

# Popis programu nteles

Petr Šafařík

Petr Zikán

19. června 2010

## 1 Popis algoritmu a jeho implementace

Program počítá úlohu n-těles podle Newtonova gravitačního zákona

$$\mathbf{F}_g = -\kappa \frac{m_1 m_2}{r^3} \cdot \mathbf{r}, \quad (1)$$

s využitím klasických pohybových rovnic v nejjednodušším tvaru

$$\mathbf{v}_{\text{new}} = \mathbf{v}_{\text{old}} + \frac{\mathbf{F}}{m} \cdot \Delta t, \quad (2)$$

$$\mathbf{r}_{\text{new}} = \mathbf{r}_{\text{old}} + \mathbf{v}_{\text{new}} \cdot \Delta t. \quad (3)$$

Tělo algoritmu tvoří dva do sebe vnořené for cykly:

```
for i := 0 to n-1 do
  for j := i+1 to n do
    ...
  end for
end for
```

První cyklus probíhá od 1. tělesa do n, druhý od i+1 do n, kde n je počet těles. S využitím 3. Newtonova zákona (akce a reakce) je spočtená síla (podle (1)) mezi uvažovaným i. a j. tělesem připočtena k celkové síle působící na i. těleso, a pak záporně vzata přičtena k síle na j. těleso. Tím jsou postupně napočítány všechny síly mezi tělesy do pomocného pole `F[i]`. Následně jsou s využitím (2) a (3) aktualizovány polohy v polích `poloha[i]` a rychlost `[i]`.

## 2 Použití, zadání vstupních dat

Program po rozbalení přeložíte prostým příkazem

```
$ make
```

a spustíte:

```
$ ./nteles
```

Po spuštění program prohlédne pracovní adresář po dřívějších výsledcích počítání a nabídne pokračovat od místa, kde minule byl ukončen. Pokud budete pokračovat, naváže. Pokud

chcete spustit nový výpočet, spustí se průvodce, do kterého zadáte množství objektů k počítání, jejich hmotnosti, souřadnice a počáteční rychlosti.

Protože program je psán pokud možno robustně (a přiznáváme, že i jednoduše), nepoužívá dynamicky alokovaná datová pole a ukazatele, ale alokace je pevná. Proto je omezení na maximální počet počítaných objektů nastaveno v hlavičkovém souboru `nteles.h` v `MAXOBJEKTU` a to na `5` (nezapomeňte po změně program znovu přeložit!)

### 3 Členění programu a popis ostatních částí kódu

Program je rozdělen na několik oddělených modulů dle jejich funkcí dle filozofie čistého kódu. I přes to je možné, že nebude dostatečně komentován. Na druhou stranu je natolik jednoduchý, že kód považujeme za sebevysvětlující.

### 4 Animace

Z důvodů názornější (rozhodně ne však dokonalé) prezentace byly vytvořeny dva skripty k jednoduché animaci napočítaných trajektorií. Skripty jsou připravené pro případ dvou a tří těles, nicméně úprava na větší počet těles je poměrně přímočará. Skript využívá program `Gnuplot` a napočítaných dat v souboru `datas.txt`.

První bash-skript `plot.plo` předá počáteční nastavení `Gnuplotu` a pak z něj spustí další jednoduchý skriptík `looper`. Ten využívá dvou prázdných řádků mezi jednotlivými polohami, které `gnuplot` interpretuje jako další datovou dávku s indexem o jedna vyšší než předchozí (viz `Gnuplotu` předaná proměnná `ii` v souboru `plot.plo`). `Gnuplot` pak postupně přes sebe zobrazuje následující polohy. Bohužel však rychlost zobrazování nebude kvůli velkému zatížení procesoru odpovídat relativnímu pohybu těles. Pro právě spočtené hodnoty je možné použít zvláštní cíl v souboru `Makefile` jménem `animate`. Příkazem:

```
$ make animate
```

spustíte vykreslování výše popsaným algoritmem.

### 5 Instalace a distribuce programu

Program je po přeložení možné nainstalovat a odinstalovat klasickými příkazy

```
# make install
```

resp.

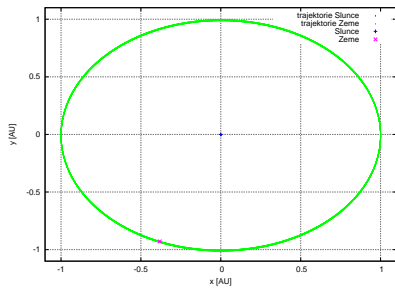
```
# make uninstall
```

Pro další distribuci programu při dodržení Licence (vizte část 7) je možné použít cíl příkazu `make` s názvem `distrib`. Tento cíl vytvoří archiv `nteles.tar.gz`, který bude obsahovat vše potřebné pro další redistribuci programu.

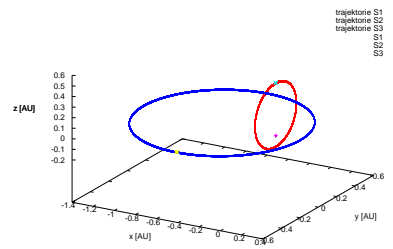
### 6 Ukázky výstupu z programu

Distribuční archiv obsahuje ještě soubor `demos.tar.gz`, ve kterém se nachází dva adresáře: `2telesa` a `3telesa` včetně animačních skriptů `plot.plo`. Výsledky práce programu jsou na obrázcích 1a a 1b pro dvě, resp. tři tělesa z demonstračních souborů.

Ukázka z terminálu při spuštění programu a práci se třemi tělesy:



(a) Numerický výpočet dvou těles



(b) Numerický výpočet tří těles (roviny obíhající těles jsou vzájemně kolmé)

Obrázek 1: Ukázky z výstupu programu nteles

Byl nalezen již existující soubor s definicemi defines.txt a daty datas.txt, což značí, že by mohlo být pokračováno ve výpočtu od místa přerušení. Přejete si použít stávající soubor [1], nebo založit nový? [0]: 1

```
Spouštím funkci init_load_def_file
Otvírám soubor defines.txt
Spouštím funkci init_load_data_file
Otvírám soubor datas.txt
----
```

Jsou předávána tato data:

Hmotnost 1. objektu je 2.000000

Načteny tyto počáteční polohy při výpočtu pohybu 1. objektu:

$R_1(x,y,z) = (0.000543, -0.000019, 0.000000)$

Načtena tato počáteční rychlost objektu 1:

$V_1(x,y,z) = (0.000002, -0.000000, 0.000000)$

----

Hmotnost 2. objektu je 0.000001

Načteny tyto počáteční polohy při výpočtu pohybu 2. objektu:

$R_2(x,y,z) = (0.000515, 0.109371, 0.461899)$

Načtena tato počáteční rychlost objektu 2:

$V_2(x,y,z) = (0.000002, 0.018410, -0.013666)$

----

Hmotnost 3. objektu je 0.000050

Načteny tyto počáteční polohy při výpočtu pohybu 3. objektu:

$R_3(x,y,z) = (-0.331199, 0.552254, -0.000000)$

Načtena tato počáteční rychlost objektu 3:

$V_3(x,y,z) = (-0.031720, 0.007593, -0.000000)$

## 7 Licence

Celý program je licencován pod GPLv3+ a ke stažení ze stránek jednoho z autorů programu <http://physics.muni.cz/~petos/> (hledejte kód F8370).

Celý obsah licence je v souboru LICENCE, který je neoddělitelnou součástí programu.

## 8 Errata

Je známo několik problémů:

- Po spuštění dostanete signál SIGABRT – Jedná se o bug glibc, vyřešíte jej zadáním do terminálu:

```
export MALLOC_CHECK_=0
```

a program spustit znovu.

- Tělesa po doteku odletí do velkých vzdáleností – program nemá ošetřeno spousta věcí a vzájemná kolize těles je jednou z nich. Je nutné upravit počáteční podmínky.
- Další chyb je tam určitě spousta. Nebojte se je opravit sami – budiž explicitně řečeno, že smíte.