PA081: Programování numerických výpočtů

6. Simulace dynamických dějů

jaro 2019

Něco, co se hýbe

- objekty reálného světa kolem nás
  - velká část počítačových her
- vesmírná tělesa
- plyn a kapaliny
  - předpověď počasí
  - simulace požárů, povodní, ...
- elektrický proud
- interakce atomů a molekul

Jak se to hýbe

- fyzikální zákony
  - postihují principy chování reálného světa
  - jejich dodržení v simulaci dodává důvěryhodnost

Jak se to hýbe

- fyzikální zákony
  - postihují principy chování reálného světa
  - jejich dodržení v simulaci dodává důvěryhodnost
- Newtonovy
  1. setrvačnosti
  2. síly – $F = ma$
  3. akce a reakce
- termodynamické
  1. *Celková energie izolovaného systému zůstává konstantní.*
  2. *Entropie izolovaného systému nikdy neklesá.*

2. termodynamický zákon

- energie má snahu rovnoměrně se rozptýlit.
- psychologické vnímání času je souhrn zkušeností s jeho projevy.
  - pozpátku puštěný film poznáme na první pohled.
  - nelze zcela snadno vyjádřit proč.

2. termodynamický zákon

- energie má snahu rovnoměrně se rozptýlit.
- psychologické vnímání času je souhrn zkušeností s jeho projevy.
  - pozpátku puštěný film poznáme na první pohled.
  - nelze zcela snadno vyjádřit proč.
- dovoluje uchovávat energii (palivo/potrava + kyslík)
  - de-facto podmínka existence života
  - ale také příčina smrti

2. termodynamický zákon

- energie má snahu rovnoměrně se rozptýlit.
- psychologické vnímání času je souhrn zkušeností s jeho projevy.
  - pozpátku puštěný film poznáme na první pohled.
  - nelze zcela snadno vyjádřit proč.
- dovoluje uchovávat energii (palivo/potrava + kyslík)
  - de-facto podmínka existence života
  - ale také příčina smrti
- příčina Murphyho zákonů
- více viz `http://secondlaw.oxy.edu`

Stavba simulace

- stavové proměnné
  - popisují, v jakém stavu se systém nachází
  - poloha, rychlost, ...
- jejich změny v čase
  - jak na sebe stavové proměnné působí vzájemně (rychlost ~ změna polohy)
  - jak se projevují vnější impulsy (síla mění rychlost)
- diferenciální rovnice (obyčejné)
  - vztah stavových proměnných a jejich časových derivací
  - postupným řešením (integrací) získáme vývoj systému v čase

Řešení diferenciálních rovnic

- analytické
  - jen speciální případy
  - i tak komplikované metody
- dopředná (explicitní) Eulerova metoda
  - použita v následujících příkladech
  - v praxi téměř nepoužitelná (viz dále)
- zpětná (implicitní) Eulerova metoda
- semiimplicitní metody
- komplexnější metody

Padající kulička

- stavové proměnné: výška z, rychlost pádu v
- změna výšky odpovídá rychlosti: $dz = -v\,dt$
- změna rychlosti – gravitační síla: $dv = g\,dt$
- korektní zápis systému rovnic

$$dz/dt = -v \qquad dv/dt = g$$

- v kódu přímočará náhrada dt konečným krokem

```
float   z = 2, v = 0;
float   dt = 0.001, g = 9.81;

while (...) {
    z += -v * dt;
    v += g * dt;
    ...
}
```

Pružinový oscilátor

- hmotný bod na pružině, pohyb jen v jednom směru
  - vše ostatní zanedbáno
- změna polohy y odpovídá rychlosti, tj. hybnost/hmotnost

$$dy = \frac{P}{m} dt$$

- změna hybnosti odpovídá působící síle, ta je úměrná vychýlení pružiny

$$dP = F = -ky\,dt$$

Pružinový oscilátor

```
float   k = 1.3, m = 0.2, p = 1, y = 0, t = 0,
        dt = 0.001;

while (...) {
    float p1 = p - k*y*dt,
          y1 = y + p/m*dt;

    t += dt;
    p = p1; y = y1;
}
```

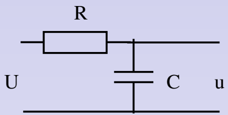
Průlet komety

- poloha $\mathbf{x}$, nenulová počáteční rychlost $\mathbf{v}$
  - pro zjednodušení vektory ve 2D
- změna polohy odpovídá rychlosti: $d\mathbf{x} = \mathbf{v}dt$
- změna rychlosti dána gravitačním působením Slunce:

$$d\mathbf{v} = -G\frac{\mathbf{x}}{\|\mathbf{x}\|}\frac{m_k m_S}{\|\mathbf{x}\|^2}dt$$

Jednoduchý elektrický obvod

- vstupním napětím U nabíjíme kondenzátor přes odpor



- výstupní napětí je úměrné kapacitě a náboji: $u = q/C$
- změna náboje je velikost proudu: $dq/dt = i$
- proud podle Ohmova zákona: $i = (U - u)/R$

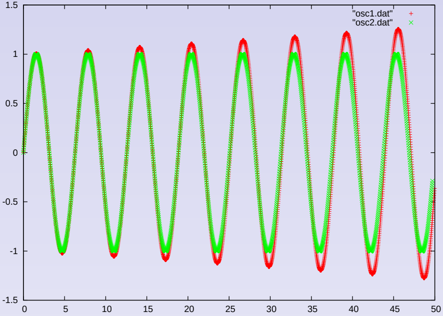
```
float u = 0, U = 14, R = 22e3, C = 1e-6, dt = 0.00001;

while (...) {
    float dq = (U-u)/R * dt;
    u += dq/C;
}
```

- stejně i pro proměnlivý vstup, např. $U = \sin(2\pi \cdot 10^{-3} \cdot t$

Problémy dopředné metody

- nejvíce se projeví na pružinovém oscilátoru



$\Delta t = 0.0001$ (zelená), $\Delta t = 0.01$ (červená)

Problémy dopředné metody

- řešíme rovnici (resp. systém rovnic)

$$\frac{dy}{dt} = f(t, y(t))$$

- počítáme posloupnost

$$y_{n+1} = y_n + \Delta t \left.\frac{dy}{dt}\right|_{t_n}$$

- výsledná $\sin x$ je ideálně nešikovná funkce
  - pro $x \in [0, \pi]$ je $\sin x > 0$ a je konkávní, pro $x \in [\pi, 2\pi]$ je $\sin x < 0$ a je konvexní
  - použitá derivace vždy nadhodnocuje
  - výpočet do systému "přidává energii"

Zpětná Eulerova metoda

- použijeme derivaci až v bodě t_{n+1}

$$y_{n+1} = y_n + \Delta t \left. \frac{dy}{dt} \right|_{t_{n+1}}$$

- dosazením vede na řešení rovnice

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1}))$$

- implicitní metoda
  - pro výpočet dalšího kroku musíme řešit rovnici nebo systém rovnic
  - obecně nemusí být triviální

Zpětná Eulerova metoda

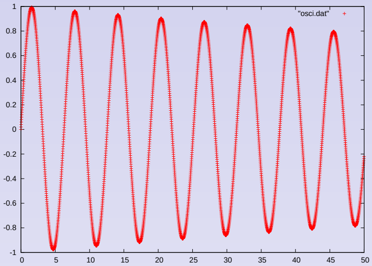
- pro pružinový oscilátor

$$y_{n+1} = y_n + \frac{\Delta t}{m} p_{n+1}$$
$$p_{n+1} = p_n - \Delta t k y_{n+1}$$

- jednoduchý systém lineárních rovnic
- řešení lze vyjádřit analyticky a naprogramovat

Zpětná Eulerova metoda

- výstup pro $\Delta t = 0.01$



- metoda také není numericky stabilní
- má sklon systém tlumit
  - pro simulace v principu vyhovuje
  - nutné řešení systému rovnic v každém kroku může být problém

Semiimplicitní metoda

- kompromis mezi dopřednou (explicitní) a zpětnou (implicitní)
- nevyžaduje řešení komplikovaného systému rovnic
- nahrazuje ho vhodnou aproximací

Semiimplicitní metoda

- kompromis mezi dopřednou (explicitní) a zpětnou (implicitní)
- nevyžaduje řešení komplikovaného systému rovnic
- nahrazuje ho vhodnou aproximací
- časovou derivaci $d\mathbf{y}/dt$ chápeme jako funkci $\mathbf{f}(\mathbf{y})$
- dosadíme ji do zpětné metody

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \mathbf{f}(\mathbf{y}_{n+1})$$

- použijeme Taylorův rozvoj

$$\mathbf{f}(\mathbf{y}_{n+1}) = \mathbf{f}(\mathbf{y}_n) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\mathbf{y}_n} (\mathbf{y}_{n+1} - \mathbf{y}_n) + \cdots$$

- zanedbáme vyšší členy

$$\left(I - \Delta t \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\mathbf{y}_n} \right) (\mathbf{y}_{n+1} - \mathbf{y}_n) = \Delta t \mathbf{f}(\mathbf{y}_n)$$

Kometa semiimplicitní metodou

- stavový vektor $\mathbf{y} = (\mathbf{x}, \mathbf{v})^T$
- pohybová rovnice

$$\mathbf{f}(\mathbf{y}) = \frac{d\mathbf{y}}{dt} = \begin{pmatrix} d\mathbf{x}/dt \\ d\mathbf{v}/dt \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{g} \end{pmatrix}$$

kde $\mathbf{g} = -Gm_k m_S \mathbf{x}/r^3$ a $r = \|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$

- Jakobián pro Taylorův rozvoj

$$\frac{\partial \mathbf{f}}{\partial \mathbf{y}} = \begin{pmatrix} \partial \mathbf{v}/\partial \mathbf{x} & \partial \mathbf{v}/\partial \mathbf{v} \\ \partial \mathbf{g}/\partial \mathbf{x} & \partial \mathbf{g}/\partial \mathbf{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \partial \mathbf{g}/\partial \mathbf{x} & 0 \end{pmatrix}$$

- výpočet $\partial \mathbf{g}/\partial \mathbf{x}$ podle pravidel o derivování složené funkce

Kometa semiimplicitní metodou

- v čase n máme přímo k dispozici hodnoty stavových proměnných $\mathbf{y} = (\mathbf{x}, \mathbf{v})^T$
- na základě pohybové rovnice $d\mathbf{y}/dt = \ldots$ (zahrnuje i externí silové působení) spočteme jejich derivace, tj. $\mathbf{f}(\mathbf{y}_n)$
- podle předem spočteného vzorce získáme $\partial \mathbf{g}/\partial \mathbf{x}$ v čase n
- dosadíme do rovnice

$$\left(I - \Delta t \left. \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right|_{\mathbf{y}_n} \right) (\mathbf{y}_{n+1} - \mathbf{y}_n) = \Delta t \mathbf{f}(\mathbf{y}_n)$$

- řešením je $\Delta y = \mathbf{y}_{n+1} - \mathbf{y}_n$
- narozdíl od zpětné metody je to jen systém 4 lineárních rovnic

Žábo, skoč

- metoda „leap frog“
- specializovaná pro systémy tvaru

$$d\mathbf{x}/dt = \mathbf{v} \qquad d\mathbf{v}/dt = \mathbf{F}(\mathbf{x})$$

- v sudém kroku integrujeme pozici, v lichém rychlost

$$\mathbf{x}_{2n} = \mathbf{x}_{2n-2} + \Delta t \mathbf{v}_{2n-1}$$
$$\mathbf{v}_{2n+1} = \mathbf{v}_{2n-1} + \Delta t \mathbf{F}(\mathbf{x}_{2n})$$

- pro harmonické systémy je numericky stabilní
  - nepřidává energii
  - chyba dopředné metody v sudém kroku je kompenzována chybou v lichém kroku
- v praxi velmi rozšířená metoda

Metody vyšších řádů

- obecná formule metody 1. řádu

$$y_{n+1} = y_n + hf(x_n, y_n)$$

  pro $f(x_n, y_n) = \left.\frac{dy}{dx}\right|_{x_n}$ je to Eulerova dopředná
- lze ukázat, že její chyba je $O(h^2)$
  - Taylorovým rozovjem hypoteticky známeho výsledku
  - o 1 vyšší exponent kroku, proto 1. řád
- metody vyšších řádů mají chybu $O(h^{k+1})$

Ilustrativní metoda 2. řádu

- začneme stejně

$$k_1 = hf(x_n, y_n)$$

- pokusný mezikrok do poloviny intervalu h

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)$$

- definitivní krok s použitím výsledků mezikroku

$$y_{n+1} = y_n + k_2$$

- lze ukázat, že chyba je $O(h^3)$
- k posunu o h potřebujeme $2\times$ vyhodnotit f
- lepší než metoda 1. řádu s polovičním krokem

Runge-Kutta

- metoda 4. řádu
- posloupnost výpočtu

$$
\begin{aligned}
k_1 &= hf(x_n, y_n) \\
k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\
k_3 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \\
k_4 &= hf(x_n + h, y_n + k_3) \\
y_{n+1} &= y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4
\end{aligned}
$$

Pokročilé metody

- Která metoda je nejlepší?

Pokročilé metody

- Která metoda je nejlepší?
  - jak kdy, záleží na konkrétním problému
  - neplatí, že čím vyšší řád, tím delší krok si můžeme dovolit

Pokročilé metody

- Která metoda je nejlepší?
  - jak kdy, záleží na konkrétním problému
  - neplatí, že čím vyšší řád, tím delší krok si můžeme dovolit
- adaptivní délka kroku
  - dlouhé kroky v plochých oblastech, krátké v bohatších
  - zdvojnásobení kroku metody Runge-Kutta
  - Fehlberguv algoritmus
  - zpětnovazební metody
  - …
- Bulirsch-Stoer
  - dlouhý krok s vnitřní extrapolací na nekonečně krátké mezikroky
- více viz Numerical Recipes

Výpočet integrálu

- neurčitý integrál
$$f(t) = \int g(t)dt$$
odpovídá nalezení libovolného řešení f rovnice
$$df(t)/dt = g(t)$$
- určitý integrál
$$y = \int_{t_0}^{t_1} g(t)dt = f(t_1) - f(t_0)$$
odpovídá konkrétnímu řešení téže rovnice s počáteční podmínkou $f(t_0) = 0$
- položíme $y_0 = 0$ a vhodnou numerickou metodou integrujeme od t_0 do t_1

Dynamika pevného tělesa

- hmotný bod je příliš zjednodušující abstrakce
- pevné těleso
  - nenulový objem, nepodléhá žádným deformacím
- k popsání stavu systému jsou nutné další veličiny
- aktuální orientace (natočení)
  - matice 3×3 nebo kvaternion
- úhlová rychlost
  - vektor ω
  - velikost – rychlost rotace
  - směr – osa rotace
- matice (tenzor) setrvačnosti
  - analogie hmotnosti
  - prostorové rozložení hmotnosti

Matice setrvačnosti a úhlová hybnost

- lineární hybnost $\mathbf{P} = m\mathbf{v}$ je relativně intuitivní veličina
- úhlová hybnost $\mathbf{L} = \mathbf{M}\omega$ už ne
  - jednoduché vyjádření pohybové rovnice

$$d\mathbf{L}(t)/dt = \mathbf{T}(t)$$

  - analogie $d\mathbf{P}(t)/dt = \mathbf{F}(t)$
  - pro volně rotující těleso zůstává $\mathbf{L}$ konstantní (ω ne)
- matice setrvačnosti
  - vychází z $\mathbf{T} = \mathbf{F} \times \mathbf{r}$
  - pro množinu diskrétních bodů

$$\mathbf{M} = \sum \begin{pmatrix} m(r_y^2 + r_z^2) & -mr_xr_y & -mr_xr_z \\ -mr_yr_x & m(r_x^2 + r_z^2) & -mr_yr_z \\ -mr_zr_x & -mr_zr_y & m(r_x^2 + r_y^2) \end{pmatrix}$$

  - ve spojitém případě analogicky integrováním
  - závisí na aktuální orientaci
- více viz `http://www.cs.cmu.edu/~baraff/sigcourse/`

Pohybové rovnice pevného tělesa

- stavový vektor $\mathbf{y} = (\mathbf{x}, \mathbf{q}, \mathbf{P}, \mathbf{L})^T$
  - $\mathbf{x}$ – poloha
  - $\mathbf{q}$ – orientace
  - $\mathbf{P}$ – lineární hybnost (vyjadřuje rychlost)
  - $\mathbf{L}$ – úhlová hybnost (vyjadřuje úhlovou rychlost)
- rovnice

$$\frac{d\mathbf{y}}{dt} = \begin{pmatrix} \frac{1}{m}\mathbf{P} \\ \frac{1}{2}\omega_q \mathbf{q} \\ \mathbf{F} \\ \mathbf{T} \end{pmatrix}$$

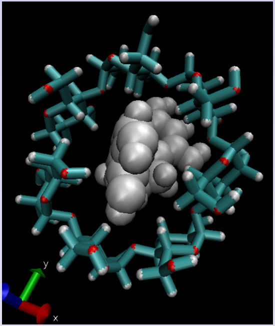
- úhlovou rychlost ω vypočteme

$$\omega = \mathbf{q}(\mathbf{M}^{-1}\mathbf{q}^{-1}(\mathbf{L}))$$

- ω_q je vnoření ω do prostoru kvaternionů
- integrace dopřednou nebo semiimplicitní metodou
  - lineární rovnice o 13 proměnných

Model interakce molekul

- dvě molekuly
  - pevná, velká – receptor
  - menší, pohyblivá – ligand
- pohyblivou molekulu ovládáme
  - myš a klávesnice,
  - zařízení silové zpětné vazby
- molekuly silově interagují
  - přitažlivé i odpudivé síly
  - každý atom s každým
- význam
  - nalezení místa, kde si ligand „sedne“
  - pochopení vlastností interakce v okolí
- simulace musí působit důvěryhodně
  - co nejdokonalejší iluze manipulace s reálnými objekty



Zařízení silové zpětné vazby

- vyvinula se z teleoperátorů
  - manipulace na dálku, např. v nebezpečném prostředí
  - manipulace s virtuálním prostředím
- impedanční (odporové)
  - zařízení snímá polohu, případně rychlost
  - působí vypočtenou silou
  - modelovaný systém je vyjádřen jako „odpor“ – silová reakce systému na změnu polohy
- admitanční (vodivostní)
  - zařízení snímá silové působení
  - přesune se do vypočtené polohy
  - modelovaný systém je vyjádřen jako „vodivost“ – ochota systému reagovat na vnější silové působení

Zařízení silové zpětné vazby

- v ideálním případě lze modelovat vše obojím
  - vstupují nedokonalosti zařízení
  - prakticky jsou impedanční zařízení lepší v modelu, kde převládá aktivita operátora

Zařízení silové zpětné vazby

- v ideálním případě lze modelovat vše obojím
  - vstupují nedokonalosti zařízení
  - prakticky jsou impedanční zařízení lepší v modelu, kde převládá aktivita operátora
- hmat má výrazně nižší latenci než zrak
- nutná obnovovací frekvence 1 kHz
- vysoká náročnost na numerické vyhodnocení modelu

Přímé mapování síly

- model je de facto statický
  - vstup (souřadnice zařízení) jsou přímo promítnuty do modelu
  - na jejich základě je vypočtena silová interakce
  - síla je opět přímo aplikována na zařízení
- např. kolize dvou koulí, modelujeme tvrdou pružinou

$$F = \begin{cases} 0 & r \geq R_1 + R_2 \\ k(R_1 + R_2 - r) & r < R_1 + R_2 \end{cases}$$

Přímé mapování síly

- model je de facto statický
  - vstup (souřadnice zařízení) jsou přímo promítnuty do modelu
  - na jejich základě je vypočtena silová interakce
  - síla je opět přímo aplikována na zařízení
- např. kolize dvou koulí, modelujeme tvrdou pružinou

$$F = \begin{cases} 0 & r \geq R_1 + R_2 \\ k(R_1 + R_2 - r) & r < R_1 + R_2 \end{cases}$$

- jednoduché, přehledné, ale takhle reálný svět nefunguje
  - veškeré dynamické chování (tj. jaký je efekt aplikované síly) jsme z modelu vyloučili
  - zejména nedokážeme modelovat různé hmotnosti kolidujících objektů
  - dynamicky se chová fyzické zařízení atd., ale to nestačí
- nezatracujeme úplně, na některé aplikace stačí

Virtuální spojení

- „virtual coupling“
- manipulovaný objekt je k zařízení připojen viskoelastickým spojem
- paralelní pružina a tlumič

$$F = k\Delta\mathbf{x} + b\Delta\mathbf{v}$$

- silová zpětná vazba
  - jakou silou má zařízení působit na operátora
- vstup pro model
  - jak působí operátor na manipulovaný virtuální objekt
- tj. 3. Newtonův zákon v praxi

Silové interakce molekul

- molekulu ligandu modelujeme jako pevné těleso
  - vzájemná poloha atomů se nemění
  - hmotnosti atomů koncentrovány v jádrech → matice setrvačnosti
  - silová interakce atomu vždy působí v místě jádra

Silové interakce molekul

- molekulu ligandu modelujeme jako pevné těleso
  - vzájemná poloha atomů se nemění
  - hmotnosti atomů koncentrovány v jádrech → matice setrvačnosti
  - silová interakce atomu vždy působí v místě jádra
- van der Waalsova – LJ potenciál

$$F_{LJ} = -\frac{12A}{r^{13}} + \frac{6B}{r^7}$$

- elektrostatická

$$F_{el} = \epsilon \frac{q_1 q_2}{r^2}$$

Silové interakce molekul

- molekulu ligandu modelujeme jako pevné těleso
  - vzájemná poloha atomů se nemění
  - hmotnosti atomů koncentrovány v jádrech → matice setrvačnosti
  - silová interakce atomu vždy působí v místě jádra
- van der Waalsova – LJ potenciál

$$F_{LJ} = -\frac{12A}{r^{13}} + \frac{6B}{r^7}$$

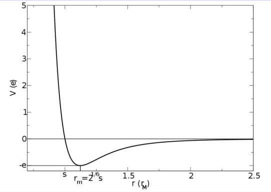
- elektrostatická

$$F_{el} = \epsilon \frac{q_1 q_2}{r^2}$$

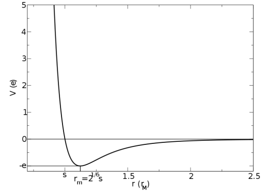
- $n \approx 10^4$ atomů receptoru, $m \approx 100$ atomů ligandu
- $m \times n$ interakcí nespočítáme $1000\times$ za vteřinu

van der Waalsova interakce

- předvýpočet potenciálu na mřížce
  - běžně používáno v chemickém software
  - interpolace příliš vyhladí strmé stěny



van der Waalsova interakce



- předvýpočet potenciálu na mřížce
  - běžně používáno v chemickém software
  - interpolace příliš vyhladí strmé stěny
- díky vysokým exponentům relativně krátký dosah
- můžeme si dovolit ořez
  - uvažujeme pouze dvojice atomů o vzájemné vzdálenosti menší než daná konstanta
  - typicky 4–8 Å
- vypočítat $m \times n$ vzdáleností dvojic atomů je pořád příliš

van der Waalsova interakce

- prostor rozdělíme na pravidelné buňky
  - tak, aby se „největší“ atom receptoru vešel do jedné buňky
  - při náhodné pozici zasahuje nanejvýš do 8 buněk
- hashovací funkce souřadnic buňky
- struktura připravená před simulací
  - projdeme všechny atomy receptoru
  - atomy, které padnou (i částečně) do buňky zařadíme do příslušného řádku

van der Waalsova interakce

- prostor rozdělíme na pravidelné buňky
  - tak, aby se „největší“ atom receptoru vešel do jedné buňky
  - při náhodné pozici zasahuje nanejvýš do 8 buněk
- hashovací funkce souřadnic buňky
- struktura připravená před simulací
  - projdeme všechny atomy receptoru
  - atomy, které padnou (i částečně) do buňky zařadíme do příslušného řádku
- atom ligandu uměle zvětšíme o vzdálenost ořezu
- překryje ≈ 10–500 buněk
- uvažujeme pouze atomy receptoru v příslušných řádcích

van der Waalsova interakce

- i po ořezu zůstává příliš mnoho atomů receptoru
- silové působení (síla i točivý moment) jsou aditivní
- první derivace jsou také aditivní
- lze vše počítat paralelně a na závěr sečíst
  - round-robin rozdělení atomů receptoru mezi jádra CPU
  - statisticky rovnoměrné
- technicky používáme MPI
- musí běžet na jednom stroji nebo na velmi rychlé síti
  - vysoké nároky na latenci
  - rozeslání dat, výpočet, sesbírání výsledku za méně než 1ms

Elektrostatická interakce

- nízký exponent (2) – dlouhý dosah
- nelze efektivně použít ořez
- vypočítáme silové pole na mřížce
  - působení na jednotkový náboj
  - lze spočítat dopředu
- při běhu simulace interpolace pro všechny atomy ligandu
  - interpolace ořezává vysoké hodnoty
  - v kombinaci s přesnou VdW interakcí to není problém
  - extrémy jsou ve stejných místech
  - VdW nedovolí tohoto stavu dosáhnout
- mřížka je příliš velká
  - $256^3 \cdot 3 \cdot 4\text{B} \approx 200\text{MB}$
  - adaptivní rozlišení mřížky

Celková struktura simulace

- off-line výpočet
  - elektrostatického pole
  - hashovací tabulka pro VdW interakce
- řídící simulační vlákno (1 kHz)
  - elektrostatické působení na všechny atomy ligandu
  - součet dílčího silového působení VdW sil
  - působení zařízení silové zpětné vazby na ligand (3 nebo 6 DoF)
  - integrace pohybové rovnice (semiimplicitní, systém 13 lineárních rovnic)
- $N\times$ výpočty dílčích VdW sil
  - synchronně s řídícím vláknem
  - pevně mezi sebe rozdělené atomy receptoru
  - každý pro všechny atomy ligandu
- řízení zařízení silové zpětné vazby (1 kHz)
  - asynchronní
  - implementace viskoelastického spojení s ligandem
- grafika (25 Hz)

Pohyblivý receptor

- model tuhé molekuly neodpovídá realitě
  - při nenulové teplotě mění svůj tvar spontánně
  - při vzájemné interakci se ovlivňují a mění tvar
- spontánní pohyby receptoru umíme napočítat dopředu
- ~ 1000 a více snímků
- hash tabulka pro všechny snímky
  - 1000×200 kB není problém
  - neřešíme, i když umíme přepočítávat změny dostatečně rychle
- elektrostatické pole
  - 1000×50 MB, kdo neplýtvá diskem s námi ...
  - komprese proměnlivého vektorového pole

Domácí úkol

Scénář

- do blízkosti planety Jupiter vletí kometa a exploduje
- simulujeme trajektorii úlomků v gravitačním poli planety a čtyř měsíců
- detekujeme kolize úlomků s planetou a měsíci
- zjednodušující předpoklady
  - vše jen ve 2D
  - stabilní kruhový oběh všech měsíců
  - úlomky na sebe vzájemně nepůsobí
  - pro výpočet gravitační síly nahrazujeme tělesa bodem
  - zanedbáváme vliv Slunce
  - žádné relativistické efekty

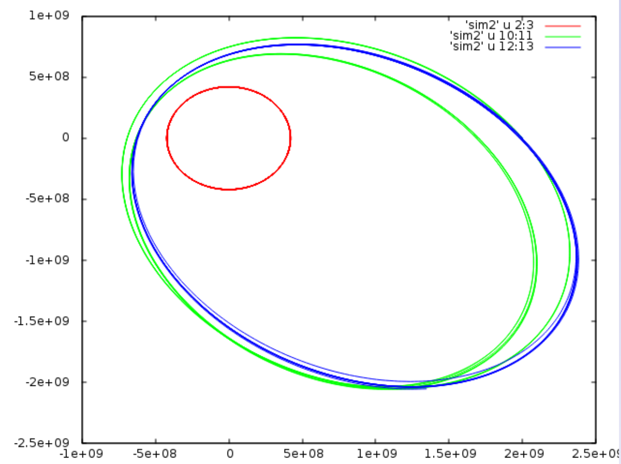
Domácí úkol

Naivní implementace

- k dispozici v materiálech k předmětu
- pouze Eulerova dopředná metoda
  - nepoužitelná, není dostatečně stabilní pro potřebný počet kroků
- parametry programu
  - časový krok (10 s)
  - počet úlomků (generovány deterministicky)
  - maximální délka simulace
  - metoda integrace
  - možnost vypsat polohy všech těles (vizuální kontrola)

Domácí úkol

Grafické znázornění



Domácí úkol

Zadání

- doimplementujte do programu stabilnější metody integrace
- hodnotí se i rychlost implementace
  - základní optimalizace výpočtu
  - vhodně volená metoda integrace, délka kroku apod.
- parametry spuštění
  - počet úlomků, délka simulace
  - tolerance na přesnost výsledku

Domácí úkol

Hodnocení

- 1 bod za funkční implementaci (v toleranci přesnosti)
- 0.5 bodu za komentované optimalizační kroky vedoucí k využití alespoň 10 % teoretického výkonu jádra
- 0.5 za paralelní implementaci s efektivitou na 16 jádrech alespoň 70 %
- připočteno k výsledku zkoušky (písemka 12, na A třeba 9)

Domácí úkol

Optimalizace

- základní
  - použijte AVX instrukce (`-mavx`, případně `-mfma`)
  - umožněte kompilátoru vektorizovat
  - dbejte na využití cache
  - double vs. float
  - vyhněte se zbytečným výpočtům náročných funkcí (sqrt, sin, cos, ...)
- paralelní implementace
  - pthreads, OpenMP, MPI, ...
  - vhodné rozdělení úlomků mezi vlákna
  - nevyužitá jádra po kolizích s měsíci

Domácí úkol

Odhad výkonu

- spočítat přesně operace je problematické
  - netriviální úsilí, snadné zanesení chyby → falešný výsledek
- čitače v procesorech
  - v Linuxu `perf` z balíku `linux-tools-generic`
    `perf stat -e r5307c6 program argumenty ...`
  - magické `r5307c6` funguje pro Haswell, více viz např.
    `http://www.bnikolic.co.uk/blog/hpc-howto-measure-flops.html`
  - ověřte si vztah hodnoty čitače ke skutečnému počtu provedených operací
    - instrukce AVX – 8 operací ve float zároveň
    - i při velmi dobrém využití cache se skutečně provede cca. 70 % instrukcí