-sign dependent solar modulation for everyone". In: *Computer Physics Communications* 207 (2016), s. 386–399. DOI: 10.1016/j.cpc.2016.05.025.
- [16] R.C. Tautz. "A graphics-card implementation of Monte-Carlo simulations for cosmic-ray transport". In: New Astronomy 45 (2016), s. 1–6. DOI: 10.1016/ j.newast.2015.10.012.
- [17] P. Dunzlaff, R.D. Strauss a M.S. Potgieter. "Solving Parker's transport equation with stochastic differential equations on GPUs". In: *Computer Physics Communications* 192 (2015), s. 156–165. DOI: 10.1016/j.cpc.2015.03.008.
- [18] Li Zhen et al. "Accelerating Large Scale Artificial Society Simulation with CPU/GPU Based Heterogeneous Parallel Method". In: 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT) (2015). DOI: 10.1109/ds-rt.2015.11.
- [19] R.D. Strauss, M.S. Potgieter a S.E.S. Ferreira. "Modelling and observing Jovian electron propagation times in the inner heliosphere". In: *Advances in Space Research* 51.3 (2013), s. 339–349. DOI: 10.1016/j.asr.2012.09.035.

- [20] P. Yu. Izotov et al. "CUDA-enabled implementation of a neural network algorithm for handwritten digit recognition". In: Optical Memory and Neural Networks 20.2 (2011), s. 98–106. DOI: 10.3103/s1060992x11020032.
- [21] C. Pei et al. "A general time-dependent stochastic method for solving Parkers transport equation in spherical coordinates". In: *Journal of Geophysical Research: Space Physics* 115.A12 (2010). DOI: 10.1029/2010ja015721.
- [22] Daniel Strigl, Klaus Kofler a Stefan Podlipnig. "Performance and Scalability of GPU-Based Convolutional Neural Networks". In: 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing (2010). DOI: 10.1109/pdp.2010.43.
- [23] Hongtao Xie et al. "GPU-based fast scale invariant interest point detector". In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (2010). DOI: 10.1109/icassp.2010.5494898.
- [24] Ilya G. Usoskin et al. "Heliospheric modulation of cosmic rays: Monthly reconstruction for 1951–2004". In: *Journal of Geophysical Research* 110.A12 (2005). DOI: 10.1029/2005ja011250.
- [25] Andrew W. Strong a Igor V. Moskalenko. "Propagation of Cosmic-Ray Nucleons in the Galaxy". In: *The Astrophysical Journal* 509.1 (1998), s. 212–228.
 DOI: 10.1086/306470.
- [26] Pawlicki et al. "Neural network models and their application to handwritten digit recognition". In: *IEEE International Conference on Neural Networks* (1988). DOI: 10.1109/icnn.1988.23913.
- [27] J. R. Jokipii, E. H. Levy a W. B. Hubbard. "Effects of particle drift on cosmicray transport. I - General properties, application to solar modulation". In: *The Astrophysical Journal* 213 (1977), s. 861. DOI: 10.1086/155218.

Zoznam skratiek

AU Astronomical unit.

- **B-p** Back propagation model with momentum.
- CLI Command Line Interface.
- **CPU** Central Processing Unit.
- CUDA Compute Unified Device Architecture.
- cuRAND NVIDIA CUDA Random Number Generation library.
- **F-p** Forward propagation model with momentum.
- F-T Forward propagation model with kinetic energy.
- GCC GNU Compiler Collection.
- GPU Graphics Processing Unit.
- **GUI** Graphical User Interface.
- IP Internet Protocol.
- JGR Journal of Geophysical Research.
- JSON JavaScript Object Notation.
- LIS Local interstellar spectra.
- NVCC Nvidia CUDA Compiler.

- **SFTP** SSH File Transfer Protocol.
- SSH Secure Shell Protocol.
- **XAML** Extensible Application Markup Language.
- XML Extensible Markup Language.

Zoznam príloh

- Príloha A Používateľská príručka simulačného programu
- Príloha B Používateľská príručka nástroja pre prehliadanie výpočtov
- Príloha C Systémová príručka nástroja pre prehliadanie výpočtov
- **Príloha D** CD médium záverečná práca a prílohy v elektronickej podobe, zdrojový kód softvérového riešenia