Dobrý deň prajem,

Volám sa Matej Magát, názov mojej bakalárskej práce bol: Newtonovská gravitácia s retardáciou

V dnešnej dobe, keď narastá potreba overovať pravdivosť informácií sa stále objavujú nové fyzikálne teórie, ktoré sa snažia vysvetliť niektoré prírodné javy iným spôsobom ako všeobecne prijímané doterajšie teórie. Niektoré z týchto teórii sa zdajú pravdivé viac iné menej, no stále je potrebné ich potvrdiť alebo vyvrátiť, preto táto bakalárska práca si dala za cieľ simuláciou ukázať, že nie je možné do newtonovských rovníc pripočítať oneskorenie teda retardáciu pôsobenia gravitácie v dôsledku  konečnej rýchlosti šírenia gravitácie bez Einsteinových rovníc.

Rozhodol som sa, že bakalársku prácu naimplementujem v Golangu, je to jazyk, s ktorým som sa stretol behom štúdia na fakulte v kurze Programovacie Paradigmy, ktorý vedie RNDr. Peter Borovanský, PhD., je to jazyk, ktorý je o niečo málo pomalší ako C/C++. Oproti iným jazykom má veľmi dobre vyriešený paralerizmus a tiež konkurenciu. Hlavne paralerizmus som plánoval využiť na zrýchlovanie výpočtov. Golang ponúka veľa zabudovaných knižníc, ale má aj veľmi efektívny nástroj na získavanie knižníc tretích strán, celkovo som do programov potreboval dve knižnice tretích strán: Grafickú knižnicu Gonutz/prototyp, ktorú som si vybral pre jej jednoduchosť  a 360EntSecGroup-Skylar/excelize, ktorá slúži na zapisovanie údajov do excelovských súborov.

Na ceste k tomuto cieľu  vznikli 3 samostatné programy, ktoré postupne riešili tieto problémy: problém jedného telesa, problém dvoch telies bez retardácie, problém dvoch telies s retardáciou, všetky programy sa otvárajú dvojklikom. Ku každému programu existuje návod, ktorý je súčasťou príloh k bakalárskej práci alebo si ho môžete pozrieť na githube. Prvým cielom bolo osvojiť si a vyskúšať si tri výpočtové metódy: Euler, Euler-Cromer a Leapfrog, pričom bol predpoklad, že Leapfrog bude s presnosťou stačiť na všetky fyzikálne problémy ak bude čas a záujem vysvetlím ich princípy na konci. Už od začiatku bolo jasné, že pri každom probléme bude dôležité aby výpočet netrval príliš dĺho a aj aby bol dostatočne presný, to potom viedlo k výpočtu dráhy analytickým spôsobom.

Pri probléme jedného telesa som spravil jednoduché grafické rozhranie, ktoré je ovládané klávesnicou, podrobný popis tu nestíham urobiť. Program som nepísal veľmi štruktúrovaný. Všetok kód sa nachádza iba v jednom petkydži. Pri tomto probléme bolo potrebné porovnávať presné polohy a rýchlosti simulovaného telesa, počas behu s očakávanými  hodnotami, preto bolo potrebné, aby program zapisoval výsledky výpočtov do excelovských súborov a nato mi slúžila knižnica excelize, ktorá pomerne rýchlo narábala s údajmi, ktoré bolo treba zapísať. V tomto probléme bolo dôležité overiť druhý Keplerov zákon, ktorý znamená, že plocha, ktorú opíše sprievodič to jest spojnica medzi telesom a hmotným bodom v strede opíše rovnakú plochu za rovnaký časový úsek. Plochy opísané sprievodičom sa dajú klávesnicou farebne označiť. Farby sú automatické a striedajú sa bez možnosti nastavenia.  Pri tomto probléme vznikol nápad konfiguračného súboru, ktorým sa veľmi ľahko dajú zmeniť počiatočné podmienky. Pri tomto programe sa počítajú všetky tri výpočtové metódy naraz v samostatných gorutinách a synchronizujú sa iba pri výstupe. Na týchto obrázkoch vidíme zábery zo samotného programu. V ľavom hornom rohu je možné vidieť úvodný pohľad programu, kde si ešte môžeme upraviť niektoré premenné skôr ako začne hlavný program. V pravom hornom obrázku si treba všimnúť, že planéta neurobila ani jeden obeh okolo centrálneho hmotného bodu a už nabrala únikovú rýchlosť, je to spôsobené veľmi silným gravitačným poľom okolo hmotného bodu. Tento obrázok ukazuje metódu Euler, ktorá nie je veľmi presná oproti tomu Euler- Cromer urobil jeden obeh okolo hmotného bodu než teleso definitívne nabralo únikovú rýchlosť, čo môžeme vidieť v ľavom dolnom obrázku. Na poslednom obrázku, v pravom dolnom rohu, je vidieť metóda Leapfrog. Ktorá je dostatočne presná na to, aby aj v silnom gravitačnom poli nedochádzalo k takej veľkej nepresnosti ako v predchádzajúcich dvoch prípadoch.

Pri klasickom probléme dvoch telies som kompletne prerobil grafické prostredie, takmer všetko sa ovláda myšou. Grafická knižnica síce neobsahovala žiadne tlačidlá, textové polia ani ček-boxy, ale pre užívateľa sú tieto prvky veľkým zľahčením ovládania. A tak som ich v tomto programe urobil vo vlastnej réžii. Pri tomto programe som urobil možnosť znovunačítať konfiguračný súbor pomocou tlačidla begin. Ukladanie do excelovských súborov nebolo dostatočne rýchle, a tak som musel vypočítané dáta ukladať do binárnych súborov, odkiaľ sa rýchlo načítavali do grafickej časti programu. Tu som využil mechanizmus označovaný ako ansejft, klasický mechanizmus konverzie poľa desatinných čísel v jazyku má časovú zložitosť O(n), čo bolo pre môj program príliš pomalé, a tak som siahol po konverzii pointera. Konverzia pointera je však mimo mechanizmus garbage collectora, a preto som musel vytvoriť štruktúru, v ktorej som si naďalej pamätal aj starý pointer na pôvodné pole. Táto konverzia pointera funguje v tomto programe obidvoma smermi, čím som výrazne zrýchlil ukladanie a načítanie dát a tým aj samotný beh programu. Napriek tomu zrýchlenie nie je dostatočné na to, aby sa program využíval v reálnom čase. Preto som do tohto programu zakomponoval aj možnosť znovunačítania, program načítava údaje z binárnych súborov a uložených džejsonov. Kvôli overeniu presnosti výpočtov som do programu zakomponoval aj analytické riešenie, ktoré je nezávislé od numerického riešenia. Toto riešenie nájde elipsu podľa energie a polohy telesa , nato slúžia matematické vzorce, kde si postupne vyrátam hlavnú poloos, excentricitu atď.

Na tomto slajde vidíme vyššie spomenutý formulár, kde na rozdiel od prvého programu sa dajú upravovať všetky parametre súvisiace so simuláciou. V dolnej časti obrázku je možné vidieť štyri tlačidlá, ktoré slúžia na generovanie náhodných dráh podľa svojho názvu. Po stlačení jedného z týchto tlačidiel sa vygeneruje dráha a údaje sa zobrazia vo formulári. Po stlačení tlačidla Dan sa začne simulácia.

 Tu vidíme výsledok simulácie 100 obehov. Na tomto obrázku si treba všimnúť zaškrtnutý ček-box. Zobrazenie dráh sa dáva do vzťažnej sústavy.

Tu vidíme rovnaký výpočet bez vzťažnej sústavy.

Na tomto obrázku je možné vidieť zmenené farby dráh, čo je preto , lebo analytické riešenie úplne prekrýva numerické riešenie, a teda môžeme prehlásiť, že numerické riešenie pri kroku, ktorý vidíme v pravom hornom rohu obrázku je dostatočne presné aj po 100 obehoch.

Tu je príklad situácie, kedy sa telesá dostávajú tak blízko k sebe, že aj leapfrog urobí takú veľkú numerickú chybu, že telesá získajú únikovú rýchlosť. Preto je potrebné mať analytické riešenie k dispozícii.

Tretí program je rozšírením druhého, pričom pridáva do výpočtu oneskorenie gravitačnej interakcie do newtonovských vzorcov. Ako budeme vidieť v obrázkoch na nasledujúcich slajdoch.  Keplerove a aj iné zákony prestávajú platiť.

Na tomto obrázku je vidieť simulácia, ktorá bola na začiatku uvedená do vzťažnej sústavy. V tejto simulácii sa ťažisko sústavy dalo samovoľne do pohybu, čo je v rozpore so zákonom zachovania hybnosti.

Tento obrázok ukazuje rovnakú situáciu ako dva predchádzajúce s tým rozdielom, že sa dynamicky vypočítava ťažisko a od toho sa vypočítava poloha každého bodu samostatne.

Medzi hlavné prínosy tejto bakalárskej práce sa považujú ukázanie nesprávnosti tohto fyzikálneho modelu, vysoká prenositeľnosť medzi jednotlivými platformami, tento program je stavaný na spustenie na bežných stolných počítačoch, kde máme pomerne dobrý pomer medzi presnosťou výpočtov a časom potrebným na realizáciu výpočtov.

Ďakujem za pozornosť.