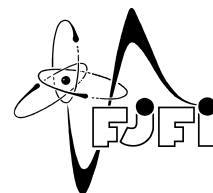




České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská
Katedra Fyzikální Elektroniky



Numerické aspekty particle-in-cell simulací

Bakalářská práce

Autor: **Viktor Kocur**
Vedoucí práce: **Ing. Jan Pšikal, Ph.D.**
Akademický rok: 2014/2015

- Zadání práce -

- Zadání práce (zadní strana) -

Pod'akovanie:

Chcel by som sa pod'akovať svojmu školiteľovi Ing. Janovi Pšikalovi, Ph.D. za starostlivosť, ochotu, ústretovosť a mnohé cenné rady pri vedení mojej bakalárskej práce.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 23. 6. 2015

Viktor Kocur

Název práce:

Numerické aspekty particle-in-cell simulací

Autor: Viktor Kocur

Obor: Informatická fyzika

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Jan Pšikal, Ph.D.

Katedra fyzikální elektroniky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

Abstrakt: Tato práce se zabývá numerickými aspekty moderních particle-in-cell simulací ve fyzice laserového plazmatu. Algoritmus PIC metody je v této práci popsán spolu s teoretickým odvozením jeho numerických vlastností. S použitím moderního PIC kódu EPOCH jsou navrženy 1D i 2D simulace pro ověření teoretických předpovědí stability a numerického ohřevu. Tyto simulace jsou následně realizovány s využitím služeb Metacentra VO. Výsledky simulací jsou vzhledem k teorii diskutovány spolu s návrhem optimálních podmínek stability a předpovědí míry numerického ohřevu pro budoucí simulace.

Klíčová slova: kód EPOCH, laserové plazma, numerický ohřev, particle-in-cell

Title:

Numerical aspects of particle-in-cell simulations

Author: Viktor Kocur

Abstract: This thesis deals with numerical aspects of modern particle-in-cell simulations in laser plasma physics. The algorithm of PIC is described along with theoretical derivation of its numerical aspects. Simulations using a modern PIC code EPOCH in both 1D and 2D to test the theoretical predictions for stability and numerical heating are proposed and then executed using the services of Metacentrum VO, provider of grid computation resources. The results of the simulations are compared with the theory. The differences and similarities are then discussed along with recommendations for optimal stability conditions and reasonable numerical heating expectations in future simulations.

Key words: EPOCH code, laser plasma, numerical heating, particle-in-cell

Obsah

Úvod	8
1 Metóda Particle In Cell	10
1.1 Základne pojmy	10
1.2 Kinetický popis plazmy	12
1.3 Základný princíp PIC	12
1.4 Tvarové funkcie	13
1.5 Výpočtový cyklus	14
1.5.1 Inicializácia	15
1.5.2 Prvý krok - interpolácia elektrických a magnetických polí na častice . . .	15
1.5.3 Druhý krok - posun častíc	16
1.5.4 Tretí krok - výpočet prúdových hustôt	17
1.5.5 Štvrtý krok - výpočet elektrického a magnetického poľa	18
1.6 Stabilita a Numerický ohrev	18
1.6.1 Parametre	18
1.6.2 CFL podmienka	19
1.6.3 Podmienka plazmovej frekvencie	19
1.6.4 Dôsledky diskkrétnej mriežky	21
1.6.5 Numerický ohrev	22
2 Implementácia	24
2.1 EPOCH	24
2.1.1 input.deck	25
2.1.2 Úprava diagnostiky	27
2.2 Spracovanie výstupu	27
2.3 Metacentrum VO	28
2.3.1 Spúšťanie úloh	29
3 Simulácie a výsledky	31
3.1 1D	31
3.1.1 Parametre simulácie	31
3.1.2 Teoretický predpoklad stability	33
3.1.3 Nestabilný ohrev	33
3.1.4 Stabilný ohrev	36
3.1.5 Hybnosti častíc	38
3.2 2D	38
3.2.1 Testovacie parametre	38

3.2.2	Teoretický predpoklad stability	39
3.2.3	Nestabilný ohrev	40
3.2.4	Stabilný ohrev	40
3.2.5	Náročnosť výpočtu	41
3.3	Diskusia	42
	Záver	44

Úvod

Moderný výskum vo fyzike laserovej plazmy napreduje a stáva sa čím ďalej tým viac atraktívnejším. Moderné vysokofrekvenčné lasery umožňujú po interakcii s terčmi skúmať takto generovanú plazmu. Počet publikácií venujúcich sa práve tejto oblasti fyziky s každým rokom rastie a neustále vznikajú nové projekty, ktoré umožňujú jej rozvoj. Ako príklad je vhodné spomenúť unikátny projekt ELI Beamlines budovaný v blízkosti Prahy.

V súčasnom výskume fyziky laserovej plazmy sa často využívajú práve počítačové simulácie. Simulácie sú alternatívou experimentu. Majú oproti nim viaceré výhody a samozrejme aj nevýhody. K výhodám patrí hlavne nižšia finančná náročnosť. Počítačové simulácie sú navyše s postupom času, čím ďalej, tým lacnejšie a skraca sa aj čas potrebný na náročné simulácie. Ďalšou výhodou simulácií je aj možnosť pozorovať väčšie množstvo veličín a získať tak detailnejšie informácie ako z experimentu. Aj napriek výhodám sa simulácia spolieha na model, ktorý, minimálne z dôvodu diskretizácie, nebude presný. Môžu sa tak v simulácii vyskytnúť nežiadané nefyzikálne javy, ktoré vedú k zlým výsledkom a možnej chybnéj interpretácii reálnych fyzikálnych javov.

Spomedzi rôznych prístupov k simulácii laserovej plazmy je práve metóda particle in cell (PIC) používaná veľmi často. Preto sa ňou táto práca zaoberá. Ako bolo spomenuté, tak veľmi dôležité sú nefyzikálne javy vyskytujúce sa v simulácii. Ťažiskom tejto práce je skúmanie nefyzikálneho ohrevu, ktorý nastáva v PIC a môže viesť až k absolútnej nevyužitelnosti celej simulácie. Fyzikálne aj nefyzikálne parametre PIC hrajú úlohu v tom, do akej miery nastane tento ohrev. Je preto výhodné poznať tieto závislosti. Ich znalosť umožňuje nastaviť simuláciu čo najlepšie s ohľadom na požadovanú presnosť a časovú náročnosť simulácie.

Významnou literatúrou k téme particle in cell je monografia [1], ktorá komplexne popisuje problematiku časticových simulácií. Samotnému numerickému ohrevu aj s teoretickým odvodením tohto javu sa bližšie venuje [2]. Tieto publikácie poskytujú dôležité informácie aplikovateľné na reálne simulácie. Žiaľ, majú tú nevýhodu, že v súčasnosti existujú aj modernejšie kódy, než tie, ktorým sa tieto publikácie venujú. Nedá sa teda vždy spoľahnúť na teoretické predpovede.

Cieľom tejto práce je preto predstaviť a overiť teoretické predpoklady pre numerické vlastnosti konkrétneho kódu. Na tento účel je využitý kód s názvom EPOCH vyvinutý na University of Warwick. Jeho výhodou je otvorenosť celého projektu, podpora od originálnych autorov a prehľadnosť, čo značne uľahčuje skúmanie jeho vlastností.

Cieľom tejto práce je samozrejme aj oboznámenie sa so spúšťaním kódu a spracovaním výsledkov. Keďže simulácie sú náročné, je rozumné realizovať ich na clustrovom systéme. Na tento účel sú v tejto práci využité služby Metacentra VO, organizácie, ktorá poskytuje členom akademických inštitúcií v Českej Republike k dispozícii clustrovú výpočtovú techniku. Celkovo sa v rámci tejto práce realizuje viacero simulácií, zameraných hlavne na výskum čisto nefyzikálnych javov v 1D a následne aj v 2D verzii kódu EPOCH, s parametrami nastavenými tak, aby ich neskôr bolo možné aplikovať aj na následné simulácie rôznych fyzikálnych javov.

Kapitola 1

Metóda Particle In Cell

1.1 Základne pojmy

Na popis teórie particle in cell simulácií je nutné byť oboznámený aspoň so základmi fyziky plazmy. Plazma je kvázineutrálny systém nabitých a neutrálnych častíc, ktorý vykazuje kolektívne správanie [3]. Kolektívne správanie znamená, že častice nie sú ovplyvňované len tou najbližšou časticou (napr. zrážky v neutrálnom plyne), ale viacerými časticami prostredníctvom makroskopických polí. Kvázineutralita plazmy sa definuje pomocou Debyovej dĺžky a plazmovej frekvencie. Práve tieto a zopár ďalších pojmov, dôležitých pre fyziku plazmy, sú vysvetlené v tejto sekcii.

Maxwellove rovnice

Pre popis makroskopických polí v plazme používame Maxwellove rovnice (1.1 - 1.4) [4]. Tie sú základom elektrodynamiky. Popisujú ako vznikajú elektrické (\mathbf{E}) a magnetické polia (\mathbf{B}), ako závisia na sebe navzájom, na prúdoch (\mathbf{j}) a na nábojovej hustote (ρ) v priestore.

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (1.1)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (1.2)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.3)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \quad (1.4)$$

Debyová dĺžka

Elektrónová Debyová dĺžka je vzdialenosť, od ktorej je elektrostatický efekt plazmy odtienený. To je možné si predstaviť pomocou vyjadrenia elektrického potenciálu testovacej častice

vloženej do plazmy, ktorý vyzerá nasledovne [3]:

$$\phi_t(r) = \frac{q_t}{4\pi\epsilon_0 r} e^{-r/\lambda_{De}} \quad (1.5)$$

V ňom vystupuje výraz Debyeovej dĺžky. Pre väčšie vzdialenosti môžeme považovať potenciál za zanedbateľný, keďže dostatočne rýchlo klesá k nule. Hovoríme potom o tienení a plazma sa pri väčších vzdialenostiach javí kvázineutrálne, tým je definovaná priestorová podmienka pre plazmu [5]. Veľkosť samotnej elektrónovej Debyeovej dĺžky sa dá vyjadriť pomocou nasledujúceho vzťahu [3]:

$$\lambda_{De} = \sqrt{\frac{\epsilon_0 k_B T_e}{n_e e^2}} \quad (1.6)$$

Plazmová elektrónová frekvencia

Ak si predstavíme, že v plazme sa elektróny vychýlia z rovnovážnej polohy, potom ich elektrostatická sila iónov pritiahne naspäť. Ak využijeme predpoklad, že ióny sú nekonečne ťažké, čo je celkom rozumné keďže sú skoro vždy mnohonásobne ťažšie ako elektróny, tak budú elektróny oscilovať. Elektrónová frekvencia je potom vlastná frekvencia oscilácie elektrónov v plazme [5]. Dá sa vyjadriť nasledujúcim vzťahom:

$$\omega_{pe} = \sqrt{\frac{e^2 n_e}{\epsilon_0 m_e}} \quad (1.7)$$

Pomocou tejto frekvencie je potom možné určiť časovú podmienku pre kvázineutralitu (1.8). Teda kvázineutralita platí, ak sa v systéme prejavujú len deje, ktorých charakteristický čas spĺňa túto podmienku.

$$\tau \gg \omega_{pe}^{-1} \quad (1.8)$$

Landauov útlm

V plazme sa môže šíriť mnoho druhov vln. Jednými z najdôležitejších sú práve elektrónové plazmové vlny. Ich disperzný vzťah sa často uvádza ako (1.9) [6].

$$\omega^2 = \omega_{pe}^2 + 3k^2 v_{Te}^2 \quad (1.9)$$

Pri hlbšej analýze plazmových vln je možné pomocou kinetickej teórie odvodiť, že sú tlmené aj bezzrážkovo. Tento jav sa nazýva Landauov útlm. Pri použití poruchovej teórie sa pri výpočte objaví integrál s pólom v bode ω/k v ktorom figuruje rozdeľovacia funkcia rýchlostí elektrónov. Po využití teórie komplexného integrálu a zopár aproximácií je možné dopracovať sa k disperznému vzťahu (1.10) [3].

$$\omega^2 = \omega_{pe}^2 + i\pi\omega^2 \frac{\omega_{pe}^2}{k^2} \frac{\partial f_0}{\partial v} \Big|_{v=\frac{\omega}{k}} \quad (1.10)$$

V ňom vystupuje f_0 , čo je elektrónová rýchlostná distribučná funkcia, ktorá je v plazme Maxwellovská.

Imaginárny člen vo vzťahu (1.10) spôsobuje už zmieňovaný útlm. Často sa na jeho vysvetlenie využíva analógia so surférom, ktorý sa urýchľuje vlnou. Ak majú elektróny malú, alebo veľkú rýchlosť oproti vlne, tak si s ňou skoro žiadnu energiu nevymenia. Ak však majú elektróny rýchlosť blízku k fázovej frekvencii vlny ($v_\phi = \omega/k$), tak si s vlnou budú vymieňať veľkú energiu. Ak je ich rýchlosť mierne nižšia ako tá fázová, tak sa na úkor vlny zrýchlia, ak je ich rýchlosť mierne vyššia budú naopak brzdené a energiu bude získavať vlna. Keďže podľa Maxwellovského rozdelenia je viac častíc s nižšími rýchlosťami, tak sa v konečnom dôsledku bude vlna tlmiť [7].

1.2 Kinetický popis plazmy

Plazma sa dá popisovať rôznymi spôsobmi. Jednoduché je popisovať plazmu tekutinovým modelom. Ten ale vyžaduje aby plazma mala dobre definovanú teplotu, teda aby mali častice Maxwellovské rýchlostné rozdelenie. Ak toto nie je splnené, je možné obrátiť sa na kinetický, alebo časticový popis plazmy. V kinetickom prístupe sa používa distribučná funkcia $f_s(\mathbf{x}, \mathbf{p}, t)$, kde \mathbf{x} , \mathbf{p} sú vektory pozície a hybnosti vo fázovom priestore. Výraz $f_s(\mathbf{x}, \mathbf{p}, t)d\mathbf{x}d\mathbf{p}$ potom určuje počet častíc typu s vo fázovom objeme $(\mathbf{x}, \mathbf{x} + d\mathbf{x}) \times (\mathbf{p}, \mathbf{p} + d\mathbf{p})$.

Za využitia predpokladu, že f_s je diferencovateľná je možné popísať časový vývoj distribučnej funkcie pre konkrétny typ častice pomocou Vlasovovej rovnice (1.11) [6].

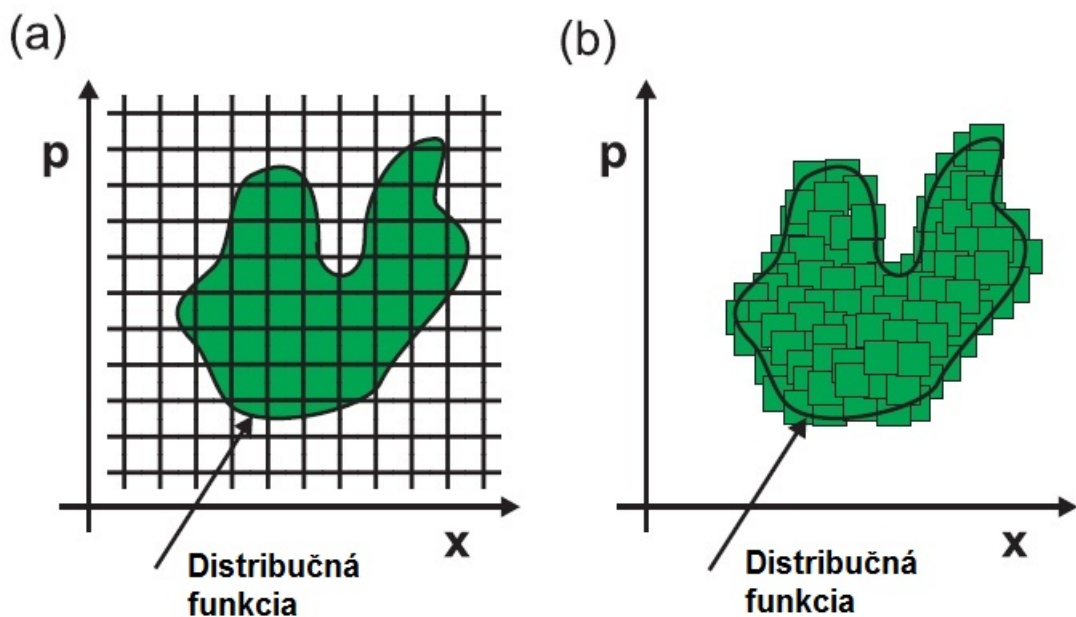
$$\left[\frac{\partial}{\partial t} + \frac{\mathbf{p}}{m_s \gamma} \cdot \frac{\partial}{\partial \mathbf{x}} + q_s \left(\mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right) \cdot \frac{\partial}{\partial \mathbf{p}} \right] f_s = 0 \quad (1.11)$$

Kde f_s je už zmieňovaná distribučná funkcia pre častice typu s , q_s je náboj a m_s hmotnosť jednej častice typu s a γ je relativistický faktor. \mathbf{E} a \mathbf{B} sú vektory vystredovaných makroskopických polí, pomocou nich sa prejaví interakcia rôznych typov častíc. Táto rovnica však zahŕňa iba interakciu pomocou kolektívneho efektu plazmy. V prípade, že zrážky nemajú zanedbateľný vplyv, je nutné pridať na pravú stranu rovnice zrážkový člen.

1.3 Základný princíp PIC

Keďže rovnica (1.11) je vo všeobecnosti v 6-dimenzionálnom fázovom priestore, bol by jej priamy výpočet pomocou numerických metód veľmi náročný. Preto sa v metóde PIC nahradí distribučná funkcia f_s makročasticami [8]. Tento prevod je demonštrovaný pomocou rovnice (1.12).

$$f_s(\mathbf{x}, \mathbf{p}, t) = \sum_{k=1}^N N_k S_x(\mathbf{x} - \mathbf{x}_k) S_p(\mathbf{p} - \mathbf{p}_k) \quad (1.12)$$



Obr. 1.1: Porovnanie priameho výpočtu Vlasovovej rovnice (a) s aproximáciou distribučnej funkcie makročasticami (b).

Kde N_k je počet skutočných častíc v danej makročastici, x_k a p_k sú polohy makročastic vo fázovom priestore. S_x a S_p sú tvarové funkcie makročastic. Výhoda tohto prístupu oproti priamemu riešeniu Vlasovovej rovnice spočíva v tom, že sa výpočet obmedzí na tú časť fázového priestoru, kde sa častice nachádzajú, ako je demonštrované na obrázku 1.1.

Pri obmedzení sa na časť fázového priestoru pomocou makročastic je však stále nutné na priestorovej eulerovskej mriežke počítať elektrické a magnetické polia. Aj napriek tomu je redukcia veľkosti výpočtovej oblasti významná.

1.4 Tvarové funkcie

Výber tvarových funkcií má výnamný vplyv na vlastnosti simulácie. Tieto funkcie musia mať viacero vlastností.

Aby funkcia dávala význam rozloženia častíc v priestore, musí byť nezáporná a splňať podmienku:

$$\int_{-\infty}^{+\infty} S_x(\mathbf{x})d\mathbf{x} = 1, \quad (1.13)$$

Ďalej je dôležité, aby mala dostatočne malý nosič. Najčastejšie býva nosič interval so stredom v nule a šírkou malého násobku veľkosti eulerovskej bunky. Čím je nosič väčší tým viac buniek jedna makročastica prekryje a to sa odzrkadlí aj v presnosti a rýchlosti výpočtu.

Väčšinou sú tvarové funkcie symetrické, aj keď to nieje podmienkou. Najčastejšie sa pre tvarové funkcie v priestore využívajú b-spliny rôznych stupňov, vid' obr. 1.2. Spomedzi nich sa najčastejšie využíva práve b-spline prvého stupňa, keďže vedie k dostatočne hladkému riešeniu a zároveň je výpočet s ním vďaka jednoduchosti relatívne rýchly.

Tieto spliny sa získavajú postupne zo splinu nultého stupňa popísaného pomocou (1.14).

$$S_0(x) = \begin{cases} 1, & \text{pre } |x| \leq \frac{\Delta x}{2} \\ 0, & \text{inak} \end{cases} \quad (1.14)$$

Následne ďalšie spliny sú konvolúcie S_0 so sebou, t.j. napríklad spline prvého stupňa $S_1(x) = S_0 * S_0$ [9]. Pre popis numerických efektov, ktorým sa venuje koniec tejto kapitoly je vhodné využiť Fourierovsky transformované spliny vyjadrené ako (1.15) [1].

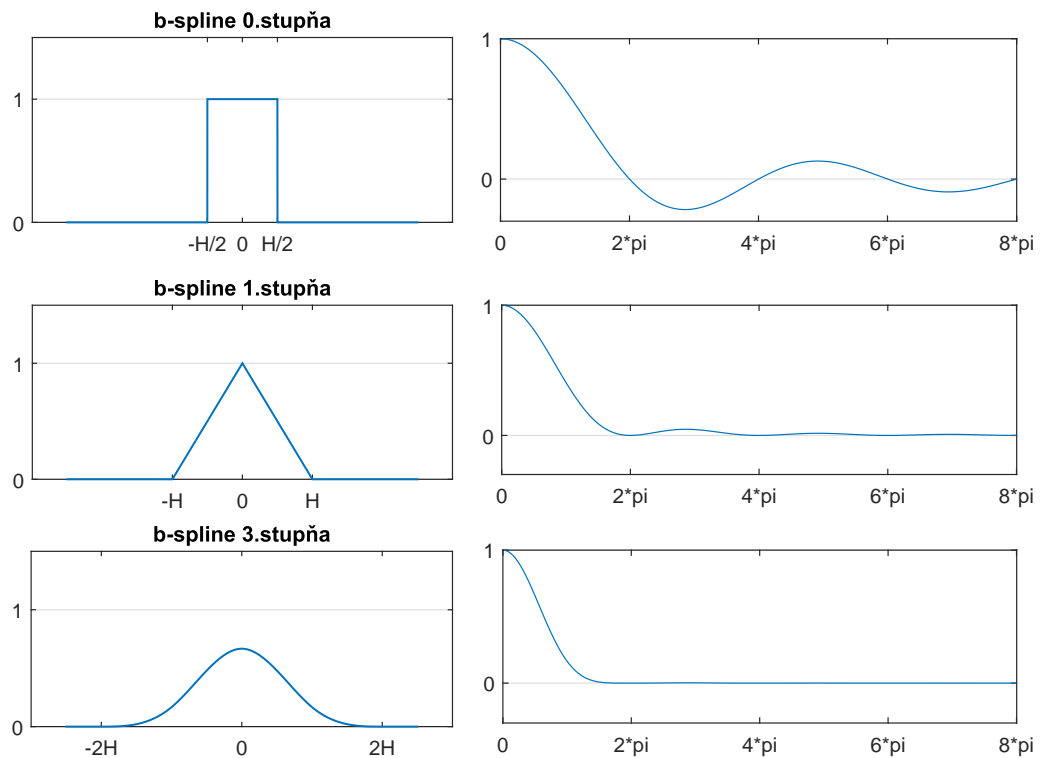
$$S_m(k) = \left(\frac{\sin\left(\frac{k\Delta x}{2}\right)}{\frac{k\Delta x}{2}} \right)^{m+1} \quad (1.15)$$

Voľba týchto tvarových funkcií je dôležitá pre presnosť simulácie. Preto je vhodné ich presnosť odstupňovať. Na to je možné využiť skutočnosť, že v simulácii sa prejaví iba pri interpolácii vlastností častíc na mriežku a naopak. Označenie preto súvisí s mocninou, s akou sa v interpolačných vzorcoch prejaví vzdialenosť stredy makročastice od najbližšej mriežky. Ako bude v jednej z nasledujúcich sekcií vidno, tak pre prípad b-splinu prvého stupňa sa v interpolačnom vzorci objaví druhá mocnina, ide teda o interpoláciu druhého stupňa. Podobne pre b-spline nultého stupňa je to interpolácia lineárna, čiže prvého stupňa. Pre b-spline tretieho stupňa, ktorá sa v tejto práci tiež používa, zas ide o interpoláciu štvrtého stupňa. Dôležité je teda mať na pamäti, že stupeň b-splinu je o jedna nižší ako stupeň interpolácie, ktorú daným splinom dosiahneme. Treba ešte dodať, že v rôznej literatúre sa tieto stupne označujú odlišne. Preto toto označenie podľa mocniny v interpolačnom vzorci nemusí byť súhlasné s označovaním v inej literatúre.

Pre rýchlostné rozdelenie sa najčastejšie používa Diracova δ -funkcia, t.j. všetky častice, ktoré makročastica predstavuje, majú rovnakú rýchlosť. Toto zabezpečí, že priestorová tvarová funkcia sa s časom meniť nebude [10].

1.5 Výpočtový cyklus

Metóda PIC má relatívne jednoduchý výpočtový cyklus. Začína inicializáciou a pokračuje opakujúcimi sa štyrmi krokmi, ktoré sa opakujú až do konca simulácie. Pri popise jednotlivých krokov sú v nasledujúcom texte opísané hlavne numerické metódy, ktoré využíva simulačný kód EPOCH, ktorému sa bližšie venuje kapitola 2.



Obr. 1.2: Vľavo: b-spliny rôznych stupňov, H označuje veľkosť bunky. Vpravo: transformované spliny, pri vyššom stupni interpolácie sa znižuje vplyv aliasingu na simuláciu [8].

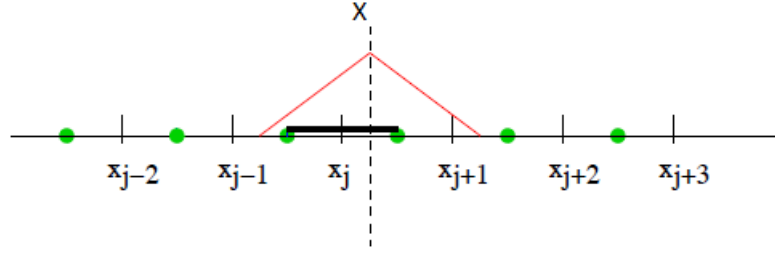
1.5.1 Inicializácia

V inicializačnej fáze sa vytvorí eulerovská mriežka. Podľa počiatočných podmienok sa potom v priestore rozmiestnia častice. V prípade plazmy je dôležité aby častice boli hlavne v hybnostnej časti fázového priestoru rozložené podľa fyzikálneho rozdelenia, tým je najčastejšie Maxwellovské rozdelenie. Ďalej je nutné nainicializovať elektrické a magnetické polia. Mali by spĺňať rovnice (1.1) a (1.2) v konečne-diferencovateľnej forme na mriežke a okrajové podmienky na okrajoch eulerovskej mriežky. Po inicializácii sa začne cyklus prvým krokom.

1.5.2 Prvý krok - interpolácia elektrických a magnetických polí na častice

Keďže makročastice sa pohybujú voľne v priestore a elektrické a magnetické polia sú známe len na mriežke, je nutné interpolovať tieto polia na častice. Vďaka znalosti tvarových funkcií je možné analyticky určiť, ako sa prejavujú polia z buniek, ktoré častica prekrýva, na celkovej sile, pôsobiacej na celú makročasticu [11].

V prípade 1D simulácie za použitia b-splinu prvého stupňa a predpokladu, že polia sú definované v stredoch buniek (je nutné si uvedomiť, že elektrické a magnetické polia sú definované



Obr. 1.3: Makročastica v tvare b-splinu 1. stupňa so stredom v bode X

na rôznych miestach, preto bunky pre elektrické pole su tiež posunuté voči tým pre magnetické pole) a v rámci jednej bunky je pole konštatné, je možné rátať vplyv poľa na časticu pomocou integrácie cez všetky 3 bunky, ktoré častica pokryje. Ak sa stred makročastice c nachádza v bode X , pre ktorý je najbližší bod x_j , kde je definované pole F_j (vid' obr.1.3), potom bude vplyv poľa na časticu daný nasledujúcou integráciou:

$$F_c = \frac{1}{\Delta x} \left[\int_{X-\Delta x}^{x_j-\frac{\Delta x}{2}} F_{j-1} \left(1 - \frac{X-x}{\Delta x}\right) dx + \int_{x_j-\frac{\Delta x}{2}}^X F_j \left(1 - \frac{X-x}{\Delta x}\right) dx \right. \\ \left. + \int_X^{x_j+\frac{\Delta x}{2}} F_j \left(1 - \frac{x-X}{\Delta x}\right) dx + \int_{x_j+\frac{\Delta x}{2}}^{X+\Delta x} F_{j+1} \left(1 - \frac{X-x}{\Delta x}\right) dx \right] \quad (1.16)$$

Kde pre F_j je hodnota poľa v bunke so stredom v x_j . Výraz sa dá ďalej upraviť na:

$$F_c = \frac{1}{\Delta x} \left[\frac{1}{2} F_{j-1} \left(\frac{1}{2} + \frac{x_j - X}{\Delta x}\right)^2 + F_j \left(\frac{3}{4} - \frac{(x_j - X)^2}{\Delta x^2}\right) + \frac{1}{2} F_{j+1} \left(\frac{1}{2} - \frac{x_j - X}{\Delta x}\right)^2 \right] \quad (1.17)$$

Z tohto odvodenia vidno, že ak je tvarová funkcia b-spline prvého stupňa, tak interpolácia bude druhého stupňa. Podobným spôsobom môžeme interpoláciu polí vyjadriť aj pre iné tvarové funkcie. V prípade interpolácie štvrtého stupňa bude nutné integrovať cez 5 buniek a výpočet sa tým skomplikuje. Pri 2D a 3D simuláciách bude dopad tvarových funkcií na náročnosť ešte väčší.

1.5.3 Druhý krok - posun častíc

V tomto kroku sa častice posunú o diskretný krok podľa schémy odvodenej z relativistických pohybových rovníc (1.18 - 1.20) pre časticu k . Niektoré PIC kódy sú nerelativistické, v tom prípade sa za parameter γ dosadí 1.

$$\frac{\partial \mathbf{x}_k}{\partial t} = \frac{\mathbf{p}_k}{\gamma m_k} \quad (1.18)$$

$$\frac{\partial \mathbf{p}_k}{\partial t} = q_k \left(\mathbf{E}_k + \frac{\mathbf{p}_k}{\gamma m_k} \times \mathbf{B}_k \right) \quad (1.19)$$

$$\gamma = \sqrt{1 + \left(\frac{\mathbf{p}_k}{m_k c} \right)^2} \quad (1.20)$$

Je možné využiť rôzne schémy na diskrétny posun častíc. Často sa využíva metóda leap-frog v ktorej sa hybnosť počíta vždy o pol časového kroku neskôr ako pozícia, čo veľmi jednoducho vedie k presnejším a stabilnejším riešeniam.

V kóde EPOCH sa využíva táto schéma, ale zapisujú sa hodnoty pozície aj v tom istom čase ako hybnosti [11]. Pre prechod z času $t = t(n)$ do $t = t(n+1)$ sa ešte pred interpoláciou polí posunie častica do pozície $\mathbf{x}^{n+1/2}$ pomocou \mathbf{p}^n . V čase $t = t(n+1/2)$ sa interpolujú polia na makročastice. Z interpolácie sa vypočíta \mathbf{p}^{n+1} a následne dôjde k posunu z $\mathbf{x}^{n+1/2}$ na \mathbf{x}^{n+1} pomocou \mathbf{p}^{n+1} . Tento posun častíc je v konečnom dôsledku druhého rádu presnosti [10].

1.5.4 Tretí krok - výpočet prúdových hustôt

V tomto kroku sa na základe posunu makročastíc vypočíta prúdová a nábojová hustota. Nábojová hustota sa dá vypočítať postupom podobným s prvým krokom. V tomto prípade, ale interpolujeme náboj z makročastice na mriežku. V 1D prípade pre interpoláciu 2. stupňa sa potom jednotlivé častice prejavia na mriežku nasledovne [11]:

$$F_c(x_j) = F_c \cdot \begin{cases} \frac{3}{4} - \frac{|X-x_j|^2}{\Delta x^2}, & |X-x_j| \leq \frac{\Delta x}{2} \\ \frac{1}{2} \left(\frac{3}{2} - \frac{|X-x_j|}{\Delta x} \right)^2, & \frac{\Delta x}{2} < |X-x_j| \leq \frac{3\Delta x}{2} \\ 0, & \frac{3\Delta x}{2} < |X-x_j| \end{cases} \quad (1.21)$$

Výpočet prúdových hustôt je náročnejší. V prípade EPOCHu sa využíva Villasenorova-Bunemanova schéma. V rámci nej sa vždy po posune častíc z $\mathbf{x}^{n-1/2}$ na $\mathbf{x}^{n+1/2}$ vyráta $\partial \rho / \partial t$ v druhom stupni presnosti. Zo zákona zachovania náboja (1.22) sa potom prevedením priestorových derivácií na jednostranné konečné diferencie vypočíta prúdová hustota na hranách buniek. Výhoda tohto prístupu je, že Poissonova rovnica (1.23) sa využije len počas inicializácie a ak je splnená na začiatku tak bude splnená aj v každom čase [12].

$$\nabla \cdot \mathbf{j} = \frac{\partial \rho}{\partial t} \quad (1.22)$$

$$\nabla^2 \phi = \frac{\rho}{\epsilon_0} \quad (1.23)$$

1.5.5 Štvrtý krok - výpočet elektrického a magnetického poľa

V poslednom a najkomplikovanejšom kroku je nutné vypočítať zmeny elektrického a magnetického poľa. V jednoduchých 1D elektrostatických kódach je napríklad možné použiť Poissonovu rovnicu (1.23) a metódou konečných diferencií ju vypočítať [1]. V prípade nezanedbania magnetického poľa sa výpočet skomplikuje.

V EPOCHu sa využíva schéma FDTD. Polia sú určené na Yeeho mriežke, t. j. v 3D prípade sú elektrické polia vyjadrené v stredoch hrán kockovitej bunky a magnetické polia v stredoch jej stien [13]. Postupným výpočtom po predchádzajúcom kroku v ktorom získame \mathbf{j}^n a využitím (1.3) a (1.4) postupujeme nasledovne:

$$\begin{aligned}\mathbf{B}^n &= \mathbf{B}^{n-1/2} - \frac{\Delta t}{2} (\nabla^+ \times \mathbf{E}^{n-1/2}) \\ \mathbf{E}^n &= \mathbf{E}^{n-1/2} + \frac{\Delta t}{2} (c^2 \nabla^- \times \mathbf{B}^n - \mathbf{j}^n) \\ \mathbf{E}^{n+1/2} &= \mathbf{E}^n + \frac{\Delta t}{2} (c^2 \nabla^+ \times \mathbf{B}^n - \mathbf{j}^n) \\ \mathbf{B}^{n+1/2} &= \mathbf{B}^n - \frac{\Delta t}{2} (\nabla^- \times \mathbf{E}^{n+1/2})\end{aligned}\tag{1.24}$$

Kde ∇^+ a ∇^- sú diskretizované operátory ∇ ktoré vychádzajú z umiestnenia zložiek polí v Yeeho mriežke, ktoré sa líšia tým, že jeden je vyjadrený pomocou doprednej diferencie a druhý pomocou spätnej diferencie [10].

Aby bola táto schéma kompletná, je ešte nutné nadefinovať okrajové podmienky. Tie sa dajú popísať rôznymi spôsobmi a kód EPOCH ich zvláda viacero. Bližšie sa im venuje sekcia 1.6.1.

1.6 Stabilita a Numerický ohrev

Pri všetkých numerických metódach je dôležitá stabilita a podmienky, za akých ju je možné dosiahnuť. Nestabilná metóda je taká, ktorá postupným nabalovaním chýb v riešení vedie k fyzikálne nezmyselným výsledkom [14]. Táto nestabilita sa v PIC metóde často prejavuje nefyzikálnym nárastom celkovej energie systému, ktorý označujeme ako numerický ohrev. Aby k tejto nestabilite nedošlo, je nutné obmedziť rôzne parametre.

1.6.1 Parametre

Metóda PIC má pár základných parametrov, ktoré určujú presnosť, stabilitu a náročnosť výpočtu. Najzákladnejšími parametrami sú **časový krok** Δt a **priestorový krok**, resp. šírka bunky v mriežke Δx , v 2D a 3D kódach ešte Δy , resp. Δz . Vo väčšine prípadov ale platí rovnosť $\Delta x = \Delta y = \Delta z$, kvôli ktorej v nasledujúcom texte budeme priestorový krok označovať iba ako Δx .

Ďalším parametrom je **počet častíc na bunku**, resp. počet častíc v simulácii. Počet častíc na bunku reprezentuje koľko makročastíc sa nainicializuje v každej bunke, ktorá má nenulovú

hustotu. Väčšinou sa aplikuje aj kritérium, podľa ktorého sa v bunkách s menšou ako určenou limitnou hustotou makročastice neinicializujú, prípadne sa ich v danej bunke nainicializuje menej.

Ako bolo spomenuté v predchádzajúcej sekcii, tak veľmi dôležité sú aj **okrajové podmienky**. Okrajové podmienky sú dôležité najmä pre výpočet polí. Existuje viacero variantov okrajových podmienok. Medzi základné patria periodické, otvorené a okraje predstavujúce perfektne vodivé steny na okrajoch [1]. Pre častice je tiež nutné rozhodnúť, čo sa stane v prípade, že "vyletia" zo simulačnej oblasti. Opäť existuje viacero možností, napríklad odraziť častice naspäť, vrátiť ich z opačného okraja, alebo ich od vyletenia ignorovať.

Významným faktorom pre numerické vlastnosti je aj voľba tvarovej funkcie. V mnohých kódoch je tvarová funkcia daná pevne. Simulačný kód EPOCH je možno kompilovať s tvarovými funkciami 1., 2. a 4. stupňa interpolácie. Preto môžeme v rámci tejto práce považovať stupeň interpolácie za parameter simulácie, aj keď to v pravom zmysle slova parameter nie je.

1.6.2 CFL podmienka

Pri elektromagnetických simuláciach je dôležité splniť Courant-Fridrichs-Lewyho (CFL) podmienku [15]. Pre prípad PIC simulácií je ju možné vyjadriť jednoduchým predpisom (1.25). Bez tejto podmienky by sa elektromagnetické vlny šírili nefyzikálne. Ďalším dôsledkom tejto podmienky je, že častica sa nemôže dostať za časový krok ďalej ako do susednej bunky.

$$c\Delta t \leq \Delta x \quad (1.25)$$

V EPOCHu je táto podmienka aplikovaná pre 2D prípad nasledovne:

$$\Delta t = \frac{0.95}{c} \cdot \frac{\Delta x \cdot \Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} \quad (1.26)$$

1.6.3 Podmienka plazmovej frekvencie

Aby sa mohli v plazme šíriť aj plazmové vlny, je potrebné overiť si, ako sa bude oscilácia správať v numerickej schéme leap-frog. Pri osciláciach ide o riešenie rovnice (1.27).

$$\frac{d^2 x}{dt^2} = -\omega_{pe}^2 x \quad (1.27)$$

Pre odvodenie riešenia v konečne diferencovanom tvare sa s pomocou rozdielu (1.29) a (1.28) dosadí do vyjadrenia derivácie rýchlosti a získa sa tak rovnica (1.30).

$$x^{n+1/2} = x^{n-1/2} + \Delta t \cdot v^n \quad (1.28)$$

$$x^{n+3/2} = x^{n+1/2} + \Delta t \cdot v^{n+1} \quad (1.29)$$

$$-\omega_{pe}^2 x^{n+1/2} = \frac{dv}{dt}(n+1/2) = \frac{v^{n+1} - v^n}{\Delta t} = \frac{x^{t+3/2} - 2x^{t+1/2} + x^{t-1/2}}{\Delta t^2} \quad (1.30)$$

Za predpokladu, že riešenie tejto konečne diferencovateľnej schémy bude v tvare $x = Ae^{-i\omega t}$, je možné do (1.30) dosadiť ako $Ae^{-i\omega(t-\Delta t/2)}$, $Ae^{-i\omega(t+\Delta t/2)}$ a $Ae^{-i\omega(t+3\Delta t/2)}$. Po jednoduchých úpravách využívajúcich exponenciálny tvar druhej mocniny sínusu potom vypláva vzťah (1.31), ktorý znázorňuje rozdiel medzi ω v simulácii a fyzikálnym ω_{pe} .

$$\frac{\omega_{pe}\Delta t}{2} = \pm \sin\left(\frac{\omega\Delta t}{2}\right) \quad (1.31)$$

Z tejto rovnice je zjavná podmienka plazmovej frekvencie (1.32) na časový krok. Ak by nebola splnená, tak by rovnica (1.31) mala aj imaginárne korene. To by spôsobilo, že amplitúda oscilácií by sa menila a to by viedlo k nestabilite.

$$\omega_{pe}\Delta t \leq 2 \quad (1.32)$$

Aj v prípade zachovania podmienky (1.32) sa bude od reality líšiť frekvencia plazmových vln, preto sa často oplatí ju ešte zosilniť.

EPOCH má podobne, ako predchádzajúcu podmienku, aj variant tejto implementovanú. V kóde sa vypočíta frekvencia podľa disperzného vzťahu (1.9), do ktorého bol chybné dosadený dvojnásobok najväčšieho vlnového čísla na mriežke (1.33).

$$\omega^2 = \omega_{pe}^2 + 3\left(\frac{2\pi}{\Delta x}\right)^2 \max\left(\frac{k_B T_i}{m_i}\right) \quad (1.33)$$

Funkcia maxima v tomto prípade prebieha cez všetky počiatkové teploty a hmotnosti typov častíc, ako sú definované jednotlivými blokmi species, ktoré môžu mať rôzne teploty a aj hmotnosti. V bežných simuláciách ale toto maximum pripadne práve na teplotu a hmotnosť elektrónov, keďže zvyčajne mávajú najväčšiu tepelnú rýchlosť.

Po tomto výpočte je aplikovaná podmienka, taká aby informácia nemohla za jeden časový krok cestovať ďalej ako do susednej bunky [16]. To sa dá reprezentovať nasledovne:

$$\Delta t \leq \frac{\Delta x}{v_\phi} = \frac{\Delta x \cdot k_{max}}{\omega} = \frac{\Delta x \cdot \pi}{\Delta x \cdot \omega} = \frac{\pi}{\omega} \quad (1.34)$$

Pre splnenie tejto podmienky vypočíta EPOCH časový krok podľa (1.35). Ak je tento časový krok menší ako ten určený (1.26), tak ho vyberieme ako časový krok pre simuláciu.

$$\Delta t = 0.95 \cdot \frac{\pi}{\omega} \quad (1.35)$$

V prípade, ak toto nastavenie časového kroku nastane, tak s veľkou pravdepodobnosťou nebude splnená podmienka (1.32). Navyše informácia sa nemôže šíriť rýchlejšie ako svetlo, takže by mala stačiť CFL podmienka. Preto sa javí táto podmienka ako zle nastavená.

V rámci tejto práce bol časový krok vždy nastavený automaticky podľa (1.26). Takže boli simulované aj prípady kde podmienka (1.32) splnená nebola. Praktické dopady jej nesplnenia skúma 3. kapitola.

1.6.4 Dôsledky diskkrétnej mriežky

Na odvodenie dôsledkov priestorovej diskretizácie sa využíva vyjadrenie rôznych veličín na mriežke pomocou Fourierovej transformácie. Vďaka nej si možno všimnúť efekt aliasingu [17], ktorý znamená, že rôzne módy oscilácií sa nedajú diskretizovať tak, aby bolo možné ich odlíšiť. Následkom toho módy s vyššími frekvenciami splynú na módy nižších frekvencií. Rovnica (1.36) určuje najvyššie vlnové číslo podporované mriežkou a k_p z rovnice (1.37) sú vlnové čísla, ktoré splývajú s k [18].

$$k_{max} = \frac{\pi}{\Delta x} \quad (1.36)$$

$$k_p = k - 2 \cdot p \cdot k_{max} \quad (1.37)$$

Pri výpočte sa potom vlnové čísla väčšie ako k_{max} previažu s vlnovými číslami k_p s faktorom $S(k_p)$, čo je fourierovsky transformovaná tvarová funkcia (vid' obr. 1.2). Pri analýze disperzného vzťahu jeho imaginárna časť výjde ako (1.38), kde f_0 je rýchlostná distribučná funkcia a suma sa sčíta cez všetky celé čísla [1].

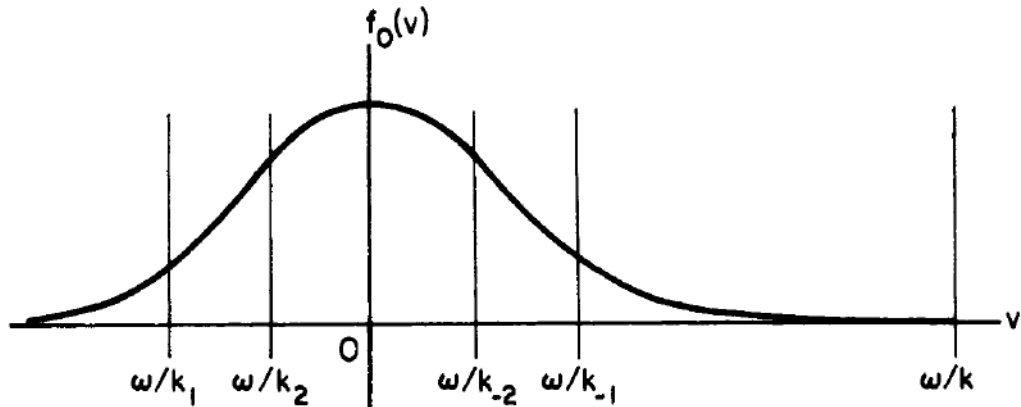
$$Im(\epsilon(\omega, k)) = -\pi \frac{\omega_{pe}^2}{k^2} \sum_p S^2(k_p) \frac{kk_p}{k_p^2} \frac{\partial f_0(\omega/k_p)}{\partial v} \quad (1.38)$$

Tento výraz by odpovedal Landauovmu útlmu, ak by v ňom nevystupovala suma. Tá zodpovedá práve previazaniu módov na diskkrétnej mriežke. Toto previazanie vedie k nefyzikálnej nestabilite, keďže energia, o ktorú vlna príde nie je zhodná s tou, ktorú získajú častice. Je možné si tento efekt predstaviť pomocou obr. 1.4, kde vidno, že prípade ak sa nesplní podmienka (1.39), tak sa previaže Landauovský netlmený mód s tými tlmenejšími.

$$\frac{\omega_{pe}}{k_{max}} \leq v_T \Rightarrow \Delta x \leq \frac{v_T \pi}{\omega_{pe}} \Rightarrow \Delta x \leq \pi \lambda_{De} \quad (1.39)$$

Ako je vidno, tak previazanosť závisí od druhej mocniny fourierovsky transformovanej tvarovej funkcie. Preto aj podmienka (1.39) môže byť nepodstatná pre vyššie stupne interpolácie. Každopádne pre prvý stupeň interpolácie existuje približná hodnota C , tak že pri nesplnení podmienky (1.40) sa plazma bude rapídne ohrievať dovtedy, kým v nej nebude približne platiť rovnosť [9, 17] (treba si uvedomiť že Debyova dĺžka sa zvyšuje s teplotou). V prípade, že toto nastalo, je možné považovať simuláciu za skrachovanú, keďže sa v nej prejavil drastický nefyzikálny ohrev.

$$\Delta x \leq C \cdot \lambda_{De} \quad (1.40)$$



Obr. 1.4: Fázová rýchlosť vln ω/k a ich aliasy ω/k_p , ak platí $k_g v_t \sim \omega_p$ vzhľadom k distribučnej funkcii rýchlosti [1].

V prípade splnenia podmienky (1.40) sa aj tak bude plazma nefyzikálne ohrievať, ale tento ohrev nespôsobí úplnú nepoužitelnosť simulácie. Tento ohrev nastane aj pre simulácie s vyššími stupňami interpolácie, bude ale slabší.

1.6.5 Numerický ohrev

V predchádzajúcich odstavcoch boli rozobraté podmienky, ktoré je nutné splniť, aby nedošlo k drastickému ohrevu, t. j. znásobeniu celkovej energie systému za pár časových krokov. Aj pri splnení týchto podmienok sa vyskytuje v simuláciách šum a nepresnosti. Tie vedú k lineárnemu zvyšovaniu celkovej energie systému [9]. Na účely určenia miery tohoto ohrevu je možné zaviesť koeficient ohrevu (1.41).

$$r_n = \frac{E_{t(n+1)} - E_{t(n)}}{E_{t(0)}} \quad (1.41)$$

V rámci tejto definície sa posudzuje ohrev iba v rámci jedného časového kroku. Pre účely tejto práce ale budeme v tretej kapitole používať koeficient v ktorom $\Delta t = 10$ fs pre všetky počítačové podmienky. Výhoda tohto prístupu je, že je z koeficientu hneď zrejmé, o koľko sa systém "ohreje" za konkrétny čas.

V literatúre [2] je možné nájsť teoretické vyjadrenie strednej hodnoty daného koeficientu ako (1.42), kde N_{ppc} je počet makročastíc na bunku a K je konštanta, ktorá závisí od konkrétnych algoritmov kódu a voľby tvarových funkcií.

$$\langle r \rangle = K \cdot \frac{1}{N_{ppc}} \cdot \left(\frac{\Delta x}{\lambda_{De}} \right)^2 \quad (1.42)$$

Experimentálne zistený koeficient ohrevu (1.43) podľa [19] sa od (1.42) líši mocninou pri podiele veľkosti bunky a Debyovej dĺžky. Treba ale spomenúť, že tento vzťah bol nadobudnutý

pomocou simulácie s prvým stupňom interpolácie a inou metódou riešenia Maxwellových rovníc ako využíva EPOCH.

$$\langle r \rangle = K \cdot \frac{1}{N_{ppc}} \cdot \left(\frac{\Delta x}{\lambda_{De}} \right)^3 \quad (1.43)$$

Znalosť vzťahu pre koeficient ohrevu je užitočná pre simulácie reálnych fyzikálnych javov. Aj preto sa v rámci praktickej časti tejto práce skúma práve koeficient ohrevu.

Kapitola 2

Implementácia

2.1 EPOCH

Extendable PIC Open Collaboration, skrátene EPOCH, je paralelizovaný relativistický particle in cell kód vyvinutý na University of Warwick. Je napísaný vo fortrane a pre paralelizáciu používa MPI (Message Passing Interface). Je postavený na jadre staršieho kódu PSC (particle simulation code) [20] napísaného Hartmutom Ruhlom [10], ktorého základné algoritmy sú popísané v predchádzajúcej kapitole.

Posledná verzia je označená ako 4.4.7 vydaná siedmeho januára 2015. V tejto práci sa využíva staršia verzia 4.3.10, ktorá bola neskôr upravená a využitie novej verzie by nemalo prínos. Deviateho februára 2015 bolo oznámené ukončenie plnej podpory a vývoja tímom na University of Warwick. Napriek tomu sa však originálni vývojári zaoberajú otázkami na diskusnom fóre a predpokladá sa, že v budúcnosti ojedinele výjdu aj ďalšie verzie. Absencia plnej podpory EPOCH-u neuberá na využiteľnosti, keďže jeho modifikácia je jednoduchá a komunita používateľov je dostatočne aktívna. V snahe uľahčiť kolaboráciu bol v máji 2015 projekt presunutý z CCPForge [21] na GitLab rozhranie [22] spravované priamo University of Warwick, kde sú po prihlásení dostupné kódy, diskusie a ďalšie materiály k EPOCHu.

Okrem samotného numerického výpočtu má EPOCH dynamický load balancer, ktorý rozdeľuje výpočet na viacero procesorov a zabezpečuje efektívne využitie dostupných zdrojov. Bez load balancingu by sa mohlo stať, že časť procesorov nečinne čaká na to, kým zvyšné dopočítajú svoju časť úlohy, čo by predĺžilo dobu celého výpočtu.

EPOCH je k dispozícii v jednej až troch priestorových dimenziách, pričom celý kód pre každú dimenziu je vo vlastnej zložke a je nutné ich kompilovať samostatne. Pri kompilácii je nutné mať k dispozícii MPI1.2 kompatibilné knižnice. V makefile je možné upraviť niektoré vlastnosti výsledného programu, spomedzi nich je pre túto prácu najdôležitejšia možnosť využitia rôznych tvarových funkcií.

2.1.1 input.deck

Počiatkové podmienky a všeobecné nastavenia simulácie sa určujú pomocou textového súboru input.deck. Samotný súbor sa skladá z viacerých blokov, v ktorých sa nastavujú rôzne parametre simulácie. Celkovo je zápis počiatkových podmienok jednoduchý. Ide v princípe o nastavenie niekoľkých konštánt, ktoré EPOCH využije počas behu. K tomu je možné používať interne definované a jednotkové konštanty, jednoduché podmienky, logické a matematické funkcie a importáciu iných .deck súborov. Celý EPOCH pracuje aj interne v sústave SI, jediné výnimky sú veľkosť náboja, ktorá sa uvádza v násobkoch elementárneho náboja a hmotnosť, ktorá sa uvádza v násobkoch hmotnosti elektrónu. Nasledujúci text je ukážkou tohto súboru pre 2D simuláciu s rozmerom eulerovskej mriežky $\Delta x = 8$ nm.

```
1 begin: control
2   nx = 1000
3   ny = 1000
4
5   # final time of simulation
6   t_end = 100 * femto
7
8   # size of domain
9   x_min = -4 * micron
10  x_max = -x_min
11  y_min = -4 * micron
12  y_max = -y_min
13 end: control
14
15 begin: constant
16  dens = 7e23 / cc
17  thick = 100 * nano
18  height = 2 * micron
19 end: constant
20
21 begin: boundaries
22  bc_x_min = periodic
23  bc_x_max = periodic
24  bc_y_min = periodic
25  bc_y_max = periodic
26 end: boundaries
27
28 begin: species
29  name = electrons
30  npart_per_cell = 800
31  charge = -1
32  mass = 1.0
33  temp_ev = 500
34  density = if((abs(x) lt thick) and (abs(y) lt height), dens, 0.0)
35 end: species
36
37 begin: species
38  name = protons
```

```

39   npart_per_cell = 800
40   charge = 1
41   mass = 1836.2
42   temp_ev = 500
43   density = density( electrons )
44 end: species
45
46 begin: output
47   dt_snapshot = 10 * femto
48   total_energy = always
49 end: output

```

V `control` bloku sa nastavujú základné vlastnosti simulácie ako veľkosť simulačnej oblasti, počet buniek a časová dĺžka simulácie.

V `constant` bloku sa nastavujú interné konštanty ktoré sú užitočné hlavne pri opakovaní hodnôt na viacerých miestach. V rámci tejto práce bolo veľmi užitočné, z dôvodu väčšieho množstva simulácií spúšťaných naraz, využiť kombináciu bloku `constant` a príkazu `import` ktorý jednoducho vloží na dané miesto v súbore obsah iného. Nasledujúci krátky príklad ukazuje `input.deck`, ktorý upravuje iba veľkosť buniek a využíva separátny súbor v nadradenom adresári na popis ostatných parametrov.

```

1 begin: constant
2   nxset = 500
3 end: constant
4
5 import: ../ all .deck

```

Týmto sa docieľi, že napríklad pri spúšťaní 10 simulácií dávkovo s rôznymi rozmermi bunky stačí pre zmenu teploty upraviť len jeden súbor. Pri využití príkazu `import` treba podotknúť, že v užívateľskom manuáli je chyba a namiesto príkazu `import` je tam uvedený príkaz `include`, ktorý vo verzii 4.3.10 nefunguje.

V `boundary` bloku sa nastavujú podmienky na okrajoch. Ako bolo spomenuté, tak tieto podmienky môžu mať významný dopad na vlastnosti simulácie.

Ďalší blok je `species`, v ktorom sa definujú vlastnosti častíc na začiatku simulácie. Každý typ častíc má vlastný blok. Rôzne častice musia mať rôzne názvy kvôli diagnostike. Zaujímavosťou sa definuje rozloženie častíc v priestore. V tomto konkrétnom prípade sa používa jednoduchá `if` funkcia. Pre komplikované počítačové rozloženie častíc je možné použiť viaceré logické a matematické funkcie.

Dôležitý blok je `output`. V ňom sa definuje, ako často sa vypisuje výstup a ktoré dáta sa vypíšu. EPOCH poskytuje výpis všetkých interných hodnôt ako sú hybnosti, polia atď. Je možné vypísať aj zopár hodnôt, ktoré je nutné špeciálne vyrátať (ako napríklad hustota po bunkách).

Okrem týchto blokov je možné využiť mnoho ďalších. Spomedzi nich je pre jednoduché použitie kódu najdôležitejší `laser` blok, v ktorom sa definujú vlastnosti laseru, ktorý ale nebude v tejto práci využitý.

2.1.2 Úprava diagnostiky

V predchádzajúcej časti bolo uvedené, že v rámci `output` bloku je možné definovať aj výstup, ktorý sa vypočítava dodatočne. Pre túto prácu je vhodné počítať celkovú energiu systému. Ideálne rozdelenú na energiu polí a kinetické energie jednotlivých typov častíc. Podľa manuálu je možné na tento účel použiť riadok `total_energy_sum = always` v `output` bloku.

Problém je, že pri využití tohto výpisu sa všetky energie vypíšu nulové. To je spôsobené tým, že v zdrojovom kóde, konkrétne v súbore `src/io/diagnostics.F90`, chýbalo zavolanie samotného výpočtu pred riadkami ktoré zapíšu vypočítané hodnoty do výstupového súboru. Táto chyba sa dala odstrániť jednoduchým pridaním riadku pred zápis. Aj po oprave má tento výpis nedostatky. Nevypisuje energie elektrického a magnetického pol'a jednotlivo, ale spoločne. Takisto aj kinetická energia je zapísaná pre všetky typy častíc spolu.

Pre pridanie novej diagnostiky, takej, ktorá nemá horezmieňované nedostatky, stačí zopár jednoduchých úprav. Základ je pridať do už spomínaného `src/io/diagnostics.F90` podmienku pre prípad keď v `input.deck` bude riadok `total_energy = always`. Na základe jej splnenia volať výpočtové subrutiny a zapísať vypočítané dáta. Nové výpočtové subrutiny sa nachádzajú v `src/io/diagnostics.F90`. Jedna má argument identifikačného čísla typu častice a vypočíta celkovú kinetickú energiu pre daný typ častice. Druhá jednoducho vypočíta energie elektrického a magnetického pol'a. Oboje nové subrutiny vznikli rozdelením originálnej subrutiny pre prípad keď `total_energy_sum = always` a pár úpravami. Na záver bolo ešte nutné upraviť súbor `src/shared_data.F90`, aby EPOCH vedel, že takáto diagnostika existuje. Našťastie, originálny EPOCH už náhodou mal priradenú aj možnosť počítať `total_energy = always` aj keď pri jej zavolaní sa nič nestalo. Preto nebolo nutné robiť úpravy v `src/deck/deck_io_block.F90`, ktoré by inak nutné boli.

2.2 Spracovanie výstupu

EPOCH generuje pri každom výstupovom cykle formát typu `.SDF` (Self Describing Format). Každý výpis sa zapisuje do zložky, v ktorej sa nachádza `input.deck`. Názov daného súboru je vždy štvorciferné číslo podľa poradia daného súboru pri výpise (t.j. `0000.sdf`, `0001.sdf` atď). V distribúcii EPOCHu je zahrnutých aj niekoľko skriptov a nástrojov na spracovanie týchto dát rôznym softvérom ako napr. `VisIt`, `IDL` a `Matlab`. V tejto práci bol využitý na analýzu iba `Matlab`. Pomocou skriptu `GetDataSDF.m`, ktorý sa nachádza v distribúcii EPOCHu v zložke `SDF/Matlab`, sa dajú jednoducho nahrať do ľubovoľného pol'a v `Matlab`e všetky dáta z určeného `.sdf` súboru.

Následne je nutné tieto dáta spracovať. Problémom je nutnosť postupného spracovania viacerých súborov v jednej zložke a následný prechod na ďalšiu zložku a tak ďalej. Na tieto účely bol vytvorený jednoduchý `Matlab`ovský skript, ktorého jadro vyzerá nasledovne:

```
1 numvec = [8 12 16 17 18 19];  
2 path = '2D-8nm-24nm/10keV/ST/';
```

```

3 for i = 1:length(numvec)
4     strcat(path,int2str(numvec(i)),'nm')
5     [Ek, EEM, x] = GetENew(strcat(path,int2str(numvec(i)),'nm'),10);
6     E = Ek + EEM;
7     ... zapis ...
8 end
9
10 ... vykreslenie grafov ...

```

V ňom sa volá ešte ďalší skript GetENew, ktorý vráti 3 vektory obsahujúce čas a časový vývoj energií pre dané počiatočné nastavenia.

```

1 function [Ek,EEM,x] = GetENew(path , steps )
2     x = zeros ([1 , steps ]);
3     Ek = zeros ([1 , steps ]);
4     EEM = zeros ([1 , steps ]);
5     for j = 0:steps
6         data = GetDataSDF( strcat ( path , '/' , sprintf ('%04d' , j) , '. sdf '));
7         x(j+1) = data.time;
8         EEM(j+1) = data.Total_Electric_Field_Energy_J + data.Total_Mag ...
9         Ek(j+1) = data.Total_Particle_Energy_J.electrons + data.Total_P ...
10    end
11 end

```

2.3 Metacentrum VO

Keďže ide o relatívne náročné výpočty, nebolo by vhodné ich uskutočňovať na osobnom počítači. Na tento účel sa hodí organizácia Metacentrum VO. Táto virtuálna organizácia je otvorená pre všetkých akademických pracovníkov, zamestnancov a študentov vedeckovýskumných inštitúcií v Českej republike. Svojim registrovaným užívateľom umožňuje bezplatne využiť výpočtovú techniku, ktorú poskytujú jednotlivé inštitúcie, zapojené v tejto organizácii. Zabezpečuje pritom jej prevádzku, rozvoj, užívateľskú podporu a bezpečnosť celého systému [23]. Metacentrum má k dispozícii vyše 10 000 CPU jadier a viac ako 1000 užívateľov. Užívateľom je k dispozícii väčšina výpočtovej techniky bez obmedzení, výnimkou je prioritizácia pri priradení výpočtovej techniky inštitúciám, ktoré ju priamo do spoločného systému poskytujú.

V rámci tejto práce sa využívali stroje z rôznych clusterov. Spomedzi nich najmä doom v Ostrave s 8-jadrovými procesormi Intel Xeon E5-2650v2 2.60 GHz, poskytnutý CESNETom, luna s rovnakými procesormi, poskytnutá FZÚ AV ČR v Prahe, losgar s 12-jadrovými AMD Opteron 6234 2.7 GHz, poskytnutý Loschmidt Laboratories na Masarykovej Univerzite, ramdal s 8-jadrovými Intel Xeon E5-4650 2,7GHz a gram s 8-jadrovými Intel Xeon E5-2670 2,6GHz, oboje poskytnuté ZČU v Plzni.

Okrem výpočtových kapacít poskytuje metacentrum aj úložné priestory na viacerých miestach v Českej republike. Okrem bežných úložísk sú k dispozícii aj archívne úložiská, z ktorých sa všetky dáta pravidelne zálohujú.

2.3.1 Spúšťanie úloh

Stroje metacentra bežia na UNIXe. Pre prístup z Windowsu je možné použiť napríklad program Putty. Pre prácu so súbormi je vhodné využiť program pracujúci s protokolom SCP, v rámci tejto práce bol využitý program na prácu so súbormi Servant Salamander.

Na spúšťanie úloh sa využíva plánovací systém Torque. V rámci neho je dovolené na prípravu úloh využívať len frontendové uzly. Najprv je nutné prihlásiť sa cez Putty na jeden z nich, napríklad na `skirit.ics.muni.cz` na Masarykovej Univerzite v Brne. Potom je možné robiť základné operácie najmä so súbormi a cez príkaz `qsub` nadefinovať úlohu a zaradiť ju do fronty. Úloha potom čaká vo fronte a spustí sa keď sa uvoľnia vhodné zdroje. Pri spúšťaní úloh sa aplikuje fairshare mechanizmus, ktorý uprednostňuje užívateľov, ktorí dovtedy využili menej zdrojov a vyprodukovali viac publikácií s pod'akovaním metacentru. Najjednoduchšie je spúšťať úlohu pomocou shellového skriptu, ktorý sa spustí jednoducho ako `qsub` súbor `.sh`. Tu je jednoduchý príklad takého súboru:

```
1 #!/bin/bash
2 #PBS -N EPOCH2Dmulti8nm
3 #PBS -l walltime=1d
4 #PBS -l nodes=1:ppn=16
5 #PBS -l mem=20gb
6 #PBS -l scratch=6gb
7 #PBS -j oe
8 #PBS -m e
9 trap 'clean_scratch' TERM EXIT
10
11 module add openmpi-1.6.5-intel
12
13 if [ ! -d "$SCRATCHDIR" ] ; then
14     echo "Adresar pro SCRATCH neni vytvoren!" 1>&2; exit 1; fi
15
16 cp -r "/storage/brno2/home/kocurvik/EPOCH/epoch-EbP/" $SCRATCHDIR/
17 cd $SCRATCHDIR
18 cd epoch-EbP/epoch2d
19
20 make clean
21 make COMPILER=intel
22 echo Data/10keV/ST | mpirun -np 16 bin/epoch2d
23
24 cp -r Data "/storage/brno2/home/kocurvik/EPOCH/epoch-EbP/epoch2d/" ||\
25     export CLEAN_SCRATCH=false
```

V tomto súbore sa potom na prvých odkomentovaných riadkoch, aké zdroje sú požadované a aký je maximálny predpokladaný čas výpočtu. Ďalej sa nainicializuje MPI modul, aby bolo možné vôbec EPOCH kompilovať. Neskôr sa prekopíruje EPOCH z úložného priestoru do SCRATCHDIR, čo je zložka, v ktorej sa bude vykonávať práca. Potom sa kompiluje samotný EPOCH pre jednu časticovú interpoláciu a finálne sa spustí s daným nastavením. V tejto práci sa bežne spúšťalo viacero nastavení, vrátane rôznych interpolácií za sebou naraz v jednom súbore.

V tomto príklade sa ale kvôli stručnosti spúšťa iba jedna simulácia. Nakoniec je nutné napočítané výsledky skopírovať na primárne úložisko. V prípade chyby sa skript z najväčšou pravdepodobnosťou ukončí predčasne, čo spôsobí, že sa ani nebudú zbytočne využívať zdroje.

V rámci tejto práce bol prepočítaný približne jeden CPU mesiac. Vďaka relatívne dobrému fairshare ratingu sa úlohy nezdržiavali vo fronte dlhšie ako pár hodín. Vo všeobecnosti po základnom zoznámení sa s úlohovým systémom je jednoduché s ním pracovať. Jediný problém môže nastať v prípade, že by úloha trvala dlhšie, ako je uvedené za parametrom `walltime`. V takom prípade sa úloha násilne ukončí a nevykoná sa presun dát. Našťastie k takejto situácii v rámci výpočtov pre túto prácu nedošlo. Je ale nutné kvôli tomuto nadsadzovať očakávané časy, aby sa nestalo, že sa úloha priradí pomalšiemu stroju a samotný výpočet potom bude zbytočný.

Pri dlhších simuláciach, kde je tento problém významnejší, je možné využiť reštartovateľnosť EPOCHu a to tak, že simulácia sa zastaví po kratšom čase ako je požadované na celý výpočet. Možnosť reštartu následnej simulácie zabezpečí reštartovateľný výstupový súbor, ktorý obsahuje všetky dáta, nutné pre pokračovanie simulácie tak, ako keby sa výpočet neprerušil. Výhoda tohto prístupu je, že je možné priebežne kontrolovať priebeh simulácie a zistiť či napríklad nedošlo k drastickému ohrevu. Takisto umožní predvídať na základe časovej náročnosti už spočítenej časti simulácie to, koľko bude kompletná simulácia trvať. Ďalšia možnosť je aj zistenie, že skúmaný jav už prebehol a šetriť tak výpočtový čas.

Kapitola 3

Simulácie a výsledky

V tejto časti práce budú predstavené výsledky realizovaných simulácií. Následne budú na ich základe prediskutované vlastnosti kódu EPOCH, ktoré boli teoreticky popísané v prvej kapitole.

3.1 1D

Najjednoduchšie je nastavenie a spracovanie 1D simulácii. Takisto sú omnoho rýchlejšie ako 2D simulácie, preto sa hodia na úvodné simulácie, pri ktorých sa užívateľ zoznamuje s fungovaním simulačného kódu. V tejto sekcii sa kvôli nízkej časovej náročnosti ani neoplatí sledovať reálne časy výpočtov. Tými sa zaoberá až sekcia s 2D výpočtami. Jednoduchosť simulácie taktiež umožňuje lepšiu analýzu niektorých javov.

3.1.1 Parametre simulácie

Keďže cieľom týchto simulácií je určiť mieru ohrevu v systéme, je vhodné zvoliť čo možno najjednoduchšie nastavenie simulácie. Preto je počiatočné rozloženie častíc jednoduché. V páse širokom 200 nm sa nachádzajú elektróny a protóny s hustotou $7 \cdot 10^{23} \text{ cm}^{-3}$. V simuláciách sú využívané dve rôzne počiatočné teploty častíc a to 10 keV a 500 eV, alebo v Kelvinoch $1.16 \cdot 10^8$ K a $5.8 \cdot 10^6$ K. Pre jednoduchosť sú v simulácii nastavené periodické okrajové podmienky na oboch okrajoch. Simulačná oblasť je 8 μm široká a plazma sa nachádza presne v jej strede. Takto zvolené počiatočné rozloženie a hustota častíc zodpovedajú ionizovanej ultratenkej fólii, tvorenej vodíkovou plazmou s hustotou ionizovaného pevného terča, čo je prípad relevantný pre interakciu laserového impulzu s pevnými terčami.

Pre skúmanie rôznych vlastností tejto simulácie je vhodné zistiť fyzikálne veličiny popísané v prvej kapitole. Pri danej hustote pre elektrónovú frekvenciu platí $\omega_{pe} \approx 4.72 \cdot 10^{16}$. Pre Debyovú dĺžku pri danej hustote platí pre teplotu 10 keV $\lambda_{De} \approx 0.88$ nm a pre teplotu 500 eV $\lambda_{De} \approx 0.2$ nm.

Čo sa týka numerických vlastností, tak počet častíc na bunku je vo všetkých simuláciách rovnaký a to 800 častíc na bunku. Toto samozrejme spôsobuje, že s počtom buniek narastá aj počet simulovaných častíc. Počet buniek a s ním súvisiaca veľkosť bunky je o niečo komplikovanejší.

Keďže zámerom je meniť veľkosť bunky v rozmedzí od 2 do 20 nm, je nutné zistiť koľko buniek je potrebných na pokrytie 8 μm oblasti. Bohužiaľ to pri niektorých veľkostiach dáva neceločíselný počet buniek. Preto je v simuláciach celočíselné zaokrúhlenie, teda využije sa vzorec:

$$N_b = \lceil X_{sim}/\Delta x \rceil, \quad (3.1)$$

kde N_b je počet buniek, X_{sim} je veľkosť simulačnej oblasti, v tomto prípade 8 μm a Δx je požadovaná veľkosť bunky. Jediná výnimka je prípad $\Delta x = 19$ nm, kde sa použije zaokrúhlenie nadol. Keďže dochádza k zaokrúhleniu, je jasné, že sa v niektorých prípadoch bude líšiť skutočná a požadovaná veľkosť bunky. Tieto rozdiely sú popísané pomocou nasledujúcich tabuliek:

N_b	4000	2000	1334	1143	1000	889	800	728	667
Δx (nm)	2	4	6	7	8	9	10	11	12
Δx_r (nm)	2	4	5.997	6.999	8	8.998	10	10.898	11.994

N_b	616	572	534	500	471	445	421	400
Δx (nm)	13	14	15	16	17	18	19	20
Δx_r (nm)	12.987	13.986	14.981	16	16.985	17.977	19.002	20

Tabuľka 3.1: Porovnanie požadovaných a reálnych veľkostí simulovaných buniek.

V nich možno vidieť, že skutočné (Δx_r) a požadované (Δx) veľkosti buniek sa líšia iba minimálne a ich rozdiel by mal mať zanedbateľný vplyv na simulácie. Preto v nasledujúcom texte a grafoch budú spomínané iba tie hodnoty veľkosti buniek, ktoré boli požadované, aj keď v simulácii mohla byť reálne daná bunka o niečo menšia.

Pri rozdielnych veľkostiach buniek sa stáva, že počiatočné energie rovnakého systému môžu byť rôzne. EPOCH sa snaží počiatočné rýchlosti častíc nastaviť tak, aby mali Maxwellovské rozdelenie. V dôsledku relatívne malého pomeru makročastíc a reálnych častíc nebude rozdelenie úplne Maxwellovské, ale bude sa k nemu blížiť. Keďže pri rôznej veľkosti buniek je celkový počet makročastíc v simulácii rôzny, tak sa budú líšiť aj počiatočné rýchlostné rozdelenia, čo má za následok aj malé rozdiely v celkovej počiatočnej energii systému. Taktiež sú pre 1D verziu v EPOCHu považované zvyšné dva rozmery bunky za 1 meter, takže sa bude celková energia líšiť aj od tej v 2D simuláciach, keďže v tomto prípade bude rozdielny počet reálnych častíc. Našťastie sa vo výpočte ohrevu koeficientu (1.41) používa pomer podľa celkovej počiatočnej energie systému pre danú simuláciu a aj v iných výpočtoch pôjde skôr o porovnanie energie vzhľadom na tú počiatočnú, takže sa tento problém nijako nepríjemne neprejaví.

Dĺžka simulácie bola pre všetky počiatočné nastavenia 100 fs a výstup sa vypisoval každých 10 fs. Pre všetky hodnoty z horeuvedených tabuliek boli prepočítané energie podľa upraveného výstupu, ako popisuje druhá kapitola. Pre veľkosti buniek 4, 6, 8, 12, 16 a 20 nm boli prepočítané na výstupe aj jednotlivé hodnoty hybností častíc, čo umožňuje analýzu hybnostnej distribučnej funkcie.

3.1.2 Teoretický predpoklad stability

Najzásadnejšia je otázka celkovej stability simulácie. V prvej kapitole boli odvodené niektoré podmienky tejto stability. Na základe podmienky pre plazmovú frekvenciu (1.32) a vzťahu medzi časovým a priestorovým krokom (1.26) je možné vyjadriť veľkosť bunky, od ktorej dôjde k nesplneniu tejto podmienky, a to nasledujúcim výrazom:

$$\Delta x_{\omega} = \frac{2 \cdot c}{0.95 \cdot \omega_{pe}} \quad (3.2)$$

Pre danú hustotu dáva tento vzťah hraničnú veľkosť bunky ako 13.38 nm. Teda pri bunke s rozmerom 14nm by podľa tejto podmienky mala byť táto simulácia nestabilná.

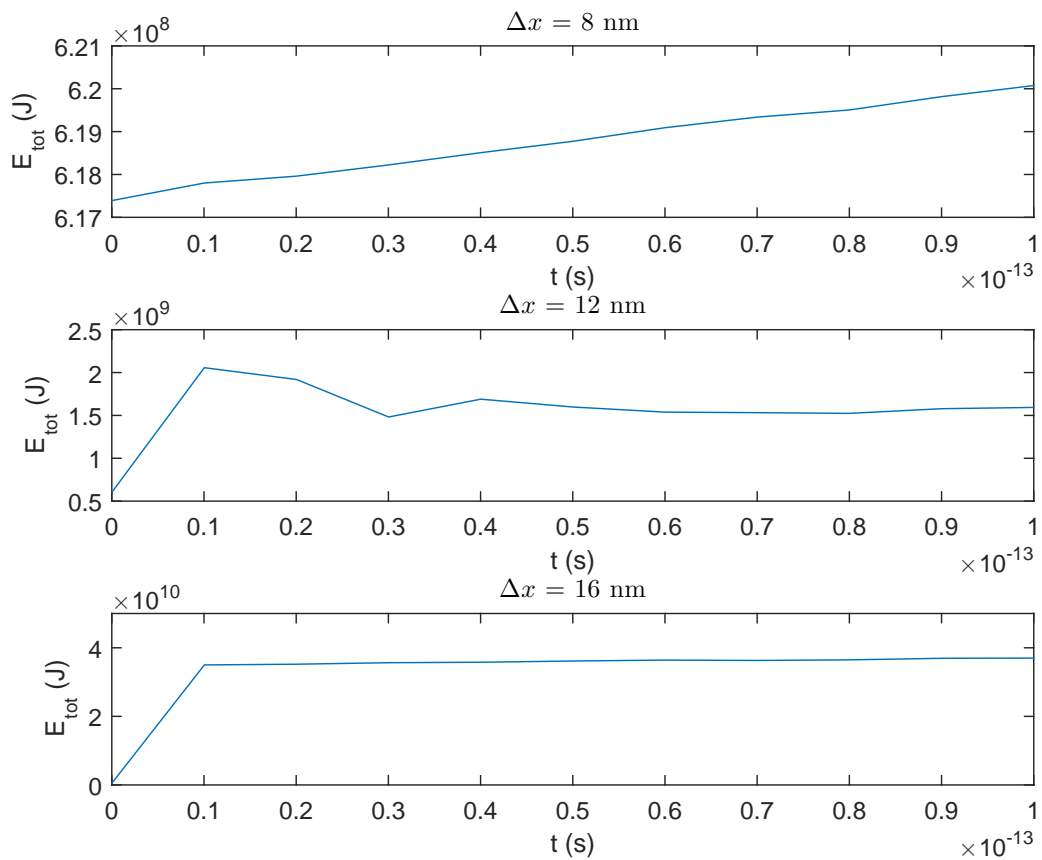
Ďalšia je podmienka vzhľadom na Debyovú dĺžku. Táto podmienka je určená vzťahom (1.40) a vyskytuje sa v nej neznáma konštanta C . Keďže existuje predpoklad, že sa plazma rapídne ohrieva dovedy, kým v tejto podmienke nenastane rovnosť, tak sa dá konštanta zhora obmedziť aj pomocou skrachovanej simulácie, v ktorej je ale nutné overiť, že skrachovala kvôli tejto podmienke a nie tej frekvenčnej.

3.1.3 Nestabilný ohrev

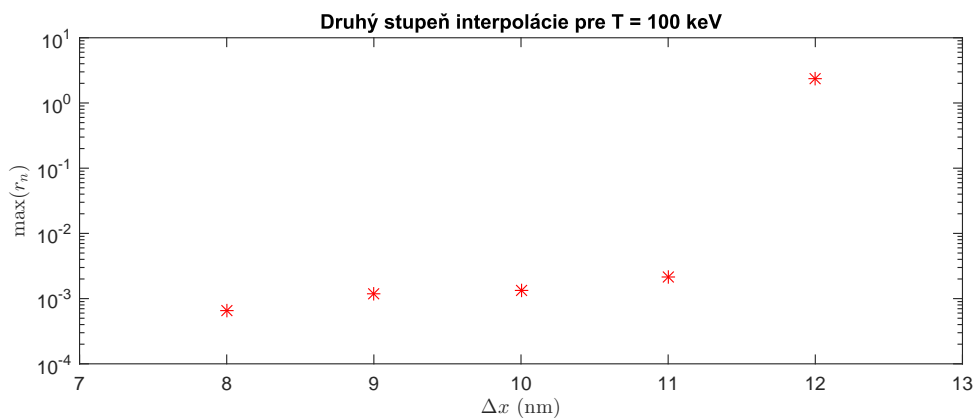
Táto kapitola sa bude zaoberať prípadmi, keď je ohrev taký výrazný, že sa simulácia dá považovať za nepoužiteľnú. V rámci tejto práce sa považuje ohrev za nestabilný ak najväčší koeficient ohrevu (1.41) pre časový rozdiel 10 fs bude väčší ako 0.1, teda za 10 femtosekúnd sa celý systém ohreje viac ako o desatinu celkovej počítateľnej energie systému. Táto definícia nestabilného ohrevu sa môže zdať ako príliš tolerantná voči ohrevu. Ale ako bude zrejmé z výsledkov, takáto definícia skrachovanej simulácie by ani nebola nutná, keďže rozdiel medzi stabilným ohrevom a tým nestabilným je výrazný a dá sa teda relatívne presne určiť veľkosť bunky pri ktorej došlo k nestabilite. Na druhej strane je vhodné stanoviť, kedy je simulácia považovaná za nestabilnú, aby nedošlo k prípadnému zmätku.

Na obr. 3.1 je možné vidieť časový vývoj celkovej energie systému pre tri rôzne veľkosti buniek pre teplotu 10 keV a interpoláciu druhého stupňa. Ako je zreteľné, tak v prípade malej bunky s rozmermi 8nm dochádza iba k miernemu ohrevu. V prípade 12nm ide o veľkosť bunky, pri ktorej simulácia začne byť nestabilná. Po ohreve na štvornásobok energie hneď na začiatku simulácie nastáva ustálenie energie. Vo väčšej bunke, a to 16 nm, už dochádza k masívnemu ohrevu a takisto k ustáleniu celkovej energie. Obrázok 3.2 ilustruje tento masívny rozdiel medzi stabilným a nestabilným ohrevom.

Z tabuľky 3.2 je vidno, že prvý stupeň interpolácie je v porovnaní s vyššími značne obmedzujúci na veľkosť bunky. V tomto prípade ale treba podotknúť, že ohrev pre prvý stupeň interpolácie je už pre teplotu 500 eV podľa definície v tejto sekcii nestabilný už pri 4 nm, avšak pri 7nm bunke tiež nastáva skok na ešte rapídnejší ohrev. Preto by bolo možné považovať za skrachovanie aj pre túto teplotu až pri 7nm bunke, aj keď reálne je simulácia nepoužiteľná aj pre menšie bunky.



Obr. 3.1: Vývoj celkovej energie v simulácií s $T = 10$ keV a druhým stupňom interpolácie pre rôzne veľkosti buniek.



Obr. 3.2: Demonštrácia rozdielu medzi maximálnymi hodnotami koeficientu ohrevu pre rôzne veľkosti buniek v simulácii pomocou druhého stupňa interpolácie s teplotou $T = 10$ keV

Pre prípad interpolácie štvrtého stupňa došlo ku krachu pre veľkosť bunky, kde nie je splnená frekvenčná podmienka (1.32). Pre druhý stupeň interpolácie, došlo ku krachu pre rovnakú veľ-

Stupeň interpolácie	1.	2.	4.
T = 500 eV	4	12	14
T = 10 keV	7	12	14

Tabuľka 3.2: Veľkosti buniek pre ktoré simulácia začala byť nestabilná pre rôzne interpolácie a teploty

kosť bunky, čo sa nezhoduje s podmienkou (1.40), takisto pomer medzi krachmi pre prvý stupeň interpolácie nasvedčuje, že podmienka pre krach nebude lineárne závisieť na Debyovej dĺžke.

Keďže nestabilná simulácia sa ohrieva iba do určitej teploty. Existuje predpoklad, že táto teplota bude tá, pre ktorú bude simulácia spĺňať podmienku (1.40), alebo ako sa javí nejaký jej ekvivalent. Zo skrachovaných simulácií je možné určiť linearný vzťah (3.3), ktorý určuje, do akej maximálnej teploty sa skrachovaná simulácia dostane. Vo vzťahu (3.3) vystupuje podiel maximálnej celkovej energie dosiahnutej v rámci simulácie a energie na začiatku simulácie. Pre prvý stupeň interpolácie sa ale pri určení tohto vzťahu využili iba simulácie s veľkosťami buniek do 15 nm, vrátane, keďže od tejto veľkosti bunky sa trend prejavuje nelineárne, čo môže byť spôsobené preletením častíc k okrajom simulačnej oblasti.

$$\frac{E_{max}}{E_0} \approx K \cdot \Delta x + L \quad (3.3)$$

Stupeň interpolácie	1.	2.	4.
T = 500 eV	$5.5 \cdot 10^{11}, -4056$	$3.3 \cdot 10^{11}, -2845$	$4.3 \cdot 10^{11}, -5325$
T = 10 keV	$2.7 \cdot 10^{10}, -188$	$1.4 \cdot 10^{10}, -218$	$1.2 \cdot 10^{10}, -213$

Tabuľka 3.3: Namerané hodnoty K v m^{-1} a L zo vzorca (3.3)

Tabuľka 3.3 znázorňuje hodnoty pre vzorec (3.3), z neho sa dá potom odvodiť istý ekvivalent podmienky (1.40) s rovnosťou (3.4) z ktorej vyplýva vzťah pre C a D (3.5), v ktorom λ_{De0} je Debyová dĺžka pre počiatočné podmienky simulácie.

$$\Delta x = C \cdot \lambda_{De}^2 + D = \frac{\lambda_{De}^2}{K \cdot \lambda_{De0}^2} + \frac{-L}{K} \quad (3.4)$$

$$C = \frac{1}{K \cdot \lambda_{De0}^2}; D = \frac{-L}{K} \quad (3.5)$$

Je zjavné, že rozdiel nastal medzi teoretickou predpoveďou (1.40), ktorá je lineárna a vypočítanými hodnotami, ktoré sú kvadratické. Žiaľ, aj konštanta D očividne závisí od Debyovej dĺžky. Z teoretických predpokladov sa dalo očakávať, že jediné výsledky, ktoré by mohli naskytnúť podmienku pre krach simulácie, sú tie pre prvý stupeň interpolácie, kde aj vyšlo, že konštanta D sa zhoduje približne s veľkosťou bunky, pre ktorú simulácia skrachuje. Žiaľ, ani pri tejto konštante

Stupeň interpolácie	1.	2.	4.
T = 500 eV	$4.5 \cdot 10^7, 7.3 \cdot 10^{-9}$	$7.5 \cdot 10^7, 8.6 \cdot 10^{-9}$	$5.8 \cdot 10^7, 1.2 \cdot 10^{-8}$
T = 10 keV	$4.7 \cdot 10^7, 6.9 \cdot 10^{-9}$	$9.2 \cdot 10^7, 1.5 \cdot 10^{-8}$	$1 \cdot 10^8, 1.7 \cdot 10^{-8}$

Tabuľka 3.4: Hodnoty C v m^{-1} a D v m zo vzorca (3.5)

nie je známe, ako závisí od parametrov simulácie. Preto sa dá usúdiť, že zo skrachovaných simulácií je v prípade prvého stupňa interpolácie možné určiť podmienku pre stabilitu simulácie. Bohužiaľ nie je zjavné, od čoho konštanty D závisí a určenie tejto závislosti by vyžadovalo väčšie množstvo simulácií.

3.1.4 Stabilný ohrev

V prípade, že simulácia ostala stabilná, dôjde aj tak k ohrevu, avšak nie tak drastickému. Tento ohrev je znázornený pomocou koeficientu ohrevu (1.41). Na obrázku 3.3 je znázornený vývoj strednej, maximálnej a minimálnej hodnoty koeficientu ohrevu.

Tento vývoj sa dal približne očakávať až na situáciu pre interpoláciu druhého stupňa a teplotu 500 eV. V tomto prípade došlo pre veľkosti bunky v okolí 6 nm k neznámemu efektu, ktorý je mimo záberu tejto práce. Okrem toho je zaujímavý aj fakt, že pre interpoláciu štvrtého stupňa nastáva aj ochladzovanie systému. Vďaka tomu ostáva priemerná hodnota koeficientu ohrevu relatívne malá, aj keď jej maximálna hodnota relatívne rýchlo rastie. Jednoznačná je aj výhoda využitia vyššieho stupňa interpolácie. Zlepšenie interpolácie z prvého na druhý stupeň prináša rádoovo nižší koeficient ohrevu.

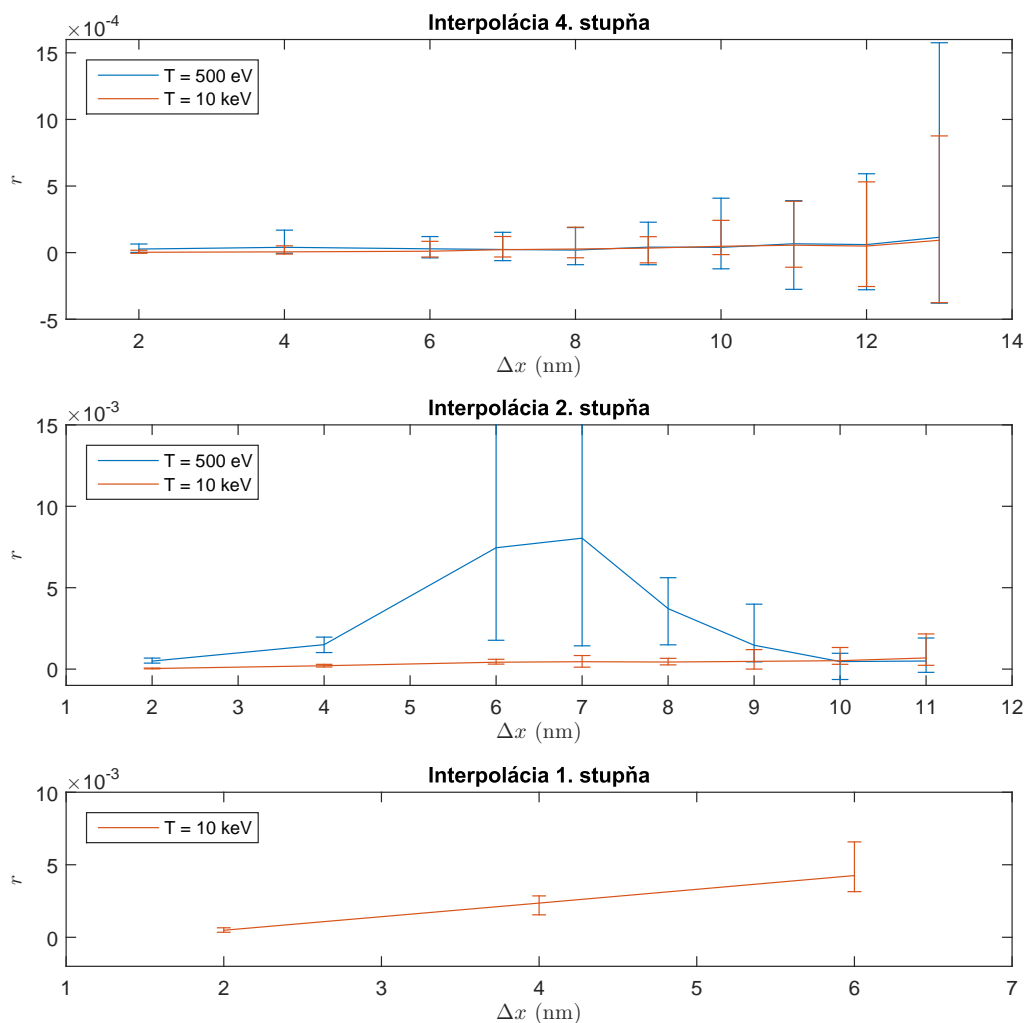
Z týchto dát o ohreve je možné odhadnúť závislosť koeficientu ohrevu na veľkosti bunky. Z grafov 3.3 je možné všimnúť si približne lineárny trend pre priemernú hodnotu a kvadratický nárast pre maximálnu hodnotu. Keďže maximálny ohrev skoro vždy nastáva hneď na začiatku simulácie, je vhodné posudzovať ohrev po prvých 10 fs a potom priemerný ohrev od 30 fs do 100 fs. Pre priemerný ohrev je preto možné využiť vzťah (3.6).

$$\langle r \rangle = A \cdot \Delta x + B \quad (3.6)$$

Stupeň interpolácie	1.	2.	4.
T = 500 eV	$2.6 \cdot 10^7, -4 \cdot 10^{-2}$	-	$-2.2 \cdot 10^3, 3.1 \cdot 10^{-5}$
T = 10 keV	$8.7 \cdot 10^5, -1.2 \cdot 10^{-3}$	$5 \cdot 10^4, -3.1 \cdot 10^{-6}$	$-1.9 \cdot 10^2, 9.1 \cdot 10^{-6}$

Tabuľka 3.5: Hodnoty A v m^{-1} a B zo vzťahu (3.6)

Podobne je možné pre koeficient ohrevu po prvých 10 fs určiť tento krát kvadratickým vzťahom (3.7).



Obr. 3.3: Vývoj koeficientu ohrevu pre rôzne teploty a interpolácie. Horný bod errorbaru označuje $\max(r_n)$, spodný $\min(r_n)$ a čiara znázorňuje $\langle r \rangle$.

$$\langle r \rangle = K \cdot \Delta x^2 + M \quad (3.7)$$

Tieto hodnoty boli určené fitovaním funkciou daného tvaru v matlabe pomocou metódy najmenších štvorcov. Keďže išlo skôr o približné porovnanie trendu, stačilo, že vo všetkých prípadoch bol koeficient determinácie $R^2 > 0.9$ [24].

Z hodnôt, ako aj z grafov sa javí, že ohrev pre vyššie stupne interpolácie nieje jednoznačne závislý od Debyovej dĺžky. Pre prvý stupeň interpolácie je zjavné, že čím väčšia Debyova dĺžka, tým slabší ohrev, čo aj zodpovedá teoretickej predpovedi, avšak na bližšie overenie závislosti by opäť bolo potrebných viacero simulácií.

Stupeň interpolácie	1.	2.	4.
T = 500 eV	$7.3 \cdot 10^{15}, 4 \cdot 10^{-3}$	-	$3.4 \cdot 10^{12}, 2.7 \cdot 10^{-5}$
T = 10 keV	$1.8 \cdot 10^{14}, -9.4 \cdot 10^{-5}$	$1.5 \cdot 10^{13}, -2.9 \cdot 10^{-5}$	$4.5 \cdot 10^{12}, -9 \cdot 10^{-5}$

Tabuľka 3.6: Hodnoty K v m^{-2} a M zo vzťahu (3.7)

3.1.5 Hybnosti častíc

To, že celková energia systému narastá, nemusí znamenať nutne ohrev, teda nárast termodynamickú teploty, hlavne ak ide o numerický jav. Pre overenie, či daný nárast energie je možné považovať za ohrev, je nutné nazrieť na hybnostnú distribúciu častíc v simulácii. V prípade ak by sa tvar hybnostnej distribučnej funkcie počas behu simulácie líšil od toho Maxwellovského nebolo by možné hovoriť o ohreve.

Z histogramov (3.4) je možné vidieť, že hybnostné rozdelenie zostáva v simulácii približne Maxwellovské pre všetky smery. Jediné čo sa zmení je samozrejme šírka distribučnej funkcie, čo znamená ohrev. Zaujímavé je, že aj pri nestabilnom ohreve sa tvar distribučnej funkcie zachováva. Preto aj samotný pojem nestabilný ohrev dáva dobrý zmysel.

3.2 2D

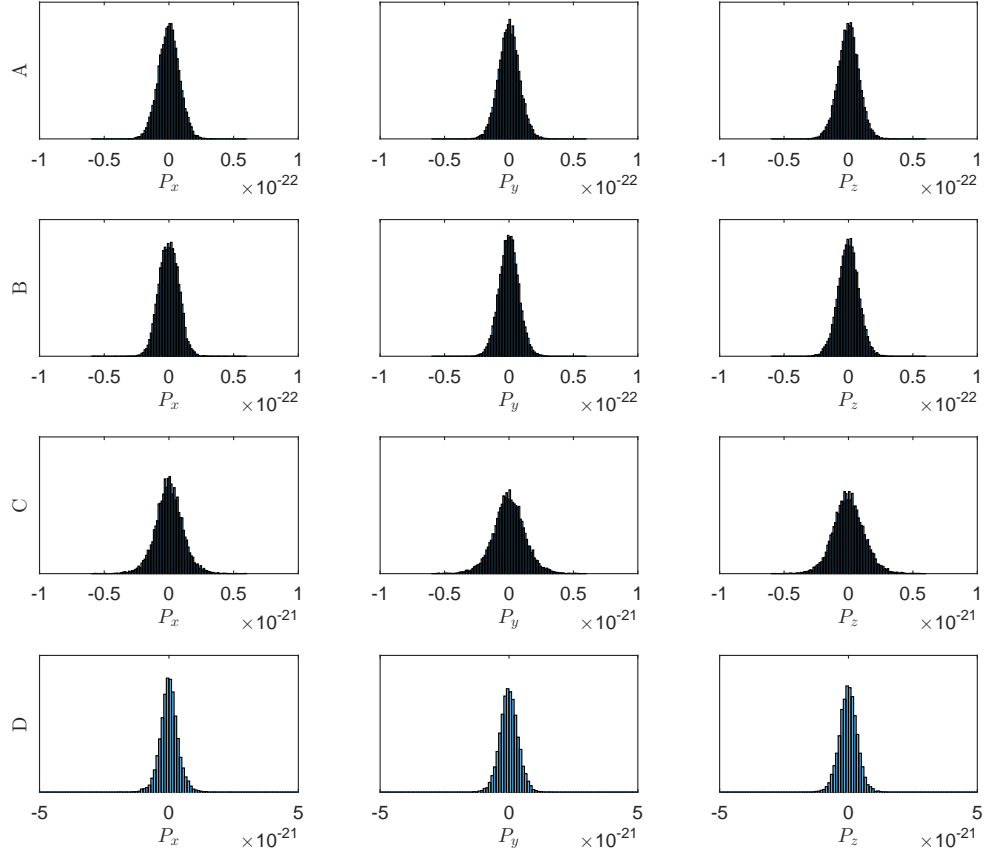
Po relatívne krátkych 1D simuláciách je vhodné prejsť na častejšie využívané 2D simulácie. Tieto simulácie zaberajú podstatne viac času. Preto je vhodné porovnávať taktiež časy pri simuláciách.

3.2.1 Testovacie parametre

Testovacie parametre su veľmi podobné tým pre 1D simulácie. Simulovaný priestor má veľkosť $8 \mu m \times 8 \mu m$. V ňom sa nachádza opäť ionizovaná ultratenká fólia tvorená vodíkovou plazmou a teda elektróny a protóny s hustotou $7 \cdot 10^{23} \text{ cm}^{-3}$ v páse s rozmermi $200 \text{ nm} \times 4 \mu m$ umiestnenom v strede simulačnej oblasti.

Opäť sa bude simulovať v dvoch rôznych počiatkových teplotách 10 keV a 500 eV. Takéto počiatkové podmienky opäť vedú na elektrónovú frekvenciu $\omega_{pe} \approx 4.72 \cdot 10^{16}$, Debyovú dĺžku pre teplotu 10 keV $\lambda_{De} \approx 0.88 \text{ nm}$ a pre teplotu 500 eV $\lambda_{De} \approx 0.2 \text{ nm}$.

Simulácie opäť bežali 100 fs s výpismi po 10 fs. Vypisované boli len celkové kinetické energie pre každý typ častíc a energie polí. Tentokrát boli simulované len simulácie s bunkami o veľkostiach 8, 12, 16, 17, 18, 19 a 20 nm. Veľkosti buniek sú zhodné v oboch smeroch. Pričom opäť boli reálne veľkosti buniek podľa tabuľky 3.1. Takisto bol počet častíc na jednu bunku nastavený vo všetkých simuláciách na 800.



Obr. 3.4: Porovnanie hybnostných distribučných funkcií pre teplotu 10 keV, druhý stupeň interpolácie a rôzne veľkosti buniek. **A:** distribučná funkcia v čase $t = 0$ pre veľkosť bunky 8 nm (počiatočný stav). **B:** distribučná funkcia v čase $t = 100$ fs pre veľkosť bunky 8 nm (stabilný ohrev). **C:** distribučná funkcia v čase $t = 100$ fs pre veľkosť bunky 12 nm (nestabilný ohrev). **D:** distribučná funkcia v čase $t = 100$ fs pre veľkosť bunky 20 nm (nestabilný ohrev).

3.2.2 Teoretický predpoklad stability

Opäť je možné urobiť jednoduchý predpoklad pre stabilitu simulácie z frekvenčnej podmienky (1.32) a vzťahu pre nastavenie časového kroku v 2D prípade (1.26). Keďže časový krok sa nastavuje pre 2D simuláciu $\sqrt{2}$ krát kratší ako ten pre 1D simuláciu, preto aj očakávaná veľkosť bunky kde dôjde k tejto nestabilite bude väčšia a to podľa (3.8) $\Delta x = 18.92$. Malo by teda dôjsť ku krachu pre veľkosť bunky 19 nm.

$$\Delta x_{\omega} = \frac{\sqrt{2} \cdot 2 \cdot c}{0.95 \cdot \omega_{pe}} \quad (3.8)$$

Pre 1D prípad sa ukázalo, že teoretická podmienka na rozmer bunky podľa Debyovej dĺžky

(1.40) nesedí s vypočítanými výsledkami. Preto ani nie je možné urobiť dobrý odhad, kedy by mohlo dôjsť k nestabilite v jej dôsledku.

3.2.3 Nestabilný ohrev

Stupeň interpolácie	1.	2.	4.
T = 500 eV	8*	20	20
T = 10 keV	12*	20	20

Tabuľka 3.7: Veľkosti buniek pre ktoré 2D simulácia začala byť nestabilná pre rôzne interpolácie a teploty. * Keďže simulácie neboli vykonané pre takú hustotu veľkosti buniek ako v 1D prípade, sú tieto údaje len približné, ale ohrev pre veľkosti bunky je dostatočne malý ($\langle r \rangle \approx 0.1$), dá sa predpokladať, že s o trochu menšími bunkami by bola simulácia stabilná.

Ako ukazuje tabuľka 3.7, tak krach pre druhý a štvrtý stupeň interpolácie nastal pre 20 nm. Podľa predpokladu mal krach nastať pre 18.92 nm. Takže je celkom možné, že chyby pri veľkosti bunky 19 nm ešte nebudú úplne zničujúce. V takom prípade sedí predpoklad o krachu kvôli frekvenčnej podmienke.

Podľa teoretického predpokladu sa dá uvažovať o ekvivalente podmienky (1.40) len pre prvý stupeň interpolácie. Opäť je ale zjavné, že reálne táto podmienka nie je lineárna. Analýza skrachovanej simulácie sa v 2D prípade nedá využiť, keďže skrachované dáta pre teplotu 500 eV už nepredstavujú žiadny trend. To je ale spôsobené skôr tým, že sa meralo iba s bunkami 8,12 a od 16 nm, pričom na túto analýzu by boli najvhodnejšie simulácie s bunkami v rozmedzí od 9 do 14 nm.

3.2.4 Stabilný ohrev

V neskrachovaných simuláciách došlo aj v 2D prípade k ohrevu. Grafy 3.5 znázorňujú tento ohrev. Keďže pre neskrachované simulácie prvého stupňa existuje len málo dát tak graf by príliš nedával zmysel. Pre nich ale platí že ohrev je výraznejší ako pre interpolácie vyšších stupňov.

Takisto sa prejavil neznámy efekt pre druhý stupeň interpolácie a teplotu 500 eV. Opäť teda došlo k významejšiemu ohrevu pre menšie bunky. To naznačuje, že tento efekt je naozaj numerický a nastáva pre simulácie s približne rovnakou veľkosťou buniek v 1D aj 2D.

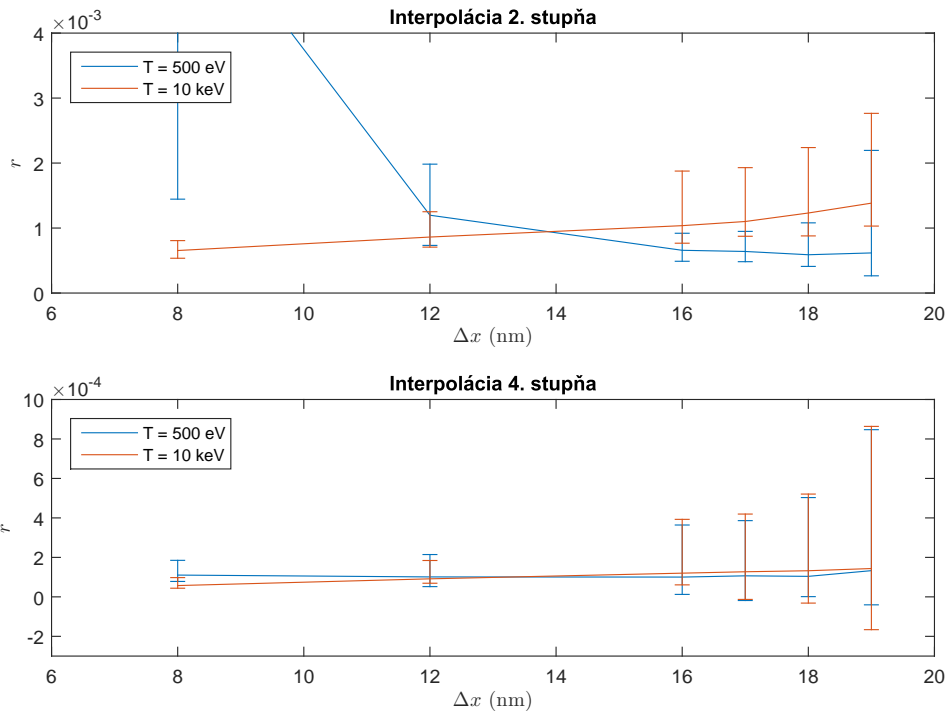
Znova je možné skúmať závislosti ohrevu od veľkosti bunky podľa vzt'ahov (3.6) a (3.7).

Stupeň interpolácie	2.	4.
T = 500 eV	-	$-4 \cdot 10^3, 1.4 \cdot 10^{-4}$
T = 10 keV	$5.1 \cdot 10^4, 1.8 \cdot 10^{-4}$	$3.6 \cdot 10^3, 9.4 \cdot 10^{-4}$

Tabuľka 3.8: Hodnoty A v m^{-1} a B zo vzt'ahu (3.6)

Stupeň interpolácie	2.	4.
T = 500 eV	-	$1.8 \cdot 10^{12}, -1.8 \cdot 10^{-5}$
T = 10 keV	$6 \cdot 10^{12}, 3.6 \cdot 10^{-4}$	$2.1 \cdot 10^{12}, -9.1 \cdot 10^{-4}$

Tabuľka 3.9: Hodnoty K v m^{-2} a M zo vzťahu (3.7)



Obr. 3.5: Vývoj koeficientu ohrevu pre rôzne teploty a interpolácie v 2D simuláciach. Horný bod errorbaru označuje $\max(r_n)$, spodný $\min(r_n)$ a čiara znázorňuje $\langle r \rangle$.

Opäť sa teda prejavili podobné vlastnosti pre ohrev ako v 1D. Teda, že ohrev pre vyššie stupne interpolácie výrazne nezávisí na Debyovej dĺžke. Ohrev pre prvý stupeň interpolácie sa skúmať nedal, pre nedostatok neschrachovaných simulácií.

3.2.5 Náročnosť výpočtu

Pri dlhších simuláciách je relevantné zaoberať sa aj výpočtovou náročnosťou simulácie. Na rozumné porovnanie časovej náročnosti je nutné aby boli porovnávané behy simulácií na tom istom stroji. To nieje problém keďže v rámci tejto práce sa simulácie spúšťali dávkovo. Navyše EPOCH vypisuje na výstupe okrem chybových hlásení aj celkový čas simulácie, čo sa presne hodí na účel porovnania časovej náročnosti.

Časy z tabuľky 3.10 boli namerané pre simulácie na strojoch z clustru doom. Tento cluster má na jeden uzol dva 8-jadrové procesory Intel Xeon E5-2650 v2 2.60GHz, 64 GB RAM k dispozícii

Teplota	St. inteprolácie	$\Delta x = 12 \text{ nm}$	$\Delta x = 16 \text{ nm}$	$\Delta x = 20 \text{ nm}$
10 keV	1.	16:34	8:34	4:55
500 eV	1.	14:42	8:54	4:18
10 keV	2.	18:56	7:09	4:28
500 eV	2.	17:10	7:01	4:39
10 keV	4.	26:54	11:00	5:43
500 eV	4.	24:09	9:48	5:47

Tabuľka 3.10: Porovnanie časov behov simulácií s rôznymi parametrami na 16 CPU z clusteru doom. Časy sú vo formáte minúty:sekundy.

a fyzicky sa nachádza v Ostrave. Pre tento výpočet bolo požadovaných 16 jadier na jednom uzle, čo presne zodpovedá konfigurácii na clustry doom.

Časová náročnosť týchto simulácií nieje prekvapujúca. Pomery medzi dĺžkami simulácií pre rozdielne veľkosti buniek zodpovedajú približne kubickému predĺženiu pre zmenšenie bunky. To vyplýva z kvadratického zväčšenia počtu buniek a tým aj počtu makročastíc a ešte jednonásobnému predĺženiu kvôli skráteniu času jedného časového kroku. Podobne je očakateľné predĺženie doby behu pre vyššie stupne interpolácie. Mierne prekvapujúca je skutočnosť, že rozdiel medzi prvým a druhým stupňom intepolácie je relatívne malý.

Rozdiely dôb behov pre rôzne interpolácie sú taktiež podľa očakávaní. Keďže sa pri interpoláciách vyšších stupňov musí pri interpoláciách z mriežky na časticu a naopak počítať hodnota viacero polí a taktiež vyššie mocniny, teda viac operácií, je logické, že bude doba behu väčšia. Mierne prekvapivý je fakt, že rozdiel medzi prvým stupňom interpolácie v dobe behu a druhým je relatívne malý vzhľadom na omnoho lepšie výsledky toho druhého. To však len naznačuje, že časť behu, ktorá sa zaoberala Maxwellovými rovnicami, ktorá je rovnaká pre všetky stupne interpolácie, bola relatívne významná v simulácii. Tieto rozdiely by mohli byť výraznejšie pre väčší počet častíc na bunku. Tento parameter bol však pre všetky simulácie v tejto práci rovnaký.

Časy pre tieto simulácie sa môžu zdať relatívne malé. Treba si ale uvedomiť, že išlo o relatívne krátku (100 fs) simuláciu, s veľmi jednoduchým výstupom. Často sú aj veľkosti buniek menšie z dôvodu schválneho použitia menšej bunky, pre ktorú sa predpokladá, že bude simulácia stabilná.

3.3 Diskusia

Podmienka na stabilitu pre veľkosť bunky podľa Debyovej dĺžky (1.40) sa ukázala ako nie presná. Isté obmedzenie pre veľkosť bunky existuje, ale nie je závislé od Debyovej dĺžky lineárne. Od tej závisí toto obmedzenie kvadraticky, ale pre malé Debyove dĺžky sa významne prejaví ďalší člen, ktorý od Debyovej dĺžky jednoduchým vzťahom nezávisí.

V súlade s teóriou predstavenou v sekcii 1.6.4 sa potvrdilo, že pre vyššie stupne inteprolácie je ekvivalent tejto podmienky voľnejší až zanedbateľný.

Vo výsledkoch sa ukázala dôležitosť frekvenčnej podmienky (1.32), ktorej dodržanie je kritické pre stabilný beh simulácie a v prípadoch pre vyššie stupne interpolácie sa ukázala ako smerodajná aj k tomu, kedy simulácia naozaj skrachuje.

V prípade stabilného ohrevu sa výsledky z tejto práce líšia od tých experimentálnych z článku [9]. Mierna zhoda nastala s teoretickým predpokladom z [2]. Ohrev v úvode simulácie totiž naozaj závisí na veľkosti bunky kvadraticky. Závislosť od Debyovej dĺžky sa prejavila iba pre prvý stupeň interpolácie a ani to nie presne podľa vzťahu (1.42), čo sa ale nedalo overiť len pri dvoch počiatkových teplotách, keďže tento vzťah je len približný a predpokladá sa pri jeho odvodení, že ohrev je spôsobený stochastickými chybami simulácie. Nezhoda však nastáva v tom, že neskôr sa ohrev ustáli a tento ustálený ohrev už závisí od veľkosti bunky len lineárne, pričom opäť pre vyššie interpolácie nezávisí od Debyovej dĺžky. Tieto výsledky ale možno vnímať pozitívne, keďže to znamená, že v modernejších kódoch s lepšími stupňami interpolácie dochádza k výrazne nižšiemu ohrevu ako v tých starších.

Časy behov sa zhodujú dobre s teóriou, ktorá pre 2D prípad vedie ku kubickému nárastu časovej náročnosti výpočtu so znižujúcou sa bunkou. A takisto umožnili porovnať výhodnosť jednotlivých stupňov interpolácie. Z neho je jasné, že prvý stupeň interpolácie sa neoplatí používať. Minimálna časová úspora vôbec nie je výhodná ak sa vezme do úvahy nutnosť používať menšie bunky na simuláciu, v ktorej aj tak nastane výraznejší ohrev. To je teoreticky odôvodniteľné silnou previazanosťou módov, odvodenou v sekcii 1.6.4, pri využití tejto interpolácie.

Využitelnosť štvrtého stupňa interpolácie je tiež otázna, keďže pri menších bunkách už dochádza k významnému predĺženiu doby behu. Preto je tento stupeň interpolácie vhodný len v situáciách, kde numerický ohrev výrazne ovplyvňuje možnosti skúmania nejakého konkrétneho fyzikálneho javu, keďže so stabilitou sú na tom druhý a štvrtý stupeň podobne.

Treba upozorniť, že v tejto práci sa bral do úvahy hlavne numerický ohrev a nie fyzikálna presnosť simulácie. Preto skutočnosť, že nedošlo k drastickému ohrevu nutne neznamena, že javy v simulácii s väčšími bunkami sú korektné. To môže byť predmetom ďalšieho výskumu.

Záver

V rámci tejto práce sa podarilo zhrnúť základy teórie metódy particle in cell. Ako po stránke priebehu samotného algoritmu, tak aj po stránke jeho numerických vlastností. Dôležitým poznatkom je teoretický princíp numerického ohrevu v týchto simuláciách.

Významnú časť praktickej práce tvorilo oboznámenie sa s kódom EPOCH nielen z užívateľského hľadiska, ale aj prístupom k samotnému kódu, jeho úpravou a komunikáciou v rámci komunity. S využívaním EPOCHu je úzko späté aj zoznámenie sa s gridovým počítaním náročných simulácií pomocou Metacentra VO, ktoré prinieslo cenné skúsenosti v oblasti informatickej fyziky, ako aj spracovanie veľkého množstva výsledkov v Matlabe a všeobecný dátový manažment výsledkov simulácií.

Prínosom je aj zistenie menších chýb v samotnom kóde ako aj nezrovnalostí medzi poslednou verziou manuálu a skutočnosťou. Tie sa týkajú nielen vnútorných záležitostí kódu, ale aj priamych inštrukcií pre užívateľa.

Samotné simulácie a výsledky, z nich vyplývajúce ukázali isté nezrovnalosti medzi teóriou a numerickými vlastnosťami kódu EPOCH. To je spôsobené tým, že literatúra sa venuje väčšinou starším kódom s nižšími stupňami interpolácie a inými algoritmami pre riešenie Maxwellových rovníc. Tieto zistenia sú veľmi užitočné, keďže napríklad podmienka na veľkosť výpočtovej bunky podľa Debyovej dĺžky významne obmedzuje veľkosť bunky, aj keď takéto obmedzenie vôbec nie je nutné pre stabilitu simulácie. Užívateľ kódu by tak mohol z obavy pred nestabilitou nastaviť zbytočne malú bunku, čo v súčinnosti s kubickým rastom výpočtovej náročnosti so zmenšením bunky v 2D simulácii vedie k zbytočne veľkej výpočtovej náročnosti a tým k neefektívnemu využívaniu dostupných zdrojov.

Celkovo je teda výhodné vedieť optimalizovať parametre simulácií, čo môže byť náročné, keďže ako už bolo spomenuté, k dispozícií je iba literatúra k numerickým vlastnostiam starších kódov, ktoré pracujú síce na podobných princípoch, ale aj menšie rozdiely v algoritmoch môžu viesť k veľmi rôznym numerickým vlastnostiam.

Užitočné je aj zistenie, že ako najvhodnejšie sa javí využívať druhý stupeň interpolácie, ktorý je dobrým kompromisom medzi rýchlosťou a stabilitou simulácie.

Zaujímavý bol aj objav zatiaľ bližšie neskúmaného efektu ktorý nastal pre druhý stupeň interpolácie a teplotu 500 eV. To umožňuje pokračovanie tejto práce. Ďalšie možné pokračovanie tejto práce je prechod na ešte náročnejšie 3D simulácie. Prípadne porovnávať numerickú stabi-

litu vzhľadom na meniaci sa počet častíc na bunku, čo je parameter, ktorý by teoreticky mal mať významný dopad na mieru ohrevu simulácie. Takisto by bolo možné porovnávať vybrané fyzikálne deje a ich priebeh v simulácii pri rôznych nastaveniach parametrov simulácie a overiť tak aj presnosť výsledkov získaných zo simulácie. To dokazuje, že možnosti skúmania numerických parametrov tejto metódy su veľmi široké a bolo by ťažké pokryť ich jednou prácou.

Každopádne je metóda particle in cell dôležitou súčasťou modernej fyziky laserovej plazmy, a preto je žiadúce zaoberať sa ňou podrobnejšie v tomto kontexte.

Literatúra

- [1] Langdon A. B, Birdsall C. K. *Plasma Physics via Computer Simulation*. Adam Hilger, 1999.
- [2] Eastwood J. W, Hockney R. W. *Computer Simulation Using Particles*. Taylor & Francis, 1998.
- [3] Chen F. *Úvod do fyziky plazmatu*. Academia, 1984.
- [4] Štoll I. *Elektřina a magnetismus*. FJFI, ČVUT, 2003.
- [5] Klimo O, Pšikal J. *Základy fyziky laserového plazmatu*. Prednáška, 2014.
- [6] Limpouch J. *Základy fyziky plazmatu*. Prednáška, 2015.
- [7] Kulhanek P. *Teorie plazmatu - studijní text pro FJFI ČVUT*. FJFI ČVUT, 2009-2015.
- [8] Klimo O. *Metody počítačové fyziky 2*. Prednáška, 2015.
- [9] Geddes C. G. R, Esarey E, Schroeder C. B, Leemans W. P, Cormier-Michel E, Shadwick B. A. Unphysical kinetic effects in particle-in-cell modeling of laser wakefield accelerators. *PHYSICAL REVIEW E*, 78(1):016404, 2008.
- [10] Ruhl H. *Classical Particle Simulations with the PSC code*. Ruhrstadt Universität Bochum, 2005.
- [11] Bennett K. *Developers Manual for the EPOCH PIC codes*. University of Warwick, 2013.
- [12] Buneman O, Villasenor J. Rigorous charge conservation for electromagnetic field solvers. *Computer Physics Communications*, 69:306–316, 1992.
- [13] Yee K. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on*, 14:302–307, 1966.
- [14] Limpouch J. *Numerické metody*. Prednáška, 2014.
- [15] Lewy H, Courant R, Friedrichs K. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11:215–234, 1967[1928].

- [16] set_plasma_frequency_dt - Maybe wrong condition (#383) | Issues | EPOCH / epoch | GitLab. <https://cfsa-pmw.warwick.ac.uk/EPOCH/epoch/issues/383>. Diskusia Vid: 3.6.2015.
- [17] Schroeder C. B, Shadwick B. A. Physical fidelity in particle-in-cell modeling of small debye-length plasmas. *Lawrence Berkeley National Laboratory*, 2009.
- [18] Dimits A. M, Dorland W, Shumaker D. E, Nevins W. M, Hammett G. W. Discrete particle noise in particle-in-cell simulations of plasma microturbulence. *Physics of Plasmas*, 12(12):122, 2005.
- [19] Matsumoto H, Okuzawa T, Ueda H, Omura Y. A study of the numerical heating in electrostatic particle simulations. *Computer Physics Communications*, 79(2):249–295, 1994.
- [20] Bennett K. *Users Manual for the EPOCH PIC codes*. University of Warwick, 2014.
- [21] CCPForge - EPOCH: Extendable PIC Open Collaboration. <http://ccpforge.cse.rl.ac.uk/gf/project/epoch/>. Vid: 29.4.2015.
- [22] Dashboard | GitLab. <https://cfsa-pmw.warwick.ac.uk/>. Vid: 3.6.2015.
- [23] O MetaCentru VO. <https://metavo.metacentrum.cz/cs/about/index.html>. Vid: 4.5.2015.
- [24] Enders F. B. Coefficient of determination. <http://www.britannica.com/science/coefficient-of-determination>. Vid: 20.6.2015.