

# PHYSICAL BASED ANIMATIONS AND MATHEMATICAL MODELING

## SKUSKA

Autor: Marián Kravec

### Set A

Pri postupe animovanie "Skeleton and skinning" sa najskôr modeluje kostra modelu (zväčša nejakej postavičky) a následne sa na túto kostru modeluje koža. Celý proces má dve časti:

- Rigging Skeleton - ide o proces pri ktorom sa vytvára hierarchická kostra objektu, keďže táto kostra je hierarchická tak okrem informácie ako je pozícia, rotácia a dĺžka kosti určujeme aj informáciu o rodičovskej kosti (výnimkou je koreňová kosť) a detských kostiach (v slovenčine to znie strašne), kosti sa navzájom prepájajú na koncoch kde vytvárajú kĺby pre ktoré sa definujú limity pohybu
- Skinning skeleton - proces pri ktorom sa na kostru aplikuje koža, zväčša definovanú ako sieť bodov (vrcholov), kde pre každý bod (vrchol) definuje množinu váh hovoriacej o príslušnosti daného bodu (vrcholu) k blízkym kostiam

Samotná animácia následne funguje tak, že animujeme iba pohyb kostry a pozície bodov (vrcholov) siete kože je nakoniec prepočítaný na základe novej pozície kostí.

### Set B

$$\begin{aligned}\ddot{\mathbf{r}}_1 &= -Gm_2 \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} - Gm_3 \frac{\mathbf{r}_1 - \mathbf{r}_3}{|\mathbf{r}_1 - \mathbf{r}_3|^3}, \\ \ddot{\mathbf{r}}_2 &= -Gm_3 \frac{\mathbf{r}_2 - \mathbf{r}_3}{|\mathbf{r}_2 - \mathbf{r}_3|^3} - Gm_1 \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|^3}, \\ \ddot{\mathbf{r}}_3 &= -Gm_1 \frac{\mathbf{r}_3 - \mathbf{r}_1}{|\mathbf{r}_3 - \mathbf{r}_1|^3} - Gm_2 \frac{\mathbf{r}_3 - \mathbf{r}_2}{|\mathbf{r}_3 - \mathbf{r}_2|^3}.\end{aligned}$$

1.

a)

Keďže som sa narodil 18.9. tak hodnoty  $MM = 9$  a  $DD = 18$ , ak si za zvyšné hodnoty počiatočných parametrov bodov zvolíme 0 naše body sa na počiatku nachádzajú:

$$\begin{aligned}r_1 &= \{MM, 0, DD\} = \{9, 0, 18\} \\ r_2 &= \{DD, MM, 0\} = \{18, 9, 0\} \\ r_3 &= \{0, DD, MM\} = \{0, 18, 9\}\end{aligned}$$

b)

Keďže nemáme tušenie ako sa systém bude správať hodnoty počiatočných derivácií určíme pomerne náhodne:

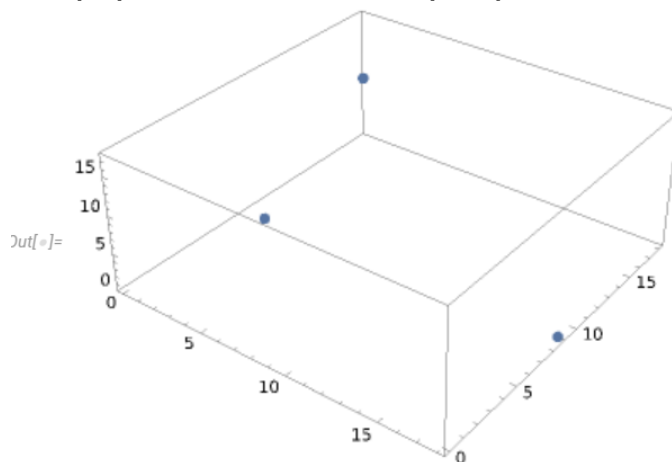
$$\begin{aligned}r'_1 &= \{0.5, 0.1, 0.1\} \\ r'_2 &= \{0.1, 0.5, 0.1\} \\ r'_3 &= \{0.1, 0.1, 0.5\}\end{aligned}$$

(ako bolo spomínané druhé derivácie nepotrebuje určiť počiatočnú podmienku ale nič nám nebráni ich všetky nastaviť na 1 a nikdy nepoužiť)

c)

Ešte posledné konštanty a to sú  $G = 6.674 * 10^{-11}$  a hmotnosti ktoré si na začiatok zadajme takto  $m_1 = 10^1$ ,  $m_2 = 10^1$ ,  $m_3 = 10^1$ .

Ak si naše počiatočné body vykreslíme dostaneme takýto výsledok:



2.

Teraz môžeme začať s samotným modelovaním pohybu troch telies.

Najskôr si zapíšeme rovnice a počiatočné podmienky:

```
{r1'[t] == -G*m2*(r1[t]-r2[t])/(Norm[r1[t]-r2[t]]^3) - G*m3*(r1[t]-r3[t])/(Norm[r1[t]-r3[t]]^3),
 r2'[t] == -G*m3*(r2[t]-r3[t])/(Norm[r2[t]-r3[t]]^3) - G*m1*(r2[t]-r1[t])/(Norm[r2[t]-r1[t]]^3),
 r3'[t] == -G*m1*(r3[t]-r1[t])/(Norm[r3[t]-r1[t]]^3) - G*m2*(r3[t]-r2[t])/(Norm[r3[t]-r2[t]]^3),
 r1[0] == r10, r2[0] == r20, r3[0] == r30,
 r1'[0] == dr10, r2'[0] == dr20, r3'[0] == dr30}
```

a)

A nastavíme metódu na numerické integrovanie na Runge-Kutta metódu (použijeme metódu priamo ponúknutú Wolframom) konkrétne použijeme jej explicitnú verziu.

**Method → "ExplicitRungeKutta"**

b)

Najskôr si zapíšeme konštanty a počiatočné hodnoty:

```

(* Autor Marian Kravec *)
(* Narodil som as 18.9 DD = 18, MM = 09 *)

DD = 18
MM = 09
G = 6.674 * (10^(-11))
r10 = {MM, 0, DD}
r20 = {DD, MM, 0}
r30 = {0, DD, MM}

dr10 = {0.5, 0.1, 0.1}
dr20 = {0.1, 0.5, 0.1}
dr30 = {0.1, 0.1, 0.5}
ddr10 = {1, 1, 1}
ddr20 = {1, 1, 1}
ddr30 = {1, 1, 1}
m1 = 10^11
m2 = 10^11
m3 = 10^11

```

Ako ďalšie rovnice a počiatočné podmienky:

```

{r1''[t] == -G * m2 * (r1[t] - r2[t]) / (Norm[r1[t] - r2[t]]^3) - G * m3 * (r1[t] - r3[t]) / (Norm[r1[t] - r3[t]]^3),
 r2''[t] == -G * m3 * (r2[t] - r3[t]) / (Norm[r2[t] - r3[t]]^3) - G * m1 * (r2[t] - r1[t]) / (Norm[r2[t] - r1[t]]^3),
 r3''[t] == -G * m1 * (r3[t] - r1[t]) / (Norm[r3[t] - r1[t]]^3) - G * m2 * (r3[t] - r2[t]) / (Norm[r3[t] - r2[t]]^3),
 r1[0] == r10, r2[0] == r20, r3[0] == r30,
 r1'[0] == dr10, r2'[0] == dr20, r3'[0] == dr30

```

A cele to poskladáme do funkcie na riešenie diferenciálnych rovníc:

```

interpolated = NDSolve[
{r1''[t] == -G * m2 * (r1[t] - r2[t]) / (Norm[r1[t] - r2[t]]^3) - G * m3 * (r1[t] - r3[t]) / (Norm[r1[t] - r3[t]]^3),
 r2''[t] == -G * m3 * (r2[t] - r3[t]) / (Norm[r2[t] - r3[t]]^3) - G * m1 * (r2[t] - r1[t]) / (Norm[r2[t] - r1[t]]^3),
 r3''[t] == -G * m1 * (r3[t] - r1[t]) / (Norm[r3[t] - r1[t]]^3) - G * m2 * (r3[t] - r2[t]) / (Norm[r3[t] - r2[t]]^3),
 r1[0] == r10, r2[0] == r20, r3[0] == r30,
 r1'[0] == dr10, r2'[0] == dr20, r3'[0] == dr30
},
{r1, r2, r3},
{t, 0, 690},
Method -> "ExplicitRungeKutta"]

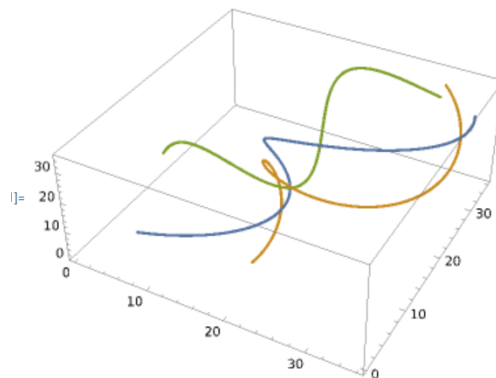
```

c)

Aby sme s najväčšou pravdepodobnosťou videli pohyb budeme interpolovať prvých 100 sekúnd a rovnaký časový úsek vykreslíme v ďalšej časti.

d)

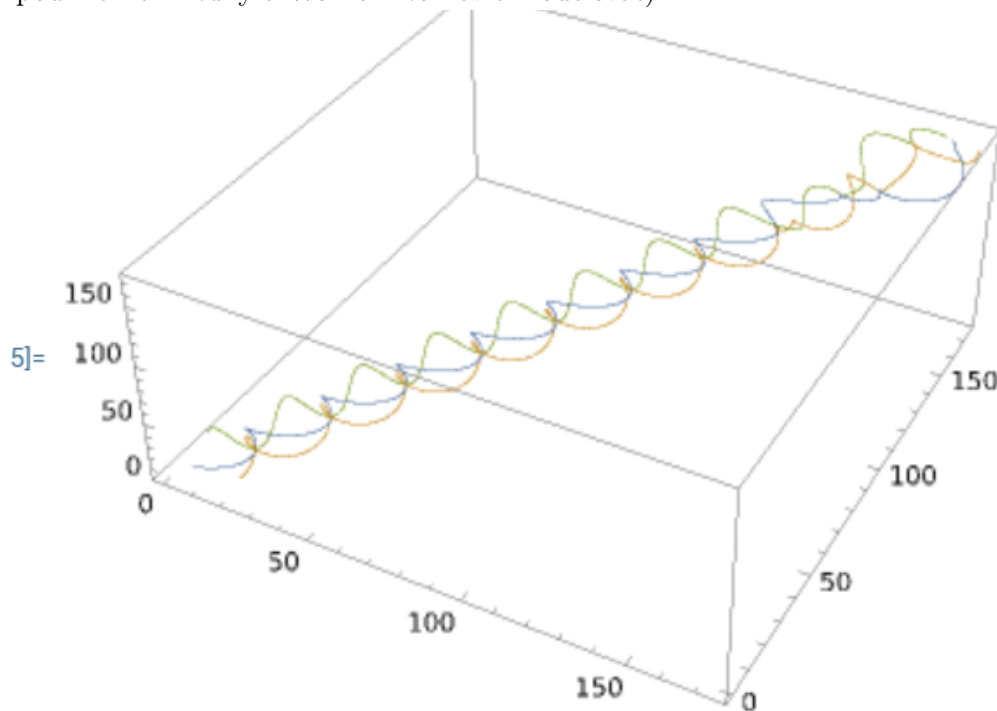
Keď budeme modelovať systém s hodnotami zadanými v časti 1. prvých sto sekúnd a výsledok vykreslíme dostaneme takýto pohyb objektov:



Vidíme peknú interakciu medzi objektami.

e)

Teraz skúsme modelovanie zopakovať ale pre 690 sekúnd (po tomto čase začíname byť náš systém so súčasnými podmienkami tuhý a Wolfram to nevie modelovať):



Tu už lepšie vidíme správanie systému, kde si môžeme všimnúť že telesá idú spoločne špirálovitým pohybom jedným smerom, z počiatku to vyzerá, že ich pohyb by sa mohol považovať za periodický (opakuje sa rovnaký pattern) avšak ten sa kazí na konci, čo môže byť spôsobené buď tým, že systém nie je periodický a iba začiatok sa tak javí alebo tým, že chyba interpolácie je už príliš veľká a model už nezopakuje rovnakú periódu aj napriek tomu, že tam je.

3.

a)

Body umiestnime na trojuholník tak, že im všetkým dáme nulovú z súradnicu, prvý bod pošleme do počiatku súradnicovej sústavy, druhému dáme x-ovú súradnicu rovnú 3 a tretiemu y-ovú rovnú 4:

$$r_1 = \{0, 0, 0\}$$

$$r_2 = \{3, 0, 0\}$$

$$r_3 = \{0, 4, 0\}$$

Hodnoty počiatočných derivácií a hmotností nastavme takto (tieto hodnoty vznikli pomocou metódy pokus omyl aby sa systém nestal tuhým veľmi rýchlo).

$$r'_1 = \{0.001, 0.01, 0.01\}$$

$$r'_2 = \{0.01, 0.001, 0.01\}$$

$$r'_3 = \{0.01, 0.01, 0.001\}$$

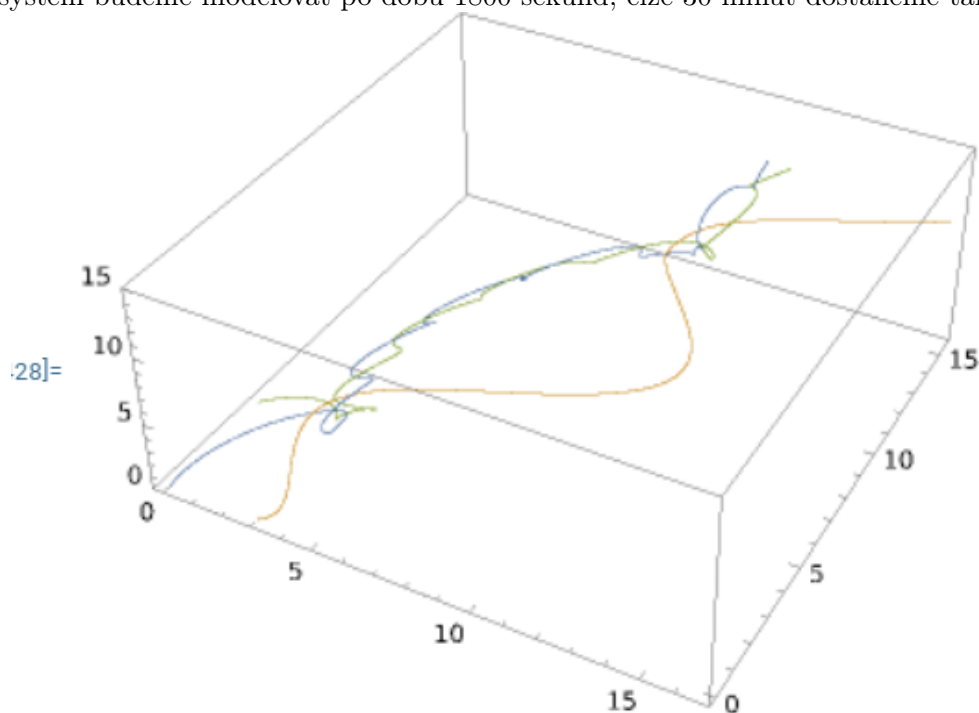
$$m_1 = 5 * 10^7$$

$$m_2 = 4 * 10^7$$

$$m_3 = 3 * 10^7$$

b)

Ak tento systém budeme modelovať po dobu 1800 sekúnd, čiže 30 minút dostaneme takýto pohyb:



Vidíme, že výrazne viac interagujú objekty 1 a 3 a objekt 2 je iba "ťahaná" ich smeromäle z môjho pohľadu to vyzerá na pomerne chaotické správanie.

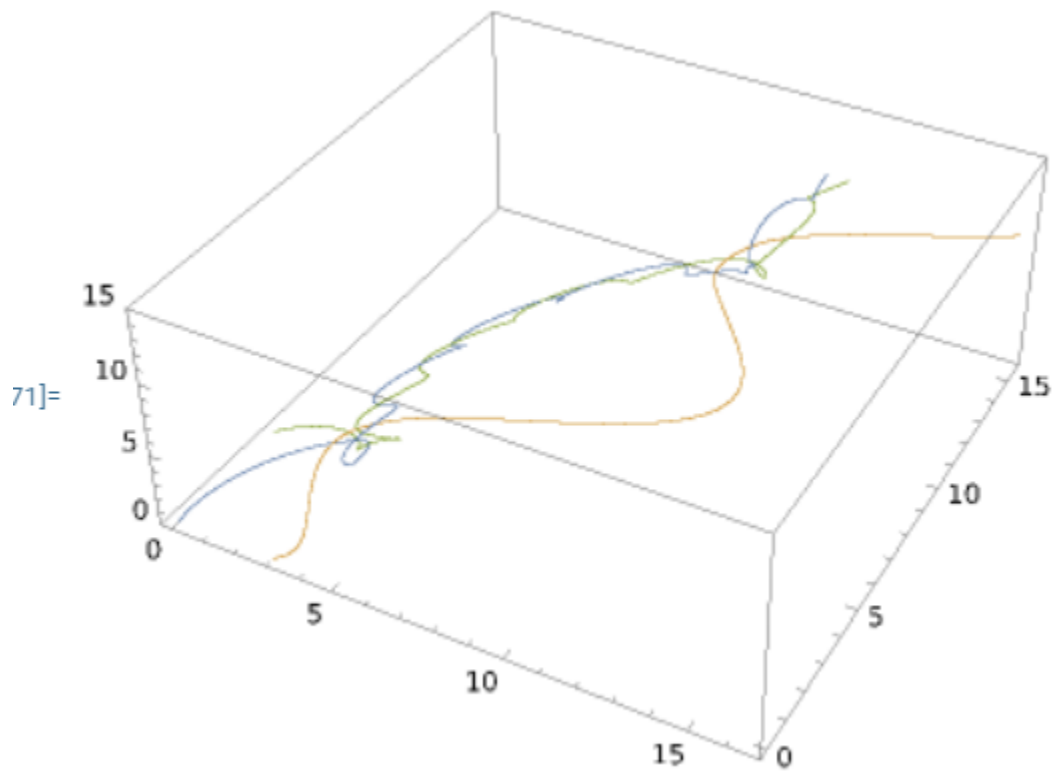
c)

Začnime tým, že budeme robiť malé zmeny v hmotnosti telies (zmena od 0 po  $10^4$  kg čo je pri ich hmotnosti ( $10^7$ ) pomerne mála zmena).

V druhom teste pridáme aj malé zmeny počiatočných derivácií (zmenu o maximálne 0.00001 pre každú hodnotu).

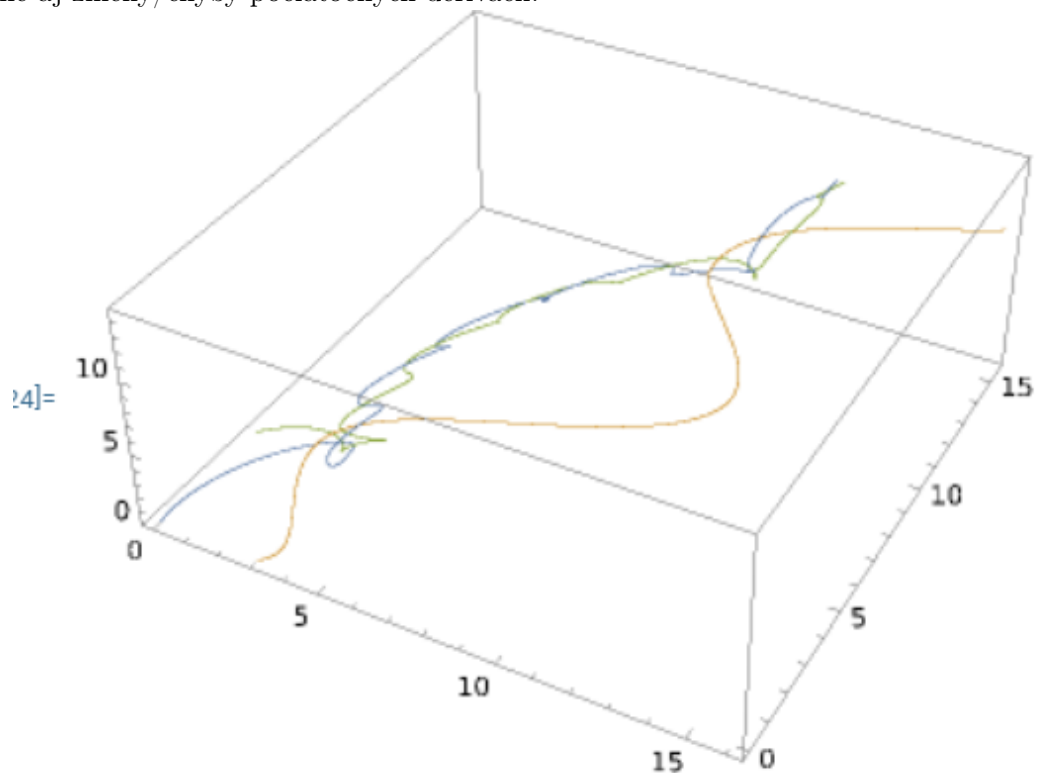
d)

Pri prvom teste zmeny iba hmotností:



Vidíme, že táto zmena nemá žiaden vplyv na pohyb telies, čiže pohyb stabilný pri malých zmenách hmotnosti.

Ak pridáme aj zmeny/chyby počiatočných derivácií:

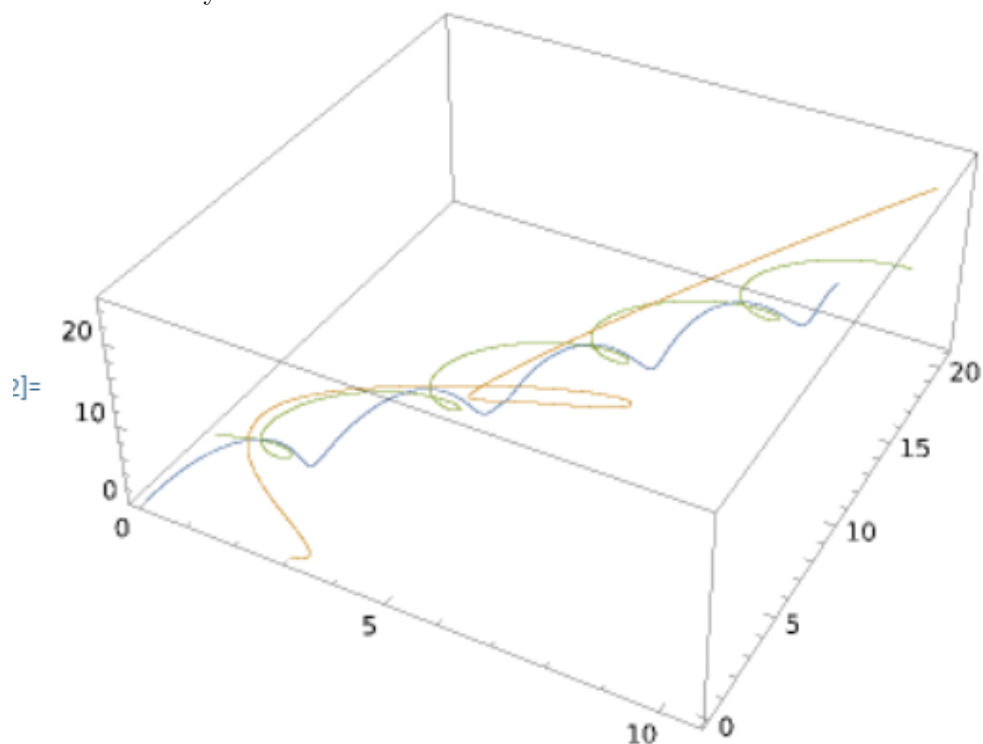


Tu vidíme, že mierne zmeny sú viditeľné ale celkové správanie je stále stabilné.

e)

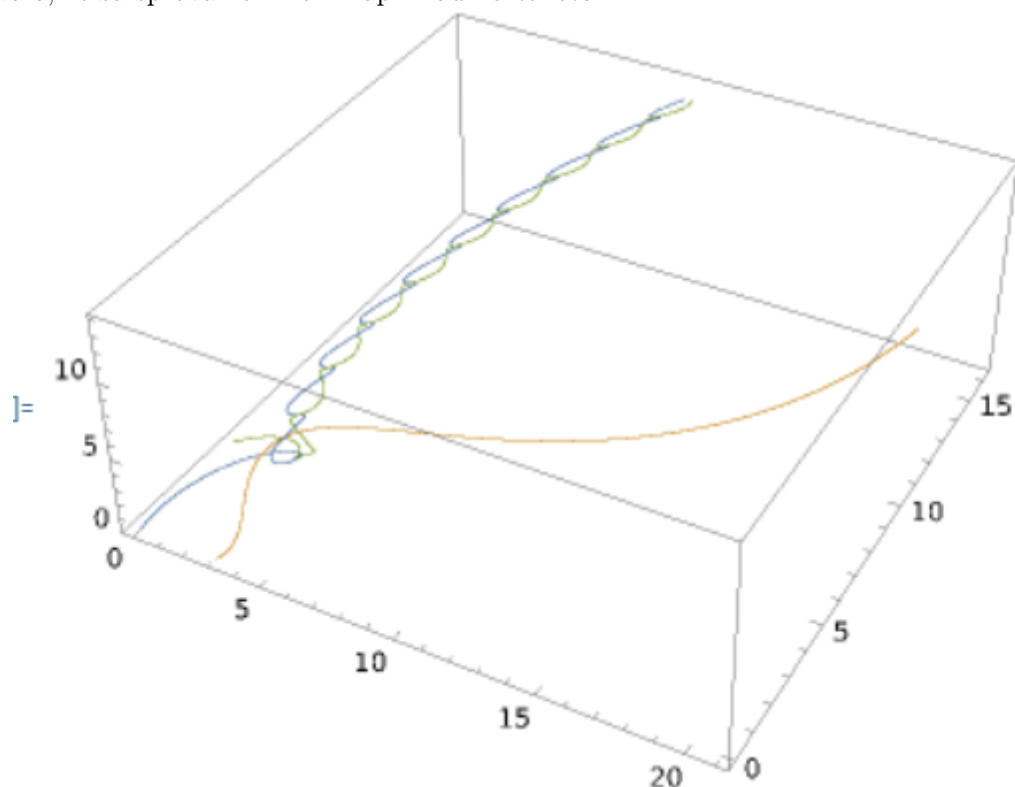
Skúsme, hodnoty chýb zväčšovať kým sa správanie zásadnejšie nezmení.

V prípade hmotnosti sa systém ukazuje ako veľmi stabilný keďže na to aby sa zmenou hmotnosti jeho správanie zmenilo takto výrazne:



Bolo potrebná aby zmena hmotnosti mohla byť až z intervalu od 0 po  $10^7$  kg. Konkrétne v ukážkovom prípade sa hmotnosť objektu 2 znížila len na približne tisícinu pôvodnej hmotnosti čo spôsobilo, že jeho vplyv na objekty 1 a 3 s výrazne väčšou hmotnosťou bola zanedbateľná a preto sa objekty 1 a 3 dostali do periodického pohybu ignorujúc objekt 2.

V prípade zmien počiatočných podmienok tak stačí aby bola zmena z intervalu od 0 po 0.001 na to aby sa bežne stávalo, že sa správanie zmení napríklad na takého:



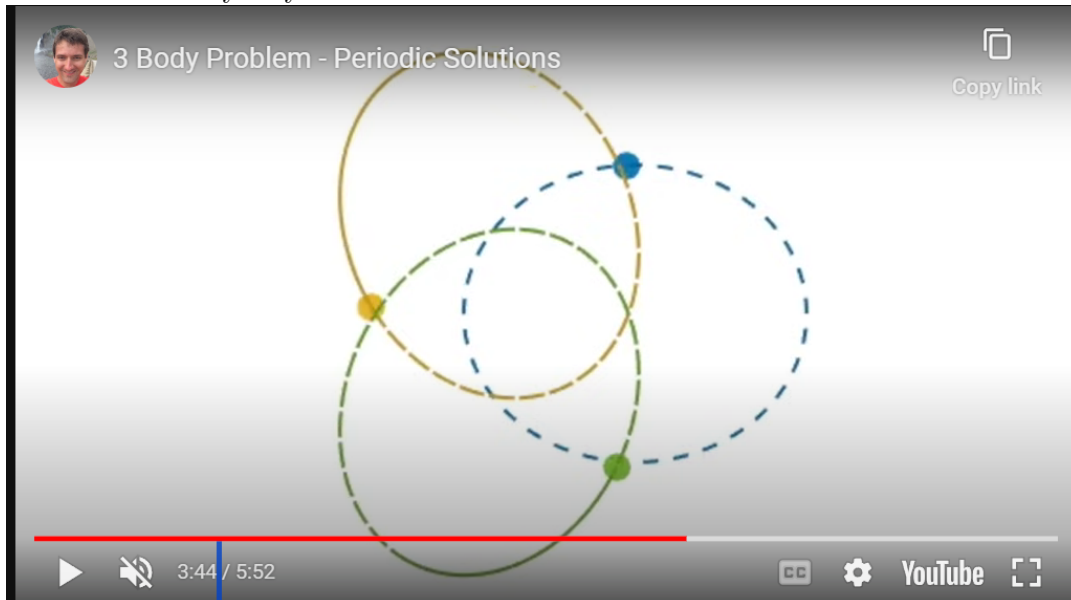
Kde aj napriek tomu, že hmotnosť objektu 2 je porovnateľná s 1 a 3 nastane situácia, že sa tento objekt

dostane príliš ďaleko a minimálne ovplyvňuje zvyšné 2 objekty ktoré sa znovu dostávajú do periodického pohybu.

4.

a)

Snaha bola nasimulovať takýto systém:



Ale kvôli časovým obmedzeniam sa to nepodarilo.

Keď som nastavil parametre následovne:

$$r'_1 = \{0.05 * \sin[\pi/3], -0.05 * \sin[\pi/6], 0\}$$

$$r'_2 = \{0.05 * \sin[\pi/6], 0.05 * \sin[\pi/3], 0\}$$

$$r'_3 = \{-0.05, 0, 0\}$$

$$m_1 = 10^9$$

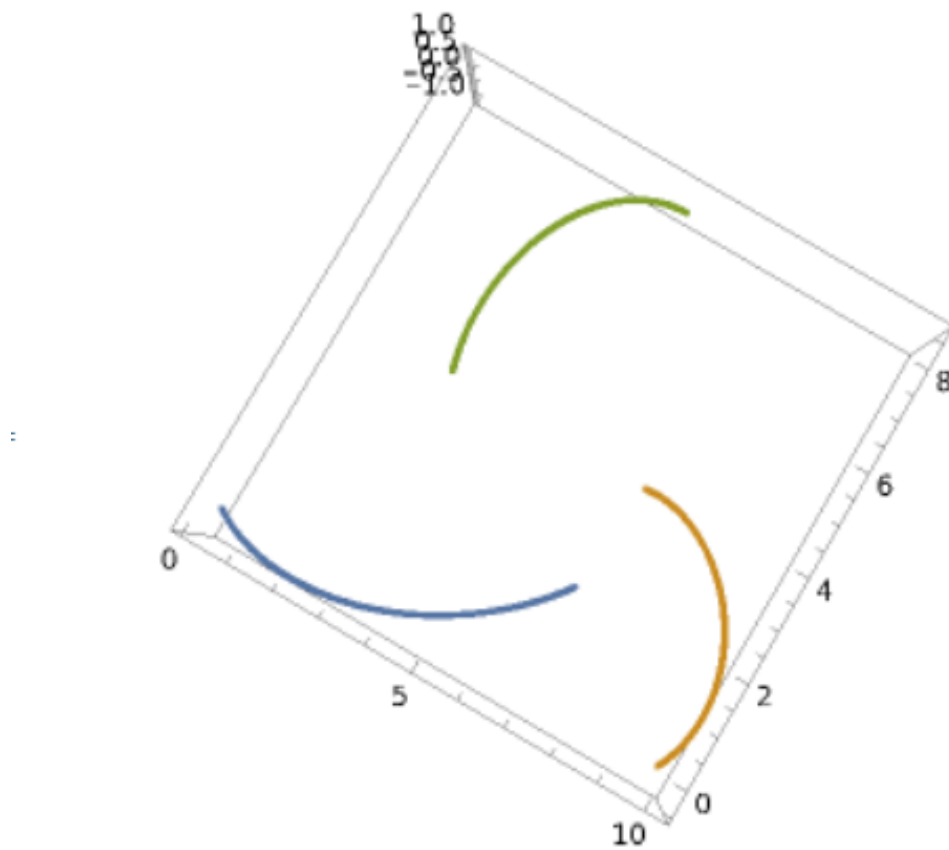
$$m_2 = 10^9$$

$$m_3 = 10^9$$

b)

Tak počiatočné správanie vyzeralo sľubne:





A potom sa to celé rozbilo:

