

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta – Katedra fyziky

Název práce

Metody počítačového modelování

Vedoucí práce: RNDr. Petr Bartoš, Ph.D.

Autor: Petr Suchan

Anotace

V tomto textu jsou uvedeny rozborů základních technik, které se často používají při vytváření matematických modelů v počítačové fyzice. Text je rozdělen do čtyř základních kapitol. V první kapitole je uvedena definice počítačového modelování a počítačových simulací. Poté následuje kapitola zabývající se částicovými, spojitými a hybridními technikami počítačového modelování. Nakonec se tento text zabývá popisem dvou nepoužívanějších metod počítačového modelování, tj. metodou molekulární dynamiky a metodou Monte Carlo.

Annotation

This text includes analyses of fundamental techniques that are often used for creating mathematic models in the computer physics. The text is divided into four basic chapters. The first chapter brings the definition of computer models and of computer simulation. The following chapter deals with particle, linking and hybrid techniques of the computer modeling. In the end the text describes the most used methods of the computer simulation, i.e. the method of molecular dynamics and the Monte Carlo method.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

Datum 4.12. 2007

Petr Suchan
Podpis studenta

Obsah:

Úvod	6
1. Počítačové modelování a simulace	7
1.1 Počítačové modelování	7
1.2 Simulace	7
1.3 Model	8
1.4 Postup řešení fyz. úkolu pomocí počítačového modelování	9
2. Techniky počítačového modelování	10
2.1 Částicové techniky	10
2.2 Spojité techniky	11
2.3 Hybridní techniky	17
3. Metoda molekulární dynamiky	18
3.1 Popis metody	18
3.2 Idea řešení metodou molekulární dynamiky	18
3.3 Pracovní oblast	19
3.4 Řešení translace	21
3.4.1 Pohybové rovnice	23
3.4.2 Výpočet silového působení	23
3.5 Metody urychlující výpočet silového působení	25
3.5.1 Metoda P-I-C	25
3.5.2 Další optimalizované metody	30
4. Metoda Monte Carlo	31
4.1 Charakteristika metody	31
4.2 Idea řešení problému metodou Monte Carlo	31
4.3 Generování náhodných čísel	32
4.4 Transformace náhodných veličin	33
4.5 Efektivnost metody	39

4.6	Transportní problém	39
4.6.1	Modelování pomocí přirozených modelů	40
4.6.1	Modelování pomocí umělých modelů	44
4.7	Ukázky řešení numerických úloh	47
4.7.1	Výpočet čísla π	47
4.7.2	Výpočet jednorozměrného určitého integrálu	48
4.7.2	Výpočet dvojného určitého integrálu	50
	Závěr	51
	Použitá literatura	52

Úvod

Počítačové modelování se hojně využívá v některých technických vědách, například ve fyzice, kybernetice, automatizaci. S rozvojem výpočetní techniky však počítačové modelování proniklo do většiny technických i netechnických věd a stalo se tak nedílnou součástí např. i meteorologie, ekonomie, biologie, chemie, medicíny....

Největší praktickou výhodou modelování je možnost pomocí pokusů a omylů vyřešit i ty úlohy, které nemají analytické řešení. A nebo vyzkoušet vlastnosti drahých zařízení před tím, než budou realizována v praxi, (např. integrované obvody, různé biologické a chemické přípravky, ...). Počítačové modelování může např. v biologii zabránit některým pokusům na zvířatech.

Pro modelování v různých vědních oborech byla vyvinuta řada specializovaných softwarových balíků. Mezi nejznámější patří MATLAB (Matrix Laboratory) - modelování dynamických systémů. V minulosti se používaly pro modelování analogové počítače - výpočet obyčejných diferenciálních rovnic s počátečními podmínkami a nebo simulační programovací jazyky, například Simula (systémy s diskretními událostmi) nebo CSSL (pro spojité systémy) viz [1].

Podstatou počítačového modelování je náhrada zkoumaného systému jeho modelem (uměle vytvořeným objektem). S tím provádíme pokusy s cílem získat informace o původním zkoumaném systému. Modely zkoumaných systémů jsou obvykle soustavy diferenciálních rovnic.

Počítačové modelování je výsadním směrem počítačové fyziky (takové postupy, při kterých jsou fyzikální problémy řešeny s výrazným využitím výpočetní techniky). Rozděluje se na řadu dílčích technik (techniky částicové, spojité, hybridní) a je také známo mimo počítačovou fyziku.

Cílem této práce je vytvořit stručný a výstižný text, který bude popisovat základní techniky počítačového modelování, často využívané při vytváření počítačových modelů. Konkrétně zde bude uvedeno dělení těchto technik a jejich stručný popis a dále popis a využití dvou nejužívanějších, tj. metody molekulární dynamiky a metody Monte Carlo.

1. Počítačové modelování a simulace

1.1 Počítačové modelování

Počítačové modelování se běžně využívá v praxi pro řešení složitějších úloh. Pomocí počítačového modelování je možné zjistit téměř cokoli, např. jak se bude chovat určitý zkoumaný problém, chemické, biologické přípravky, vývoj počasí,

V technické praxi je velmi často nutné řešit nejrůznější úlohy, které ve většině případů není možno řešit analyticky (přesně). To proto, že takové řešení by bylo příliš složité, a proto není vhodné jej v praxi používat a nebo vůbec neexistuje. Proto se téměř všechny reálné problémy z oblasti vědy a techniky řeší pomocí počítačového modelování.

Stále se zvyšující výkonnost výpočetní techniky dovoluje řešit nejrůznější úlohy ve stále kratším čase a provádět obrovská množství pokusů s modely reálných úloh (simulací) s nejrůznějšími vstupními hodnotami.

Podstatou modelování ve smyslu výzkumné techniky je náhrada zkoumaného systému jeho modelem (přesněji: systémem, který jej modeluje). Cílem je získat pomocí pokusů s modelem informaci o původním zkoumaném systému viz [2].

1.2 Simulace

Simulace je v matematice, kybernetice a fyzice vědecká metoda, při které se zkoumají vlastnosti studovaného systému pomocí experimentů s jeho matematickým modelem viz [1].

Simulace na počítačích

Ve starších dobách byl simulátor (zařízení pro provádění experimentů s modelem) realizován na speciálních zařízeních a podle nich dostávaly příslušné simulace jména. Jednalo se např. o simulaci elektromechanickou, hydrodynamickou, analogovou (pomocí analogových počítačů). Dnes se výhradně využívají simulace na číslicovém počítači. Jedná se tedy o simulaci **číslicovou**.

Pokud je charakter simulovaného systému spojité, tj. jestliže se vlastnosti, které popisují systém nemění v čase skokově, mluví se o **simulaci spojité**. Jestliže je simulovaný systém diskrétní, tj. vlastnosti popisující systém se mění skokově, mluví se o **diskrétní simulaci**. Když je simulovaný systém kombinovaný, tj. má vlastnosti jak diskrétní tak i kombinované, mluví se o **kombinované diskrétně - spojité simulaci** nebo častěji o **kombinované simulaci**.

1.3 Model

Prvním krokem při počítačovém modelování bývá sestavení matematického modelu zkoumaného systému. Podstata je v tom, že nahradíme zkoumaný systém modelem (uměle vytvořeným objektem). Model může být získán buď empiricky, tj. z naměřených hodnot, nebo teoreticky, tj. ze základních fyzikálních vlastností systému. Matematický model musí vhodně charakterizovat závislost výstupů systému na jeho vstupech viz [3]. Můžeme ho naformulovat buď s použitím klasické fyziky a nebo se započítáním kvantové mechaniky. Model s využitím kvantové mechaniky bude značně složitější. Používat se ale musí pouze při zkoumání jevů z mikrosvěta, např. studium elektronů. Do modelu bychom měli zahrnout pouze podstatné vlastnosti zkoumaného systému a ty méně podstatné vypustit. Modely fyzikálních systémů jsou obvykle sestaveny jako soustavy několika diferenciálních rovnic. U modelu je též možné měnit parametry a sledovat tak jejich vliv na výsledné řešení zkoumaného problému.

Model ale málokdy dokáže popsat dění dostatečně přesně. Proto je nutné znát omezení použitého modelu a nevyvozovat z modelování nepatřičné závěry. Výsledky modelu vypovídají pouze o modelu a ne o skutečnosti.

Podle charakteru studovaného procesu (systému) se modely dělí na:

- a) **Deterministické** - v modelu nejsou zahrnuty náhodné veličiny. Modely se vyznačují jednoznačně určenými příčinami a jejich následky. Hodnoty výstupních veličin jsou jednoznačně určeny průběhem veličin vstupních.
- b) **Stochastické** - zkoumaný problém nebo metoda řešení mají náhodný charakter viz [4].

1.4 Postup řešení fyzikálního úkolu pomocí počítačového modelování

Při řešení fyzikálního úkolu pomocí počítačového modelování zpravidla postupujeme dle následujících kroků:

- **Formulace problému a vytvoření modelu** - rozeberme fyzikální problém a na jeho bázi zformulujeme model pomocí zvolené techniky modelování. Model je téměř vždy jednodušší než studovaný jev. Při jeho formulaci se do něj snažíme zahrnout pouze podstatné vlastnosti zkoumaného systému. Ovšem nemáme jistotu, zda vůbec ty podstatné vlastnosti známe. Proto musíme výsledky modelování porovnat s daty zjištěnými experimentálně - viz níže.
- **Řešení modelu** - podle techniky modelování zvolíme numerickou metodu, kterou model vyřešíme. Z matematického hlediska budeme řešit soustavy obyčejných diferenciálních rovnic nebo rovnice integrodiferenciální, případně použijeme i postupy počtu pravděpodobnosti a matematické statistiky.
- **Porovnání výsledků, diskuse s experimentálními údaji (nezávislémi daty)** - výsledky modelování srovnáme s experimentálními daty, pokud je máme. Experimentální data by měla být přibližně stejná jako výsledky modelování. Pokud tomu tak není a výsledky modelování budou příliš odlišné, tak je třeba model opravit.

2. Techniky počítačového modelování

Techniky počítačového modelování odpovídají různým úrovním zobecnění, na kterých formulujeme matematický model zkoumaného systému. Rozeznáváme následující techniky:

2.1 Částicové (mikroskopické) techniky

Částicové techniky popisují studovaný jev na základě chování jeho dílčích částí, ze kterých se skládá. Částicovými technikami dosáhneme nejpřesnějšího popisu studovaného jevu, protože ho popisujeme na mikroskopické úrovni. Slovo „mikroskopický“ je třeba brát relativně, protože např. při popisu pevné látky pracujeme s atomy, zatímco při popisu galaxie si vystačíme s hvězdami. Výhodou částicového modelování je to, že přináší detailní informace o studovaném systému. Jeho nevýhoda je v tom, že programy jsou výpočetně značně náročné a to kvůli tomu, že rovnicemi popisujeme velký počet částic (10^4 až 10^8).

Při porovnání výsledků modelování s experimentálními údaji provádíme i statistické vyhodnocení těchto dat, protože experimentální fyzik není schopen poskytnout informace např. o poloze a rychlosti všech elektronů, atomů nebo molekul ve studovaném souboru, čímž by se část informace z modelování ztratila. Ovšem při statistickém vyhodnocení se tato informace neztratí a zůstane nám k dispozici a my se k ní můžeme dle potřeby vracet a získávat ještě další výsledky.

Podle chování studovaného systému rozeznáváme dvě techniky částicového modelování:

- **Deterministické techniky** - v modelu řešíme numericky pohybové rovnice popisující studovaný soubor částic. Matematicky to bude řešení soustav obyčejných diferenciálních rovnic. Příkladem deterministických technik je metoda molekulární dynamiky.
- **Stochastické techniky** - v modelu popisujeme chování jednotlivých částic stochasticky, tj. pomocí zákonů počtu pravděpodobnosti a matematické statistiky. Příkladem stochastických technik je metoda Monte Carlo.

2.2 Spojité (makroskopické) techniky

Studovaný jev popisujeme na makroskopické úrovni, tj. díváme se na něj jako na celek, např. na plazma se díváme jako na kontinuum o určité teplotě, tlaku, složení, rychlosti proudění, atd. Mezi těmito makroskopickými veličinami platí zákony zachování - hybnosti, energie, náboje, rovnice continuity, apod. Matematicky jsou tyto zákony zachování popsány parciálními diferenciálními rovnicemi [3].

Mezi výhody spojitěho modelování patří:

- Programy nebývají výpočetně náročné, jde o minuty strojového času proti dnům u modelování částicového.
- V modelu zpravidla řešíme menší počet rovnic než u technik částicových.

Mezi nevýhody spojitěho modelování patří:

- Použité rovnice jsou značně obecné a i přes to, že je upřesníme volbou počátečních a okrajových podmínek, nebývají získávané výsledky tak přesné jako u modelování částicového.
- Rovnice jsou ale mnohem složitější a nalezení vhodného algoritmu řešení včetně naprogramování bývá dosti těžké a na čas náročné.

Spojité techniky se nejvíce používají v experimentální fyzice. Příkladem jsou různé závislosti, např.: závislost tlaku na teplotě, elektrického proudu na teplotě,....

Pro spojitě modelovací techniky je k dispozici komerční software - viz např. COMSOL Multiphysics, Fluent, atd.

Numerické metody pro řešení spojitě formulovaných problémů dělíme na **klasické** (metody řešení diferenciálních rovnic) a **moderní** (např. metoda konečných prvků, metoda hraničních prvků, metoda okrajových elementů, metoda konečných diferencí, ...).

a) Klasické spojitě techniky

Fyzikální zákony jsou ve většině případů formulovány jako diferenciální rovnice. Jedná se buď o obyčejné nebo o parciální diferenciální rovnice . Tyto rovnice nelze často analyticky řešit. Je tedy třeba použít metody numerické matematiky [5].

1) Řešení obyčejných diferenciálních rovnic

Příkladem obyčejných diferenciálních rovnic je závislost fyzikálních veličin na čase, jako je tomu např. u pohybu částic plynu.

Budeme uvažovat diferenciální rovnice 1. řádu, kde je derivace přímo vyjádřena. Řešením je reálná funkce $y(x)$, která splňuje tuto rovnici:

$$\frac{dy(x)}{dx} = f(x, y) \quad (1)$$

Většinou existuje nekonečně mnoho funkcí, které splňují tuto rovnici. Jedno konkrétní řešení určíme tzv. **počáteční podmínkou**. Ta je známa ze zadání úlohy a má následující tvar:

$$y(x_0) = y_0 \quad (2)$$

x_0, y_0 - dané hodnoty

Metody řešení pro obyčejné diferenciální rovnice jsou následující:

□ Eulerova metoda

Je nejjednodušší metodou pro řešení diferenciálních rovnic. V diferenciální rovnici (1) nahradíme derivaci přibližným vzorcem:

$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i) \quad (3)$$

h - délka kroku

Dále pak vyjádříme y_{i+1} tj.:

$$y_{i+1} = y_i + hf(x_i, y_i) \quad (4)$$

Vzorec (4) je Eulerova metoda, která umožňuje při znalosti řešení v bodě x_i určit řešení v bodě x_{i+1} . Eulerova metoda je metodou **jednokrokovou** (k výpočtu nové hodnoty použijeme pouze jednu předchozí hodnotu).

Eulerova metoda je jednoduchá, ale není příliš přesná. Nepřesnost je způsobena tím, že z kroku x_i do x_{i+1} používáme na celém intervalu konstantní hodnotu derivace $f(x_i, y_i)$. Tato hodnota se však ve skutečnosti v průběhu kroku mění. Tuto chybu se snaží odstranit různé modifikace Eulerovy metody viz [5].

□ Rungovy-Kuttovy metody

Těmito metodami dostaneme několik odhadů derivace k_i v r různých bodech. Pro celý krok se potom využije vážený průměr těchto derivací s váhami w_i tj.:

$$y_{n+1} = y_n + h(w_1 k_1 + \dots + w_r k_r) \quad (5)$$

Bližší informace lze nalézt například v publikaci [5].

□ Metoda středního bodu

Je to metoda druhého řádu. První krok této metody se provede pomocí Eulerovy metody tj.:

$$y_1 = y_0 + hf(x_0, y_0) \quad (6)$$

Druhý krok vychází z předposledního bodu x_{i-1} za použití derivace z posledního bodu x_i tj.:

$$y_{i+1} = y_{i-1} + hf(x_i, y_i) \quad (7)$$

Metoda středního bodu je metodou **dvoukrokovou**, protože k výpočtu nové hodnoty y_{i+1} potřebujeme znát dvě předchozí hodnoty, tj. y_i a y_{i-1} .

□ Metody prediktor-korektor

Jedná se o **mnohokrokové** metody (k výpočtu nové hodnoty se využije znalost řešení ve více předchozích bodech). Pracujeme vždy s dvojicí metod. Jedna je metoda explicitní (prediktor), druhá je implicitní (korektor). Dvojici prediktor - korektor volíme tak, aby řád prediktoru byl o jedna menší než řád korektoru.

Nejprve pomocí prediktoru získáme první odhad nového řešení. Poté tento odhad použijeme jako start iteračního procesu, kterým se řeší korektor. Tímto iteračním

procesem se první odhad z prediktoru zpřesní. Výpočty se opakují, dokud nedosáhneme požadované přesnosti řešení korektoru.

2) Řešení parciálních diferenciálních rovnic

Na internetu lze nalézt mnoho materiálů zabývajících se numerických řešením diferenciálních rovnic a to jak obyčejných, tak i parciálních. Jako příklad lze uvést publikaci [5].

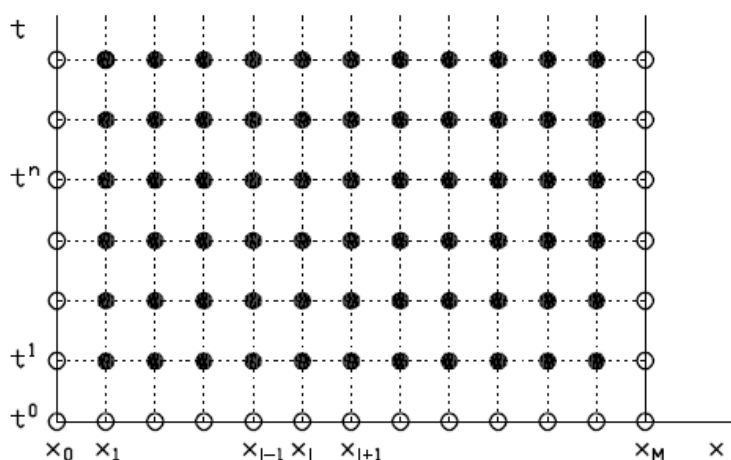
Budeme uvažovat lineární parciální diferenciální rovnice druhého řádu. Ty lze rozdělit na rovnice parabolické, hyperbolické a eliptické. Obecný tvar těchto rovnic pro dvě proměnné tj. $u = u(x_1, x_2)$ je následující:

$$a_1 \frac{\partial^2 u}{\partial x_1^2} + a_2 \frac{\partial^2 u}{\partial x_2^2} + a_{12} \frac{\partial^2 u}{\partial x_1 \partial x_2} + b_1 \frac{\partial u}{\partial x_1} + b_2 \frac{\partial u}{\partial x_2} + c = 0 \quad (8)$$

a_i, b_i, c_i - zadané funkce x_1 a x_2 .

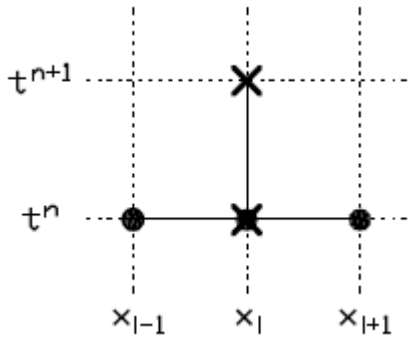
Základní metodou pro řešení lineárních parabolických parciálních diferenciálních rovnic je:

Metoda sítí (metoda konečných diferencí) - princip spočívá v tom, že místo spojité funkce $u(t, x)$ se hledají pouze odhady řešení v konečném počtu bodů. Body tvoří síť - viz obr. 1, odtud pochází název metody. Mezi metody sítí patří následující:



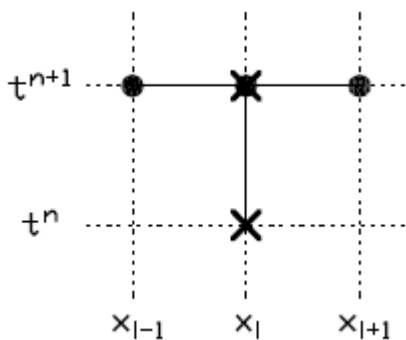
Obrázek 1: Konstrukce rovnoměrné ortogonální sítě. Převzato z [5].

- **Explicitní metoda** - je nejjednodušší variantou metody sítí. Jedná se o metodu **podmíněně stabilní** (k zajištění stability potřebujeme nějakou podmínku). Schéma řešení je na obr. 2.



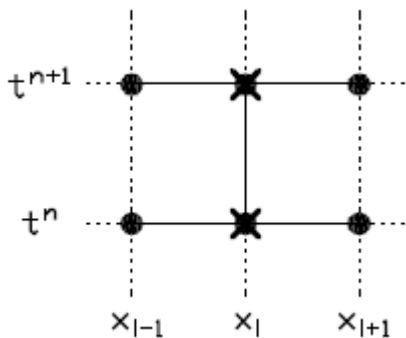
Obrázek 2: Schéma řešení pro explicitní metodu. Převzato z [5].

- **Implicitní metoda** - tato metoda vylepšuje předcházející metodu tím, že odstraňuje její nedostatek, tj. podmínku stability. Implicitní metoda je tedy **stabilní** (nepodmíněně stabilní). Schéma řešení je na obr. 3.



Obrázek 3: Schéma řešení pro implicitní metodu. Převzato z [5].

- **Crankovo - Nicholsonovo schéma** - je **stabilní** (nepodmíněně stabilní) metoda. Časově je stejně náročná jako implicitní metoda, ale je přesnější. Schéma řešení je na obr. 4.



Obrázek 4: Schéma řešení pro Crankovo-Nicholsonovo schéma. Převzato z [5].

b) Moderní spojité techniky

- **Metoda konečných prvků** - v současnosti je považována za jednu z neúčinnějších numerických (přibližných) metod, které se využívají pro řešení úloh popsanych diferenciálními rovnicemi viz [6].
- **Metoda hraničních prvků** - je to univerzální metoda pro řešení integrálních rovnic. Hojně se využívá pro řešení elektromagnetických polí viz [7].
- **Metoda konečných diferencí** - používá se tam, kde se využije její vhodnosti při aproximaci řešení spojených s teplotním přenosem.
- **Metoda okrajových elementů** - je novější metoda pro řešení okrajových počátečních problémů. Používá se ve spojení s teplotní teorií u slévarenských procesů.

2.3 Hybridní techniky

Je to kombinace přístupu spojitého a částicového, opět s cílem získat výsledek co nejrychleji a s co největší přesností. Rozlišujeme dva druhy hybridního modelování:

- **Hybridní modelování na částicové úrovni** - zde kombinujeme modelování deterministické (metoda molekulární dynamiky) a modelování stochastické (metoda Monte Carlo), tzn. vycházíme pouze z částicového modelování. Příkladem hybridních technik na částicové úrovni je studium jevů z mikrosvěta, tj. studovanými částicemi jsou elektrony, atomy, molekuly, ionty. Při studiu těchto částic musíme použít hybridních technik, protože nelze určit jejich počáteční stavy, tj. rychlost a souřadnice přímým pozorováním. Proto se využívá stochastických postupů, tj. postupů počtu pravděpodobnosti a matematické statistiky, na základě kterých se pomocí rozdělovacích funkcí získaných teoreticky určí počáteční stavy studovaných částic.

- **Úplné hybridní modelování** - zde kombinujeme částicové a spojitě modelování a to tak, abychom z modelování částicového převzali přesnost výpočtu a z modelování spojitého rychlost výpočtu.

Výsledkem kombinování dvou nebo i více technik modelování je jejich kompromis. Tzn., že výsledek nemá žádné výrazné nedostatky z původních metod, ale nemá ani žádné špičkové parametry rychlosti nebo přesnosti výpočtu. Zbavíme se sice nedostatků, které měly původní metody, ale je to na úkor času. Příprava modelu a výsledného programu je totiž časově náročnější než kdybychom sečetli výsledné časy příprav původními technikami.

3. Metoda Molekulární dynamiky

3.1 Popis metody

Metoda molekulární dynamiky je deterministická (v modelu nejsou náhodné veličiny, tj. hodnoty výstupních veličin jsou jednoznačně určeny průběhem vstupních veličin) částicová metoda. Studovaný systém popisujeme pomocí chování jeho dílčích částí. Principem této metody je řešení Newtonových pohybových rovnic pro N částic. Z matematického hlediska se jedná o soustavu obyčejných diferenciálních rovnic. Částice, jejichž chování popisujeme, mezi sebou vzájemně interagují, proto je tato metoda mnohočásticová.

Použití metody molekulární dynamiky je značně rozsáhlé. Typickým použitím je modelování trajektorií nebo studium dynamiky (příčin pohybu) částic a těles. V současnosti se tato metoda používá téměř všude, např. atomy v plynech, elektrony a ionty v plazmatu, hvězdy při studiu galaxií, chování kapalin, pevných látek, dynamika velkých makromolekul, biologické systémy - proteiny, nukleové kyseliny, ... viz [8], [9].

3.2 Idea řešení metodou molekulární dynamiky

1) Rozbor problému a vytvoření modelu

Rozebereme zkoumaný problém a určíme, zda budeme pracovat v jedno, dvou a nebo v trojrozměrném prostoru. Poté popíšeme zkoumaný systém pomocí N částic, určíme pracovní oblast (viz níže), zjistíme síly působící na jednotlivé částice a počáteční stavy těchto částic. Následně vytvoříme model, tj. popíšeme jednotlivé části zkoumaného systému jejich pohybovými rovnicemi.

2) Řešení modelu

Určíme počáteční podmínky pohybových rovnic a následně je vyřešíme. Cílem těchto snah je obvykle nalézt trajektorii částic a nebo pouze částice v nějakém časovém intervalu.

3.3 Pracovní oblast

Před zahájením simulace je nutné zvolit **velikost pracovní oblasti**, její **tvár** a **okrajové podmínky**. Tyto parametry se v jednotlivých případech mohou lišit, protože rozlišujeme dva základní případy:

- **Popis chování velkého počtu částic (těles), které zaujímají velký objem** - pracovní oblast je úměrná velikosti částic, tj. bude výrazně menší než objem, který zaujímají studované částice. To znamená, že zvolíme pouze jakýsi „výřez“ do zkoumaného systému. V tomto případě musíme určit tvar a velikost pracovní oblasti a další okrajové podmínky, protože pracovní oblast je pouze malá část z celkového prostoru zaplněného částicemi.
- **Popis chování částic (těles) v omezené oblasti** - pracovní oblast musí být tak velká, aby zahrnula celý studovaný systém, tj. tvar a velikost přizpůsobujeme zkoumanému systému. V tomto případě se okrajové podmínky neformulují, protože nejsou zapotřebí. Studovaná částice nebo těleso obvykle nemůže pracovní oblast opustit, pokud se ale tak stane, tak dojde k ukončení programu.

Velikost pracovní oblasti

Při volbě velikosti pracovní oblasti posuzujeme tato kritéria:

1) fyzikální možnost

Tato možnost závisí na charakteru silového působení v našem modelu. Rozeznáváme dva typy sil:

- **síly dalekodosahové** - jedná se např. o elektrostatickou nebo o gravitační sílu. V tomto případě zvolíme velikost pracovní oblasti tak, aby na částici ve středu oblasti působily zanedbatelné síly od částic mimo pracovní oblast, tzn., že na vzdálenosti $L/2$, což je polovina pracovní oblasti, klesnou síly na zanedbatelnou hodnotu.

- **síly krátkodosahové** - jedná se např. o síly v pevných látkách. V tomto případě je silové působení částic mimo pracovní oblast na částici ve středu pracovní oblasti téměř nulové.

2) výpočetní možnost

Velikost pracovní oblasti volíme podle počtu částic, které by měla obsahovat. Pracovní oblast bývá v přirozeném měřítku, protože pokud používáme přirozená měřítka všech veličin, tak je jejich vyhodnocení nejjednodušší.

Tvar pracovní oblasti

Tvar pracovní oblasti záleží na tom, zda jsou částice v řešeném problému rozloženy symetricky. Pokud ano, tak tvar pracovní oblasti podřídíme tomuto rozložení, neboť tím zjednodušíme popis procesů na hranici pracovní oblasti.

Pokud studovaný problém symetrii nemá, nebo má jen nějakou nevýraznou, tak se volí tvar pracovní oblasti co nejjednodušší. To je velice výhodné, protože se nám tím zjednoduší další manipulace s daty při modelování.

Volba okrajových podmínek

Pokud je pracovní oblast pouze jakýmsi výřezem z celého objemu zaplněného částicemi, tak budou částice neustále vylétat z pracovní oblasti do okolního prostoru a také budou neustále vstupovat z okolního prostoru do pracovní oblasti. Tento proces se v modelu simuluje tzv. **cyklickými okrajovými podmínkami**. Tzn., když částice vylétne z pracovní oblasti, tak vstoupí hned zase protilehlou stěnou zpět. V případě, že má pracovní oblast jednoduchý tvar, tak cyklické okrajové podmínky budou realizovány zcela jednoduše. Tj. v trojrozměrné pracovní oblasti o hraně L má každá částice souřadnici x, y, z . Každá souřadnice každé částice musí ležet v rozmezí $\langle 0, L \rangle$. V modelu již netestujeme, zda se částice pokouší opustit pracovní oblast, ale sledujeme pouze hodnoty souřadnic a k těm pak buď přičteme nebo odečteme hodnotu L k příslušné souřadnici, která z intervalu $\langle 0, L \rangle$ vybočí viz [3].

Je nutné si uvědomit, že při použití cyklických okrajových podmínek klademe na zkoumaný systém jistá fyzikální omezení a to:

- a) Prostředí v systému musí být homogenní.
- b) Prostředí nesmí mít žádný usměrněný tok částic.

Pokud systém tyto předpoklady nespĺňuje, je nutné mít v modelu zdroj částic, který bude kompenzovat částice, které vylétnou z pracovní oblasti.

3.4 Řešení translace

V této části se budeme zabývat užitím metody molekulární dynamiky při řešení pohybových rovnic a při výpočtu výsledného silového působení na konkrétní částici v zkoumaném systému.

3.4.1 Pohybové rovnice

Řešíme soustavy pohybových rovnic pro všechny částice. Jsou to rovnice typu :

$$\begin{aligned}\vec{F}_i &= m_i \vec{a}_i \\ i &= 1, \dots, N\end{aligned}\tag{9}$$

tj. 2. Newtonův zákon. Zrychlení vyjádříme jako druhou derivaci dráhy podle času a soustavu rovnic (9) převedeme na soustavu diferenciálních rovnic 2. řádu.

$$\vec{F}_i = m \frac{d^2 \vec{r}_i}{dt^2}\tag{10}$$

Tyto diferenciální rovnice převedeme na diferenční rovnice (spojitou časovou osu diskretizujeme posloupností časů $t_0, t_1, t_2, \dots, t_{\max}$, kde $t_{k+1} - t_k = \Delta t$). Celý výpočet je nutné „odstartovat“ pomocí počátečních podmínek, které nám např. udávají rychlosti částic v čase t_0 . Tímto postupem dostaneme algoritmus pro řešení soustavy pohybových rovnic pro všech N částic. Na řešení soustavy pohybových rovnic existuje spousta numerických metod. Základní jsou Eulerova metoda, Verletův algoritmus a metoda leap - frog.

Eulerova metoda

Tato metoda je univerzální, ale pomalejší než Verletův algoritmus a metoda leap - frog. Algoritmus lze shrnout do tří kroků:

- 1) Zvolíme počáteční podmínky tj.: \vec{r}_i^0, \vec{v}_i^0
- 2) Přejdeme z času t_0 do t_1, \dots, t_k do t_{k+1} .
- 3) Dále pak určíme nové polohy a rychlosti.

$$\begin{aligned}
 \vec{r}_i^{K+1} &= \vec{r}_i^K + \vec{v}_i^K \Delta t + \frac{\vec{F}_i^K \Delta t^2}{2m_i} \\
 \vec{v}_i^{K+1} &= \vec{v}_i^K + \frac{\vec{F}_i^K \Delta t}{m_i} \\
 \vec{F}_i^{K+1} &= \dots
 \end{aligned}
 \tag{11}$$

Eulerovu metodu lze použít i tam, kde síla závisí na rychlosti. Příkladem tedy může být elektromagnetické pole.

Verletova metoda

Je to metoda 2. řádu. Rychlost počítáme na základě znalosti síly \vec{F}_i^{k+1} , proto musí být na rozdíl od Eulerova algoritmu (11) pořadí 2. a 3. řádku zaměněno.

Postup:

- 1) Zvolíme počáteční podmínky tj.: \vec{r}_i^0, \vec{v}_i^0
- 2) Přejdeme z času t_0 do t_1, \dots, t_k do t_{k+1} .
- 3) Dále pak postupujeme následovně:

$$\begin{aligned}
 \vec{r}_i^{K+1} &= \vec{r}_i^K + \vec{v}_i^K \Delta t + \frac{\vec{F}_i^K \Delta t^2}{2m_i} \\
 \vec{F}_i^{K+1} &= \dots \\
 \vec{v}_i^{K+1} &= \vec{v}_i^K + \frac{(\vec{F}_i^K + \vec{F}_i^{K+1}) \Delta t}{2m_i}
 \end{aligned}
 \tag{12}$$

Tento algoritmus je použitelný tam, kde síla nezávisí na rychlosti částice. Příkladem použití může být kinetická teorie plynů, fyzika plazmatu.

Metoda „leap -frog“

Je to metoda 2. řádu. Stejně jako v předchozí metodě počítáme rychlost na základě znalosti síly \vec{F}_i^{k+1} , proto musí být na rozdíl od Eulerova algoritmu (11) pořadí 2. a 3. řádku zaměněno.

Od Verletova algoritmu se liší tím, že rychlosti a polohy jsou vypočítány v různých časech vůči sobě posunutých o půl diferenčního kroku. Její výhodou oproti Verletově algoritmu je to, že rychlost je explicitně vypočítána a že k uložení síly nepotřebujeme dva vektory, což snižuje nároky na paměť.

Postup:

- 1) Zvolíme počáteční podmínky tj.: $\vec{r}_i^0, \vec{v}_i^{\frac{1}{2}}$
- 2) Přejdeme z času t_0 do t_1, \dots, t_k do t_{k+1} .
- 3) Dále pak postupujeme následovně:

$$\begin{aligned}\vec{r}_i^{K+1} &= \vec{r}_i^K + \vec{v}_i^{K+\frac{1}{2}}\Delta t + \frac{\vec{F}_i^K \Delta t^2}{2m_i} \\ \vec{F}_i^{K+1} &= \dots \\ \vec{v}_i^{K+\frac{3}{2}} &= \vec{v}_i^{K+\frac{1}{2}} + \frac{\vec{F}_i^{K+1} \Delta t}{m_i}\end{aligned}\tag{13}$$

Tento algoritmus stejně jako ten předchozí je použitelný tam, kde síla nezávisí na rychlosti částice. Příkladem použití může být kinetická teorie plynů, fyzika plazmatu. Nevýhodou tohoto algoritmu je to, že na začátku potřebujeme počáteční hodnoty polohy a rychlosti v čase $t_{1/2}$. Ovšem tyto hodnoty nebývají z experimentu známé a proto je nutné je z počátečních hodnot dopočítat.

3.4.2 Výpočet silového působení

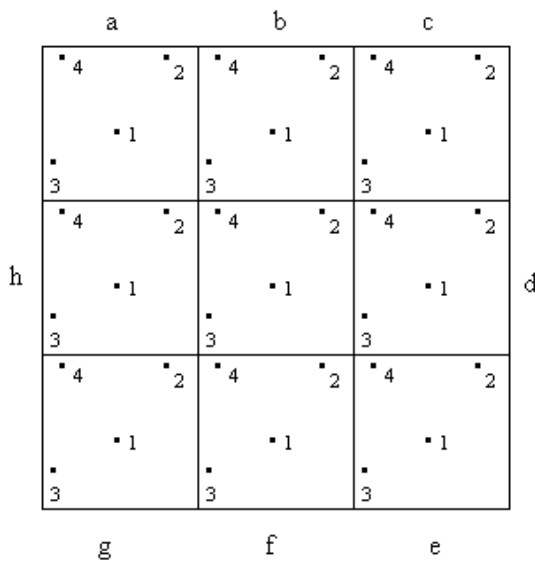
Při výpočtu celkové síly působící na částici je potřeba rozlišit, zda se částice nachází blízko středu pracovní oblasti nebo na okraji.

Leží - li částice blízko středu pracovní oblasti, tak sílu působící na konkrétní částici vypočítáme jako vektorový součet sil, které působí od ostatních $N - 1$ částic v pracovní oblasti.

Pokud se částice nachází na okraji pracovní oblasti a my započítáme jen silové působení od ostatních částic v pracovní oblasti je výpočet síly chybný, neboť zanedbáváme větší silové působení od částic, které jsou blízko za hranicí pracovní oblasti. Vliv částic, které jsou blízko za hranicí pracovní oblasti nelze zahrnout přímo. Tím bychom pouze zvětšili pracovní oblast. Nejčastěji zahrneme vliv těchto částic tak, že soubor o velkém počtu náhodně rozložených částic obklopíme soubory periodickými,

tj. kopiemi původního souboru. Tento postup je přesný pouze v případě přesné periodicity částic, např. modelování krystalů. V případech, kdy periodicitu není přesná, např. nabitě částice v plazmatu, je tento postup správný pouze když je počet částic v pracovní oblasti veliký.

Pracovní oblast obsahující N částic obklopíme ze všech stran kopiemi této oblasti i s jejími částicemi - viz obr. 5. Na tomto obrázku jsou kopie označeny písmeny a - h, $N = 4$, pracovní oblast je čtverec uprostřed. Pokud pracujeme ve dvojrozměrném modelu, je kopií 8, v třírozměrném modelu 26. Kopie zůstávají spjaty s původní oblastí, tj. každý pohyb původní částice se projeví ve všech jejích kopiích.



Obrázek 5: Kopie pracovní oblasti.

V případě obrázku 5 lze pro výslednou sílu psát:

$$\vec{F}_1 = \vec{F}_{12} + \vec{F}_{13} + \vec{F}_{14} \quad (14)$$

Síla působící na částici 1 se vypočítá jako vektorový součet sil, které působí od ostatních $N - 1$ částic v pracovní oblasti.

$$\begin{aligned} \vec{F}_2 &= \vec{F}_{21} + \vec{F}_{23} + \vec{F}_{24} \\ \vec{F}_3 &= \vec{F}_{31} + \vec{F}_{32} + \vec{F}_{34} \\ \vec{F}_4 &= \vec{F}_{41} + \vec{F}_{42} + \vec{F}_{43} \end{aligned} \quad (15)$$

Zde horní indexy označují to, že částice nepatří do původní oblasti ale do některé z kopií.

I u částic 2, 3, 4 vypočítáme sílu působící na tyto částice stejným způsobem jako u částice 1. Jenom s tím rozdílem, že uvažujeme síly od nejbližších částic a to bez ohledu na jejich umístění.

Ve výsledném součtu silového působení se každá částice nebo její kopie objeví jednou. Tato skutečnost umožňuje výše uvedený postup zjednodušit. A to následovně:

- Nepočítáme se silami, ale pouze se vzdálenostmi částic, protože všechny typy sil ubývají se vzdáleností.
- Nepočítáme s absolutními hodnotami vzdáleností, ale jen s jejich pořadím. Stačí tedy počítat s druhými mocninami vzdáleností.

3.5 Metody urychlující výpočet silového působení

3.5.1 Metoda P-I-C („Particle-In-Cell“)

Popis metody

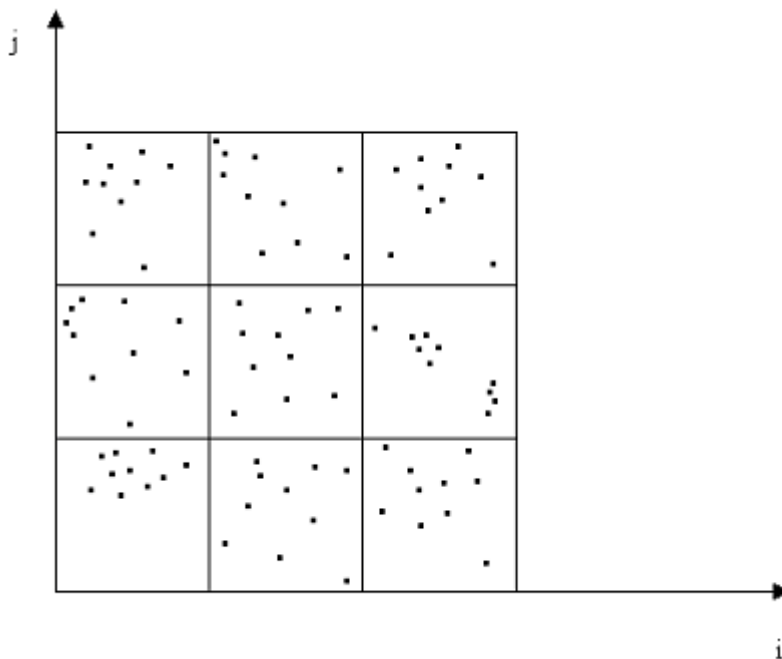
Jsou to efektivnější metody výpočtu síly, než které dává metoda molekulární dynamiky, ale snižují přesnost. V metodě P-I-C provádíme diskretizaci časové osy a prostoru. To znamená, že nelze studovat procesy, které probíhají v menší délce než je délka hrany buňky Δx a kvůli zavedení diskretizace času není možné stejně jako v metodě molekulární dynamiky studovat velice rychlé procesy. Metoda P-I-C má několik modifikací. Z nichž základní jsou Nearest Grid Point (NGP) a Cloud in Cell (CIC). Používá se tam, kde máme větší počet částic, tj. kinetická teorie plynů, fyzika plazmatu ,....

Metodou P-I-C dostaneme pouze dílčí algoritmus pro nalezení síly působící na i - tou částici v souboru vzájemně interagujících částic. Tuto sílu musíme dosadit do pohybových rovnic, např. Eulerova algoritmu, Verletova algoritmu, „leap-frog“ algoritmu a provést integraci celé soustavy v požadovaném časovém intervalu.

Idea řešení metodou P-I-C

Pracovní oblast se v závislosti na jejím tvaru rozdělí na soustavu menších buněk (částí) - viz obr. 6. Na tomto obrázku je pracovní oblast rozdělena na čtvercové buňky.

To znamená, že N částic v původní pracovní oblasti se tak rozdělí mezi jednotlivé buňky. Podle počtu částic volíme hustotu (počet buňek) prostorové sítě. Když bude délka hrany buňky Δx příliš malá, bude prostorové rozložení lokálního pole příliš jemné. Tím narostou fluktuace v intenzitách lokálního pole mezi jednotlivými buňkami (částmi) sítě. Ovšem pokud zvolíme délku hrany buňky Δx příliš velkou, tak bude prostorové rozložení lokálního pole příliš hrubé. Empiricky lze odhadnout, že v jedné buňce by mělo být asi 10 částic.



Obrázek 6: Pracovní oblast metody P- I- C.

Výpočet silového působení metodou P-I-C má dvě složky a to: výpočet síly působící na částici v lokálním elektrickém nebo gravitačním poli a určení tohoto pole.

Vlastní postup výpočtu touto metodou pro lokální elektrické je následující:

1) Nejprve použijeme vztah na výpočet síly působící na částici z externího zdroje tj:

$$\vec{F}_i^{\text{ext}} = q_i \vec{E}^{\text{ext}}(\vec{r}_i) \quad (16)$$

kde:

$\vec{E}^{\text{ext}}(\vec{r}_i)$ – intenzita vnějšího pole v místě i - té částice

q_i – náboj i - té částice

2) Intenzitu lokálního elektrického pole E^{lok} určíme následujícím způsobem:

- Elektrický náboj všech částic v buňce q_{ij} (i,j - souřadnice buňky) poskládáme do jedné výsledné částice uprostřed buňky. Další postup viz níže - metoda NGP a CIC.
- Vydělíme náboj v buňce q_{ij} jejím objemem a tím získáme prostorovou hustotu náboje ρ_{ij} tj.:

$$\rho_{ij} = \frac{Q_{ij}}{V_{ij}} \quad (17)$$

- Vyřešíme Poissonovu rovnici tj.:

$$\Delta U = -\frac{\rho}{\epsilon_0} \quad (18)$$

a tím dostaneme hodnoty elektrického potenciálu U_{ij} v každé buňce.

- Přejdeme pomocí diferenčních schémat tj.:

$$(E_{ij})_x = \frac{U_{i+1j} - U_{i-1j}}{2\Delta x} \quad (19)$$

Δx - je délka hrany buňky

od elektrostatického potenciálu k hledané intenzitě elektrického pole $\vec{E}_{ij}^{\text{lok}}$ v jednotlivých buňkách viz [3].

3) Potom vypočítáme sílu působící na částici z lokálního zdroje tj.:

$$\vec{F}_i^{\text{lok}} = q_i \vec{E}^{\text{lok}}(\vec{r}_i) \quad (20)$$

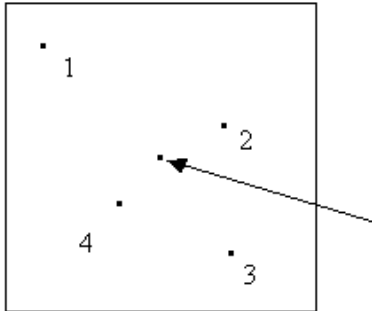
4) Nyní již můžeme pro výslednou sílu použít následujícího vztahu:

$$\vec{F}_i(\vec{r}_i) = \vec{F}_i^{\text{ext}}(\vec{r}_i) + \vec{F}_i^{\text{lok}}(\vec{r}_i) \quad (21)$$

Tuto sílu dosadíme do pohybových rovnic a provedeme integraci celé soustavy v požadovaném časovém intervalu.

a) **Algoritmus NGP** („Nearest Grid Point“)

Tento algoritmus je jednou z možností, jak získat celkový elektrický náboj q_{ij} v buňce. Tato metoda je založena na sečtení všech nábojů v buňce. Je nutné pochopitelně dbát na znaménka jednotlivých nábojů. Výsledný náboj je poté přesunut do středu buňky viz obr. 7.



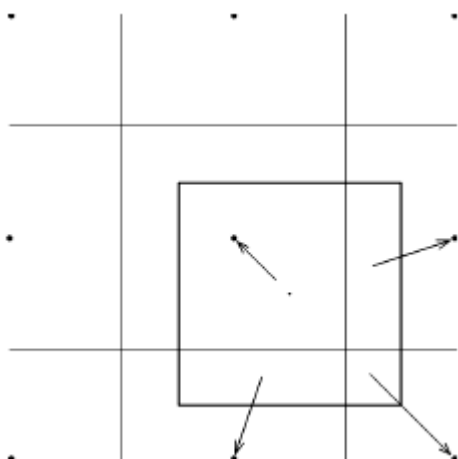
Obrázek 7: Skládání náboje metodou NGP.

Metoda NGP je při skládání náboje rychlejší než metoda CIC, ale má při rozložení náboje podstatně větší fluktuace. To se promítne do většího rozptylu hodnot intenzit lokálního elektrického pole.

b) **Algoritmus CIC** („Cloud In Cell“)

Tento algoritmus je druhou z možností, jak získat celkový elektrický náboj q_{ij} v buňce. Algoritmus CIC v podstatě není nic jiného než lineární interpolace náboje. Předpokladem ovšem je, že náboj není bodový, ale má tvar oblaku. Uvažuje se konstantní hustota nábojového oblaku a tvar stejný jako buňka. Tento oblak se obvykle rozprostírá ve více buňkách a tím přispívá k několika výsledným nábojům ve středu těchto buněk samozřejmě úměrně svému objemu v příslušných buňkách viz obr. 8.

Metoda CIC vede na rozložení náboje s menšími fluktuacemi, což umožňuje integrovat pohybové rovnice v metodě CIC s větším časovým skokem. Tato výhoda vykompenzuje pomalejší skládání náboje touto metodou.



Obrázek 8: Skládání náboje metodou CIC. Převzato z [3].

Nedostatek metody P-I-C

Metoda P-I-C má slabé místo v tom, že náboje posunujeme z jejich správných poloh do středů buněk. Poté vypočítaná hodnota intenzity lokálního pole bude odpovídat této posunuté poloze a ne té správné. V případě částic, které jsou vzdálenější od i -té částice pro kterou počítáme sílu, je tato chyba zanedbatelná. Ovšem pokud jsou částice blíže, tak tato chyba narůstá. Byly navrženy metody, které tuto chybu odstraňují.

3.5.2 Další optimalizované metody

Když je počet interagujících částic výrazně vyšší než 10^6 , tak přestává být využitelná i metoda P-I-C. Na výpočet vzájemného silového působení interagujících částic je tedy nutné hledat jiné algoritmy. Například se může jednat o Barnesův-Hutův algoritmus, Ewaldovu sumaci nebo metodu P^3M .

Postup je obdobný jako v metodě P-I-C. Pracovní oblast se rozdělí na soustavu buněk (sítě), poté se určí hodnota lokálního pole. Tato síť se postupně zjemňuje. Tím se vykompenzuje chyba předešlé metody. Zjemňování sítě lze provést dvěma způsoby. Při prvním způsobu se zjemňování provádí pro každou částici, na kterou počítáme silové působení zvlášť. Nevýhodou tohoto postupu je pomalost. Při druhém způsobu se zjemňování provádí pro všechny částice najednou. Nevýhodou tohoto způsobu je složitost.

Jedná se o složité algoritmy, které se programátorsky řeší hodně dlouho. Pro srovnání v metodě P-I-C se jedná o desítky hodin. Proto se nejběžněji využívají jako předem připravené podprogramy.

Pochopitelně existují i postupy, které z náboje v síti vypočítají sílu působící na částici přímo, tj. bez výpočtu hodnoty lokálního pole. Mezi tyto postupy patří např. princip superpozice nebo metoda multipólového rozvoje.

4. Metoda Monte Carlo

4.1 Charakteristika metody

Metoda Monte Carlo patří mezi částicové metody. Je to metoda stochastická, což znamená, že hledaný výsledek je získáván na základě počtu pravděpodobnosti. Značný rozvoj této metody lze datovat do období na konci druhé světové války. V té době jsou již v oblasti atomového výzkumu využívány počítače. Mezi tvůrce metody patří: J. von Neumann (formuloval statistické základy), E. Fermi, S. A. Ulam, N. Metropolis a H. Kahn.

Základní princip metody je však mnohem starší. Již v roce 1777 Georges de Buffon formuloval úlohu s jehlou. Řešením této úlohy bylo možné s využitím náhodných jevů určit hodnotu čísla π [10].

Použití metody Monte Carlo je rozsáhlé. Používá se téměř ve všech vědních disciplínách. Ovšem řešení problému pomocí metody Monte Carlo není vždy tím nejvýhodnějším a to jak z hlediska jednoduchosti, přesnosti i rychlosti výpočtu.

4.2 Idea řešení problému metodou Monte Carlo

1. Rozbor problému a vytvoření modelu

Rozebereme studovaný fyzikální problém a popíšeme ho pomocí náhodné veličiny s pevně daným rozdělením pravděpodobností a oborem hodnot.

2. Generování náhodných veličin, jejich transformace na veličiny s daným statistickým rozdělením

Náhodná veličina se nagenereje (vytvoří) s pevně daným rozdělením pravděpodobnosti a poté se přetransformuje v námi hledanou náhodnou veličinu. Tímto krokem dostaneme pouze jednu realizaci hledané náhodné veličiny. Proto ho musíme opakovat v cyklech, abychom dostali potřebný počet realizací hledané náhodné veličiny. Počet opakování musí být značný, neboť metoda Monte Carlo nekonverguje příliš rychle.

Experimentálně bylo zjištěno, že chyba metody Monte Carlo roste se zvětšujícím se počtem pokusů N podle vztahu

$$\vartheta \approx 1/\sqrt{N} \quad (22)$$

Ze vztahu (22) vyplývá, že metoda Monte Carlo nepatří k nejefektivnějším.

3. Statistické zpracování výsledků

Pokud hledáme jednočíselný výsledek, tak ten je zpravidla dán prvním nebo druhým momentem námi zvolené náhodné veličiny. Nejčastěji to bývá první moment, tj. střední hodnota. Když bude řešením našeho úkolu nějaké rozdělení, tak se snažíme model vytvořit tak, aby hledané rozdělení představovalo rozdělení pravděpodobností naší náhodné veličiny a to již z posloupnosti nagenерованých náhodných veličin přibližně zrekonstruujeme [3].

4.3 Generování náhodných čísel

Metoda Monte Carlo modeluje náhodný proces pomocí operací s náhodnými čísly. Tato čísla jsou získávána speciálním způsobem. Pro metodu Monte Carlo byly navrženy následující způsoby:

- Fyzikální generátory
- Tabulky náhodných čísel
- Vypočítaná náhodná čísla

Fyzikální generátory

Nejprve se využívala ruleta, později systémy přímo propojené s počítačem využívající náhodných procesů. Tyto generátory se používaly až do začátku šedesátých let. Jejich nevýhodou byla pomalost.

Tabulky náhodných čísel

Jednalo se o fyzikální generátory, které produkovaly náhodná čísla. Ta se ukládala na magnetickou nebo děrnou pásku. Vzhledem k rychle se zvyšující výkonnosti výpočetní techniky i toto řešení brzy přestalo vyhovovat.

Vypočítaná náhodná čísla

V současnosti nejčastěji používaný typ náhodných čísel. Jedná se o algoritmy, které vedou na posloupnosti náhodných čísel. Tato čísla se také nazývají **pseudonáhodná**. To protože se od skutečně náhodných čísel odlišují tím, že posloupnost nagenovaných čísel není nekonečná, po uplynutí určitého časového intervalu se začne celá posloupnost opakovat. Tento interval nazýváme periodou generátoru.

Vývoj postupoval nezadržitelně dál a v 90. letech byla vytvořena nová generace fyzikálních generátorů náhodných čísel s využitím nových principů experimentální fyziky. Tyto generátory již vyhovují svou rychlostí i současným počítačům.

V současnosti se také vrací myšlenka na vytváření tabulek náhodných čísel využívajících moderních metod, tj. využití rychlých velkokapacitních médií - CD-ROM a DVD-ROM viz [3].

4.4 Transformace náhodných veličin

Generátor náhodných čísel zpravidla nagenuje náhodnou veličinu s rovnoměrným rozdělením na intervalu (0,1). Ta se poté musí přetransformovat v námi požadovanou náhodnou veličinu. K tomuto účelu bylo navrženo velké množství algoritmů. Můžeme je rozdělit na:

- **Algoritmy obecné** - vyřeší větší množství problémů, ale s nízkou efektivitou.
- **Algoritmy specializované** - pro rozehrávání konkrétní náhodné veličiny.

Předpokladem je, že máme k dispozici spojitou náhodnou veličinu Y rovnoměrně rozdělenou na intervalu (0,1). Parametry jejího rozdělení jsou:

$$\begin{aligned} E(Y) &= \frac{1}{2} \\ D(Y) &= \frac{1}{12} \end{aligned} \tag{23}$$

Při realizaci náhodné veličiny Y na počítači získáme m -místná diskrétní čísla Y' , která budou mít jiné momenty:

$$E(Y') = \frac{(1 - 2^{-m})}{2} \tag{24}$$

$$D(Y') = \frac{(1 - 2^{-2m})}{12}$$

Rozdíly mezi Y a Y' lze díky vícemístné aritmetice většinou zanedbat a pak můžeme nagenеровaná náhodná čísla podle potřeby považovat buď za spojitou a nebo diskrétní náhodnou veličinu viz [3].

Obecné algoritmy pro transformaci náhodných veličin

V metodě Monte Carlo se místo pojmu transformace používá pojem „rozehraní“ náhodné veličiny.

Rozehraní diskrétní náhodné veličiny

Budeme mít veličinu Z , která je zadána následovně:

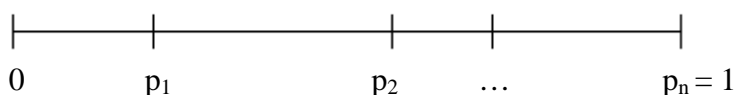
$$Z = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ p_1 & p_2 & p_3 & \dots & p_n \end{pmatrix} \tag{25}$$

Postup:

- V počítači vytvoříme vektor o n složkách: $p_1, p_2, p_3, \dots, p_{n-1}, 1$ (viz obr. 9).
- Nagenерujeme jednu hodnotu Y a určíme do kterého intervalu padne, tzn. budeme postupně testovat podmínku:

$$Y \leq \sum_{i=1}^j p_i \tag{26}$$

- První interval j , pro který bude tato podmínka splněna, určí příslušnou hodnotu veličiny Z , tj. $Z = x_j$.



Obrázek 9: Rozložení pravděpodobností p_i na intervalu $\langle 0,1 \rangle$.

Uvedený postup je použitelný pro transformaci libovolné diskrétní náhodné veličiny. Ovšem pokud je počet možných hodnot veličiny Z , n příliš velký, tak tento postup není nejvhodnější, protože hledání příslušného intervalu j trvá dlouho. Proto lze v této situaci použít poněkud rychlejší postup - např. metodu půlení intervalu.

Rozehrání spojité náhodné veličiny

Pro rozehrávání spojité náhodné veličiny je algoritmů více. Uvedeme si:

- Metodu superpozice
- Metodu inverzní funkce
- Metodu výběru

Metoda superpozice

Při použití této metody hledáme náhodnou veličinu X s distribuční funkcí:

$$F(x) = \sum_{i=1}^m c_i F_i(x) \quad (27)$$

$F_i(x)$ - jsou též distribuční funkce

$c_1 + \dots + c_m = 1$ a všechna $c_i > 0$.

K tomuto účelu zavedeme diskrétní náhodnou veličinu η s rozdělením:

$$\eta = \begin{pmatrix} 1 & 2 & \dots & m \\ c_1 & c_2 & \dots & c_m \end{pmatrix} \quad (28)$$

Postup jak najít náhodnou veličinu X je následující:

- Nageneryjeme dvě nezávislé hodnoty veličiny Y tj. Y_1 a Y_2 .
- Číslem Y_1 rozehráme hodnotu η .
- Z rovnice $F_k(X) = Y_2$, kde $\eta = k$ určíme veličinu X viz [3].

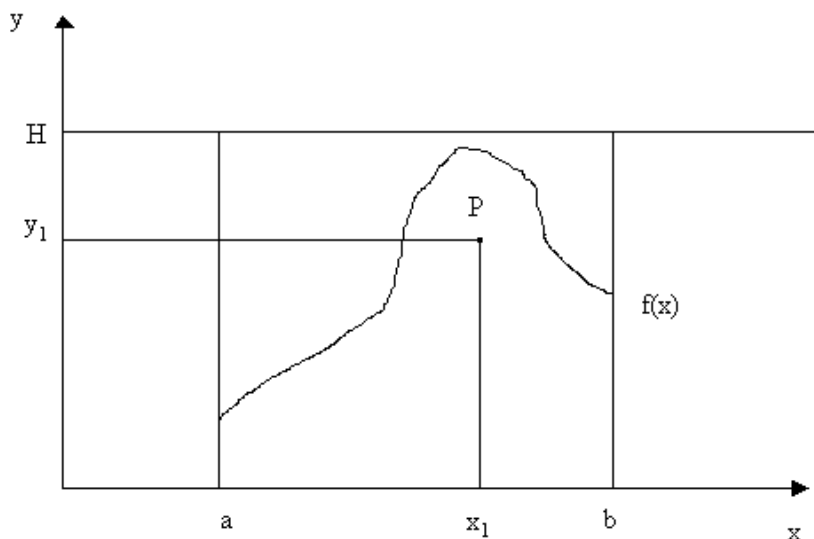
Metoda inverzní funkce

Známe-li hustotu pravděpodobnosti $f(x)$, obor hodnot $\langle a, b \rangle$ a hledáme spojitou náhodnou veličinu X , postupujeme takto:

- Na základě veličiny Y vyřešíme následující rovnici :

$$\int_a^x f(x)dx = Y \quad (29)$$

Tím získáme transformační vztah $X = g(Y)$. Tato rovnice je implicitní pro veličinu X a nemusí mít analytické řešení. V případě, že analytické řešení neexistuje, tak nelze metodu inverzní funkce použít a je nutné využít jiné metody.



Obrázek 10: Metoda výběru pro generování náhodné veličiny X .

Metoda výběru (nebo-li von Neumannova)

Předpokládejme, že máme rozehrát náhodnou veličinu s hustotou prsti $f(x)$ definovanou na intervalu $\langle a, b \rangle$ (viz obr. 10).

Postup:

- Zvolíme konstantu H tak, aby platilo:

$$f(x) \leq H, x \in \langle a, b \rangle \quad (30)$$

Hodnotu H je nutno zvolit co nejmenší, abychom dosáhli co největší účinnosti. Tzn., že hodnota H musí být co nejblíže:

$$H = \sup f(x) \text{ pro } a \leq x \leq b \quad (31)$$

- Nagenerujeme dvě hodnoty náhodné veličiny γ , tj. γ_1, γ_2 a poté vytvoříme čísla x_1, y_1 .

$$\begin{aligned}x_1 &= a + \gamma_1(b - a) \\ y_1 &= H\gamma_2\end{aligned}\tag{32}$$

- Když bod P o souřadnicích (x_1, y_1) bude ležet pod křivkou $y = f(x)$, tj. bude-li $y_1 < f(x_1)$, zvolíme $X = x_1$.
- Když nebude podmínka splněna, tak nagenrujeme novou dvojici γ_1 a γ_2 a postup opakujeme.

Specializované algoritmy pro transformaci náhodných veličin

Tyto algoritmy jsou pro řešení některých problémů efektivnější než algoritmy obecné. V literatuře jich lze nalézt velké množství. Zpravidla se používají při řešení problémů se speciálně definovanou hustotou pravděpodobnosti. My si uvedeme následující:

- Rozehrávání gaussovské náhodné veličiny X.
- Modelování n - rozměrného náhodného bodu.
- Rozehrávání náhodného bodu v kouli o poloměru R.

Rozehrávání gaussovské náhodné veličiny X

Rozehráváme náhodnou veličinu X s parametry $N(0,1)$, tj:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x^2}{2}\right)}, x \in (-\infty, +\infty)\tag{33}$$

Algoritmů pro rozehrávání gaussovské náhodné veličiny je více, my si uvedeme dva.

a) Zjednodušený postup

Hodnotu náhodné veličiny X získáme jako součet k hodnot rovnoměrně rozdělené veličiny γ .

Postup:

- Vytvoříme veličinu X' :

$$X' = \sum_{i=1}^k \gamma_i\tag{34}$$

Vytvořená veličina X' bude mít parametry:

$$\begin{aligned}\mu &= \frac{k}{2} \\ \sigma &= \frac{1}{2} \sqrt{\frac{k}{3}}\end{aligned}\tag{35}$$

- Poté přejdeme na požadovanou normalizovanou veličinu X . To provedeme pomocí následujícího vztahu:

$$X = \frac{X' - \mu}{\sigma}\tag{36}$$

Tento algoritmus je poměrně rychlý, ale s nižší přesností. Dostatečné přesnosti se dosáhne pro $k = 12$, pro $k = 5$ získáme veličinu X s nízkou přesností viz [3].

b) Přesný postup

Když požadujeme přesnou gaussovskou veličinu, tak používáme tento vztah:

$$X = \sqrt{-2 \ln \gamma_1} \cos(2\pi\gamma_2)\tag{37}$$

γ_1, γ_2 - jsou hodnoty rovnoměrně rozdělené náhodné veličiny γ .

Modelování n - rozměrného náhodného bodu

Tato metoda je vhodná pro generování bodů ze složitějšího tvaru. Postupujeme tímto způsobem:

- Zvolíme co nejjednodušší oblast, ve které bude zadaná oblast obsažena.
- Generujeme body v této oblasti a testujeme, padnou-li takto určené body do původní oblasti.
- Body, které padnou do původní oblasti použijeme dále ve výpočtu, ostatní hodnoty odstraníme z paměti.

Rozehrávání náhodného bodu v kouli o poloměru R

Ve sférických souřadnicích (r, θ, φ) jsou transformační vztahy získané s použitím tří hodnot náhodné veličiny γ :

$$\begin{aligned}
r &= R^3 \sqrt{\gamma_1} \\
\cos \theta &= 2\gamma_2 - 1 \\
\varphi &= 2\pi\gamma_3
\end{aligned}
\tag{38}$$

Pomocí druhého a třetího vztahu rozehráváme náhodný směr viz [3].

4.5 Efektivnost metody Monte Carlo

Chyba výpočtu metodou Monte Carlo závisí na charakteristice studované náhodné veličiny (na jejím rozptylu $D(X)$) a na počtu pokusů. Vztah pro chybu metody Monte Carlo je

$$\vartheta = 0,67 \sqrt{\frac{D(x)}{n}}
\tag{39}$$

n - počet pokusů

Rychlost konvergence zvýšíme tím, že zvolíme model s menší disperzí (rozptylem) $D(X)$.

4.6 Transportní problém (průchod částic prostředím)

Transportní problém je nejrozšířenějším problémem částicového modelování ve fyzice. Studuje průchod částic hmotným prostředím. Na transportní problém lze převést velké množství nejrůznějších fyzikálních úloh, jako je například

- pohyb elektronů a iontů v plazmatu,
- průchod záření látkou,
- pohyb elektronů v polovodičích, ...

Při použití metody Monte Carlo není nutné znát detailně celý průběh transportu částic. Před začátkem modelování určíme experimentálně nebo teoreticky důležité fyzikální procesy, ke kterým při transportu dochází a jejich rozdělovací zákony. Při vlastní simulaci pak dochází postupně k rozehrávání jednotlivých hodnot těchto náhodných

procesů. Započítáním vlivů těchto procesů určíme konkrétní realizaci studovaného problému.

Uvedeným postupem můžeme naši představu o reálném procesu srovnat s experimentem. Proto se tomuto postupu také říká **metoda počítačového experimentu**.

Při provádění metody počítačového experimentu můžeme postupovat dvěma způsoby. Při prvním způsobu nejprve analyzujeme celý studovaný fyzikální jev a snažíme se vytvořit model, který bude co nejjednodušší pro výpočet a který bude co nejuvěrněji napodobovat skutečnost. Takové modely jsou ovšem často neefektivní, jejich řešení zabere více času. Těmto modelům se říká **přirozené** (viz 4.6.1). Při druhém způsobu provádění metody počítačového experimentu vytváříme modely se zvýšenou účinností. Těmto modelům se říká **umělé** - viz kapitola 4.6.2.

4.6.1 Modelování pomocí přirozených modelů

Při počítačovém modelování ukládáme data do datových struktur. Tam jsou například ukládány údaje o jednotlivých částicích, jako je rychlost částice, její prostorové souřadnice, typ či hmotnost. Při částicovém modelování rozeznáváme dva typy algoritmů: **jednočásticové** a **mnohočásticové**. Odlišují se v počtu částic (v rámci modelu) přítomných v pracovní oblasti.

Jednočásticová metoda

Když je rozhodující pouze interakce částic s prostředím a ne vzájemná interakce částic mezi sebou, zjednodušujeme si model tím, že při simulaci vypouštíme částice postupně. Teprve když jedna částice urazí celou svou trajektorii, tak vypustíme do pracovní oblasti další. Při použití jednočásticové metody pracujeme s malou datovou strukturou.

Mnohočásticová metoda

Když nesledujeme jen interakci částic s prostředím, ale i vzájemnou interakci mezi částicemi a nebo jen interakci mezi částicemi, musíme sledovat chování velkého souboru částic. Při použití mnohočásticové metody zpravidla pracujeme s velkou datovou strukturou.

Stochastická metoda Monte Carlo je jednočásticová, deterministická metoda molekulární dynamiky je mnohočásticová. V současnosti mají algoritmy částicového

modelování rysy jak deterministické tak i stochastické, a proto i metoda Monte Carlo může být mnohočásticová.

Pracovní oblast

Při použití částicového modelování musíme nejprve určit pracovní oblast v níž budeme studovaný proces simulovat. V případě metody Monte Carlo bude pracovní oblast tvořena třemi částmi: zdrojová oblast, vlastní pracovní oblast a cílová oblast.

Zdrojová oblast je generátor částic. Odtud vstupují částice do pracovní oblasti.

Pracovní oblast je část prostoru, kde dochází k interakcím mezi procházejícími částicemi a látkovým prostředím.

Cílová oblast je oblast, kde končí trajektorie částice. Trajektorie však nemusí skončit pouze v cílové oblasti, ale může skončit i návratem částice do zdrojové oblasti a nebo jejím záchytem.

Procesy v látkovém prostředí

Při průchodu částic látkovým prostředím dochází v látce k tzv. **rozptylovým procesům**. Tyto procesy charakterizujeme středními volnými dráhami nebo účinnými průřezy jednotlivých typů interakcí. **Střední volná dráha** λ je průměrná vzdálenost mezi dvěma srážkami stejného typu, udává se v metrech. **Účinný průřez** S je dán vzorcem:

$$S_i = \frac{1}{X_i} \quad (40)$$

udává se v m^{-1} .

Další veličinou, se kterou se často setkáváme je **náhodná volná dráha** X . Je to vzdálenost, kterou urazí částice mezi dvěma po sobě následujícími interakcemi. Náhodná volná dráha je náhodnou veličinou. Mezi náhodnou volnou dráhou a střední volnou dráhou platí následující vztah:

$$\lambda = E(X) \doteq \frac{1}{n} \sum_{i=1}^n X_i \quad (41)$$

Po uražení náhodné volné dráhy dojde určitě k interakci. Je třeba rozhodnout, která interakce to bude. Při tom se využijí hodnoty středních volných drah dílčích

interakcí λ_i (nebo účinných průřezů S_i) k nalezení pravděpodobností dílčích interakcí v daném místě a při dané energii částice. Pro pravděpodobnost výskytu i -té interakce použijeme jeden z následujících vztahů:

$$p_i = \frac{\lambda}{\lambda_i} \tag{42}$$

$$p_i = \frac{S_i}{S}$$

Když tyto pravděpodobnosti známe, tak použijeme standardní postup na rozehrávání diskrétní náhodné veličiny (obr. 9) viz [3].

Základní druhy rozptylových procesů jsou následující:

□ **Nepružný rozptyl**

Nepružný rozptyl je pro částici vždy spojen se ztrátou energie E . Tato ztráta bývá náhodnou veličinou. Směr, který má částice po interakci, je obvykle zadán v experimentu a nebo ho získáme ze vztahů (38).

□ **Pružný rozptyl**

Při pružném rozptylu se nemění celková energie interagujících částic. To znamená, že energie soustav částic před i po srážce zůstane stejná. Ke změně energie mezi interagujícími částicemi obecně může dojít. Pokud chceme počítat zcela přesně, tak se při srážce musí tato změna v energetické bilanci částice započítat. Směr, který má částice po interakci je buď zadán z experimentu a nebo se předpokládá úhlově izotropní rozptyl, kde jsou směrové kosíny rozehrávány ze vztahů (38).

□ **Štěpení**

Štěpení je proces, kdy z jedné studované částice (primární částice) vznikají dvě nebo i více sekundárních. Příkladem je třeba štěpení jader atomů neutrony, při nichž dojde v důsledku interakce neutronu s těžkými jádry atomů k rozštěpení. Při tomto procesu dochází tedy k štěpení trajektorie. Tzn., že z jedné stopy původní částice se vytvoří „stromy“. Studovaný jev začne být mnohočásticový. Neefektivní by bylo simulovat všechny interakce, které vytvářejí „strom“ a pak je postupně analyzovat. Pro lepší efektivitu byly navrženy tyto postupy:

Analýza po generacích

Částice, které vzniknou štěpením, zařadíme do generací. Tzn., že původní částice bude v generaci I. Z původní částice vzniknou nové částice, např. dvě a tyto dvě částice budou zařazeny do generace II, atd. Generace, v kterých částice zanikly se vymažou. Při analyzování „stromu“ po generacích postupujeme tak, že nejprve zpracujeme všechny větve jedné generace a pak přejdeme na generaci následující a provedeme totéž. Analýza po generacích se používá tam, kde je srážek hodně, např. řetězová reakce.

Analýza po větvích

Částice, které vzniknou štěpením, čísujeme. První částice, která vletne do pracovní oblasti bude mít číslo 1. Ta se při srážce rozštěpí a vzniknou nové částice, např. dvě. Ty budou mít čísla 2 a 3. My budeme částici 1 sledovat, dokud neopustí pracovní oblast. Ta bude stále vytvářet nové částice. My ovšem nejsme schopni zjistit, která částice je nová a která je původní. Řídíme se proto tím, že původní částice se vždy odklání jedním směrem. Strom analyzujeme postupně po větvích. Nejprve zanalyzujeme (zaznamenáme parametry) částici 1 a poté odstraníme větev až k předchozímu uzlu. Tzn., že pokud nám částice 1 vytvořila např. 20 nových částic, tak po analyzaci částice 1 odstraníme její větev až k předchozímu uzlu, kde se větví větev 20 a větev 1. Tímto způsobem zanalyzujeme celý soubor částic. Analýza po větvích se používá tam, kde je málo srážkových procesů vedoucích na štěpení trajektorie.

□ Záchyt

Záchytem se myslí různé mechanismy. Např. neutron je absorbován jádrem a část energie je vyzářena, záchycení elektronu v zakázaném pásu dielektrika. V modelu je záchyt řešen tak, že model obsahuje procesy popisující záchyt částice a následné uvolnění energie nebo náboje. Pokud to ale studovaný jev nevyžaduje, je záchyt realizován tak, že trajektorie částice v daném bodě prostě končí a ze zdroje se vypustí další částice.

Trajektorie částice

Z experimentu získáme informace o četnosti rozptylových procesů, jejich počtu a vlivu na procházející částice. Poté tyto údaje zpracujeme do jedné makroskopické veličiny, kterou bývá výsledná střední volná dráha nebo výsledný účinný průřez. Tím charakterizujeme kvantitativně průměrný vliv prostředí na procházející částice.

Využijeme makroskopickou veličinu a na jejím základě vytvoříme **náhodnou volnou dráhu**. Po uražení náhodné volné dráhy X se částice srazí a trajektorie změní. Poté určíme druh konkrétní interakce a podle jeho druhu změníme rychlost částice. Výše uvedený postup opakujeme a tím získáme celou trajektorii částice.

Generování náhodných volných drah

Vztah pro generování náhodných volných drah lze odvodit za předpokladu, že nezáleží na dílčích parametrech částice (hlavně energii) a že střední volná dráha je konstantní (tzn. látkové prostředí je homogenní). Za těchto předpokladů je vztah, kterým generujeme náhodné volné drahy X_i následující:

$$X_i = -\lambda \ln \gamma_i \quad (43)$$

γ - je rovnoměrně rozdělená náhodná veličina na intervalu $(0, 1)$, nula nemůže být mezi generovanými hodnotami zastoupena.

Pokud nejsou výše uvedené předpoklady splněny nelze vztah (43) použít. Místo něj použijeme umělého obratu, který je popsán v následující kapitole.

4.6.2 Modelování pomocí umělých modelů

Jedná se o modelování fyzikálních procesů se zvýšenou účinností. Umělý obrat vede k vytvoření náhodného procesu s menším rozptylem. Lze ho tedy charakterizovat tím, že přesnost simulace vyměníme za její rychlost. Tento proces však nepopisuje dostatečně dobře studovaný jev. Abychom tento problém vykompenzovali, musí být součástí umělého obratu postup, který z nesprávného výsledku dostane správný.

Druhy umělých obrátů

□ Statistické váhy

Zrychlí konvergenci (sbíhavost) studovaného procesu. Statistická váha w představuje další parametr částice, ale nemá přímý fyzikální význam.

Předpokladem modelu je, že do pracovní oblasti nevstoupí jedna částice, ale rovnou celý „svazek (balík)“ w_0 částic a že se při srážkách svazek částic nerozpadne. Místo celého svazku částic budeme mluvit o jedné částici, která má statistickou hmotnost w . Na počátku transportu má částice statistickou hmotnost $w_0 = 1$ a v průběhu transportu statistická hmotnost dále klesá. Statistická váha se zpravidla nenechává libovolně zmenšovat. Udává se pro ni limitní hodnota, pod kterou se již částice nesleduje a trajektorie se dále nemodeluje (protože se to nevyplatí), tzn., že dojde k ukončení trajektorie.

Tento postup je efektivnější než přirozená metoda, protože stačí pracovat s menším počtem částic, řádově s 1000. Manipulace s počty částic ve svazku sice zpomaluje, ale výhoda v úspoře počtu částic výrazně převáží.

□ Metoda nulové srážky

Této metodě se říká tak proto, že nemění směr pohybu interagující částice a ani energii této částice. Při řešení transportu částic má tato metoda výsadní postavení. Pro rozehrávání náhodné volné dráhy lze i v nehomogenních prostředích využívat jednoduchý předpis (43).

Při formulaci modelu budeme pracovat s prostředím obsahujícím $k+1$ typů srážkových procesů charakterizovaných účinnými průřezy S_1, \dots, S_k, S_{k+1} . Srážkový proces $k+1$ jsme si vymysleli, proto si můžeme navrhnout i jeho závislost na nějakém fyzikálním parametru, který způsobuje nekonstantnost původního účinného průřezu. Závislost $k+1$ srážkového procesu vytvoříme tak, aby celkový nový účinný průřez $S' = S_1 + S_2 + \dots + S_k + S_{k+1}$ byl konstantní. Pak i upravená střední volná dráha λ' bude konstantní a pro rozehrávání náhodné volné dráhy X'_i použijeme tento vztah:

$$X'_i = -\lambda' \ln \gamma_i \quad (44)$$

□ Metoda podobných trajektorií

Tato metoda se uplatňuje při řešení geometricky podobných úloh. Vyřešíme pouze jednu z nich a poté v dalších úlohách přetransformujeme trajektorii. To tedy

znamená, že změním náhodné volné dráhy X a zavedeme váhové faktory částic, které budou závislé na změně měřítka úlohy.

□ **Metoda rozdělení trajektorie**

Tato metoda slouží ke zpřesnění výpočtu kdekoliv v pracovní oblasti. V pracovním prostoru zavedeme umělou interakci (štěpení trajektorie) a předpokládáme, že po srážce jedné částice o statistické váze w_i vznikne k nových částic o vahách w_i/k . Trajektorie těchto nových částic budeme dále studovat odděleně. Předpokládá se vznik většího množství nových částic k , kterých bývá obvykle 10^2 až 10^3 .

□ **Metoda ukončení trajektorie**

Je metodou inverzní k metodě rozdělení trajektorie. Postup je následující: pokud je váha částice příliš malá, rozehrajeme fiktivní srážku. Při této srážce částice s pravděpodobností $1/k$ zvětší svojí váhu k krát a částice s pravděpodobností $1-1/k$ zanikne.

Výše uvedené umělé obraty nejsou jediné, které existují. Byla jich navržena celá řada a v jednotlivých oblastech počítačové fyziky vznikají stále nové.

4.7 Ukázky řešení numerických úloh

Metoda Monte Carlo se uplatňuje i v matematice, kde byly její algoritmy zpracovány pro různé oblasti. Příkladem může být výpočet jednoduchých i násobných integrálů, operace s maticemi, výpočet soustav lineárních algebraických rovnic, výpočet diferenciálních i integrálních rovnic,

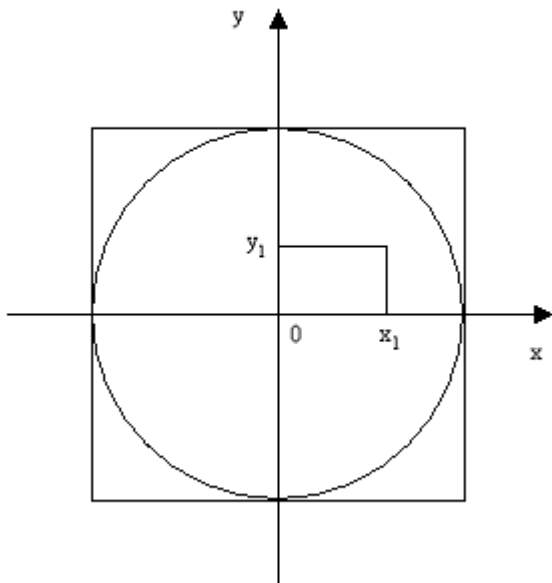
4.7.1 Výpočet čísla π

Metodu Monte Carlo lze s úspěchem použít i k hledání hodnoty čísla π . Do čtverce o stranách rovnoběžných s osami souřadné soustavy vepíšeme kruh (viz obr. 11). Počátek souřadné soustavy leží ve středu kruhu. Každý bod uvnitř čtverce je dán souřadnicemi $[x_i, y_i]$, přičemž $x_i, y_i \in \langle -1, 1 \rangle$ je náhodné číslo. Podíl počtu čísel ležících uvnitř kruhu N_{kruh} a všech čísel generovaných uvnitř čtverce N_{celkem} je roven podílu obsahu kruhu S_{kruh} a obsahu čtverce $S_{\text{čtverec}}$. To je vyjádřeno následujícím vzorcem viz [10]:

$$\frac{N_{\text{kruh}}}{N_{\text{celkem}}} = \frac{S_{\text{kruh}}}{S_{\text{čtverec}}} = \frac{\pi r^2}{S_{\text{čtverec}}} \quad (45)$$

Ze vzorce (45) vyplývá pro hodnotu π :

$$\pi = \frac{N_{\text{kruh}} S_{\text{čtverec}}}{r^2 N_{\text{celkem}}} \quad (46)$$



Obrázek 11: Výpočet čísla π . Překresleno z [10].

4.7.2 Výpočet jednorozměrného určitého integrálu

Budeme určovat určitý integrál I z funkce $f(x)$.

$$I = \int_a^b f(x) dx \quad (47)$$

Ukážeme si dvě metody:

a) **Výpočet střední hodnoty funkce f_s** - princip metody je v tom, že se pokoušíme nalézt střední hodnotu integrované funkce v zadaných mezích. Pokud se nám to podaří, tak hodnotu integrálu určíme podle vztahu:

$$I = (b - a) f_s \quad (48)$$

Pokud se nám to nepodaří, tak k nalezení střední hodnoty f_s použijeme následujícího principu:

Vezmeme náhodnou veličinu X rovnoměrně rozdělenou na intervalu $\langle a, b \rangle$ a zavedeme náhodnou veličinu $\eta = f(X)$, jejíž matematické očekávání $E \eta$ je rovno průměrné hodnotě

funkce $f(x)$ na intervalu $\langle a, b \rangle$. Hodnotu E_η odhadneme pomocí aritmetického průměru n nezávislých realizací veličiny η . Potom bude hodnota integrálu I rovna:

$$I \doteq (b-a) \frac{1}{n} \sum_{i=1}^n f(X_i) \quad (49)$$

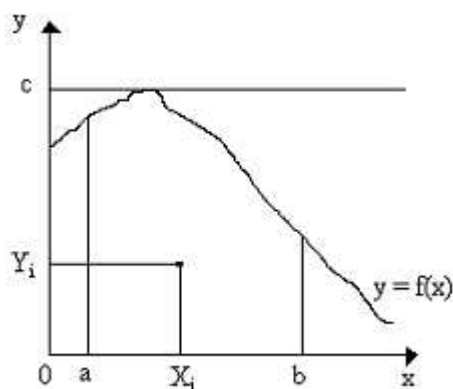
viz [3].

b) Geometrická metoda - princip této metody je v tom, že se vychází z geometrického významu integrálu, tj. integrál je plocha pod křivkou $y = f(x)$ ohraničena mezemi $\langle a, b \rangle$ viz obr. 12. Předpokladem je, že funkce $f(x)$ je na celém intervalu omezená.

Postup:

- 1) Vezmeme dvě náhodné veličiny. Veličinu X z intervalu $\langle a, b \rangle$ a veličinu Y z intervalu $\langle 0, c \rangle$. Obě veličiny jsou rovnoměrně rozdělené na příslušných intervalech.
- 2) Vytvoříme n náhodných bodů (X, Y) .
- 3) Podmínkou $Y_i < f(X_i)$ budeme určovat kolik z nich padne pod křivku $y = f(x)$.
- 4) Počet bodů, které se vejdou pod křivku označíme n' .
- 5) Hodnotu integrálu I určíme takto:

$$I \doteq c(b-a) \frac{n'}{n} \quad (50)$$



Obrázek 12 : Geometrická metoda

Přesnost uvedných metod

Srovnání přesnosti ukazuje, že metoda střední hodnoty má zpravidla menší rozptyl než metoda geometrická, a proto bývá obvykle přesnější.

Byly navrženy různé metody vylepšující přesnost, protože metoda Monte Carlo konverguje pomaleji. Přesnost výpočtu se zvyšuje zmenšováním rozptylu $D(X)$. Rozptyl můžeme zmenšit různými metodami např:

a) **Metoda symetrizace integrované funkce** - integrovanou funkci budeme upravovat tak, aby se na integračním intervalu měnila co nejméně, protože rozptyl je tím menší, čím je změna integrované funkce pomalejší.

b) **Nalezení hlavní části** - princip je v tom, že integrál rozdělíme na dvě části. Část první bude tzv. hlavní a bude představovat hlavní část výsledné hodnoty. Část druhá bude upřesňovat výsledek.

$$I = \int_a^b f(x)dx = \int_a^b g(x)dx + \int_a^b h(x)dx \quad (51)$$

Poté pro výpočty integrálů pomocí metody střední hodnoty a geometrické metody budeme muset vztahy (49) a (50) modifikovat.

4.7.3 Výpočet dvojného určitého integrálu

Dvojný integrál je zobecněním určitého integrálu funkce jedné proměnné na funkci dvou proměnných. Představuje objem vymezený souřadnicovými plochami, které omezují integrační oblast a plochou tvořenou danou funkcí.

$$Z = \int_a^b \int_c^d f(x, y)dx dy \quad (52)$$

Obě metody použité pro jednorozměrný integrál lze zobecnit na vícerozměrný integrál bez výrazné ztráty účinnosti.

Závěr

Metoda molekulární dynamiky a metoda Monte Carlo jsou nejvíce používanými metodami při vytváření počítačových modelů. Počítačové modelování těmito metodami patří v dnešní době k nepostradatelnému nástroji určenému ke studiu nejrůznějších problémů fyziky, chemie, biologie, meteorologie a dalších technických i netechnických věd na částicové (mikroskopické) úrovni.

Počítačové modelování dovoluje získat pohled na mikrosvět, který je klasickým experimentem neproveditelný. Umožňuje provádět simulace, které nelze z nejrůznějších důvodů nahradit laboratorními pokusy. A to buď proto že jsou pokusy v laboratoři prostě neproveditelné (pokusy za nízkých teplot, vysokých tlaků) a nebo příliš drahé (biologické, medicínské, chemické pokusy).

Uvedený text se snažil na 51 stránkách stručně a výstižně popsat počítačové modelování a jeho v dnešní době dvě nejpoužívanější metody, tj. metodu molekulární dynamiky a metodu Monte Carlo.

Použitá literatura:

- [1] Encyklopedie, http://cs.wikipedia.org/wiki/Počítačová_simulace, staženo 5. 3. 2007.
- [2] Křivý, J., Kindler, E. Simulace a modelování, Ostravská univerzita, Ostrava, 2001.
- [3] Hrach, R. Počítačová fyzika I, PF UJEP, Ústí nad Labem, 2003.
- [4] Počítačová simulace konfekční výroby, http://www.kod.vslib.cz/ucebni_materialy/PSI/Skripta_PSI-5.pdf, staženo 8. 2. 2007.
- [5] Vicher, M. Numerická matematika, http://alpha.ujep.cz/~mlisal/nm_2/vicher_nm1.pdf, staženo 25. 9. 2007.
- [6] Boucnik, P. Simulace mikrostruktury s ohledem na dosažení požadovaných vlastností odlitku, VUT, Brno, 2001.
- [7] Metoda hraničních prvků, http://home.zcu.cz/~karban/teaching/mmem/dolezel/course_1/LECT05.pdf, staženo 5. 10. 2007.
- [8] Ercolessi, F. A molecular dynamics primer, International School for Advanced Studies, Trieste Italy, 1997.
- [9] Macháček, J. Molekulová dynamika alkalicko-křemičitých skel, VŠCHT Praha, 2001.
- [10] Bartoš, P., Tesař, J. Metoda Monte Carlo a programovací jazyk MATLAB při přípravě učitelů na pedagogických fakultách, http://dsp.vscht.cz/konference/matlab/MATLAB06/prispevky/tesar_bartos/tesar_bartos.pdf, staženo 3. 2. 2007.

Internetové stránky a prezentace jsou uloženy na CD archivovaném fakultou.