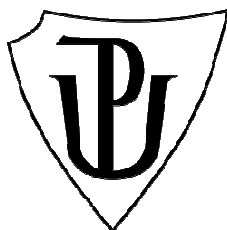


Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra experimentální fyziky

BAKALÁŘSKÁ PRÁCE

Problém tří těles v klasické mechanice



Autor: Jonáš Jarkuliš
Studijní program: B1701 Fyzika
Studijní obor: Obecná fyzika a matematická fyzika
Forma studia: Prezenční
Vedoucí práce: Mgr. Lukáš Richterek, Ph.D.
Termín odevzdání práce: 16.5.2011

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně pod vedením a že jsem použil zdrojů, které cituji a uvádím v seznamu použitých zdrojů.

V Olomouci

.....

Bibliografická identifikace:

Jméno a příjmení autora Jonáš Jarkuliš
Název práce Problém tří těles v klasické mechanice
Typ práce Bakalářská
Pracoviště Katedra experimentální fyziky
Vedoucí práce Mgr. Lukáš Richterek, Ph.D.
Rok obhajoby práce 2011
Abstrakt V rámci bakalářské práce byly vytvořeny simulace ilustrující zákony pohybu tří těles pod vlivem vzájemné gravitace v klasické mechanice.

Klíčová slova Tři tělesa, Lagrangeovy body, Easy Java Simulation, Arenstorfova orbita, Hillova plocha
Počet stran 44
Počet příloh 0
Jazyk Český

Bibliographical identification:

Autor's first name and surname	Jonáš Jarkuliš
Title	Three body problem in classical mechanics
Type of thesis	Bachelor
Department	Department of Experimental Physics
Supervisor	Mgr. Lukáš Richterek, Ph.D.
The year of presentation	2011
Abstract	The bachelor thesis concentrates on design of the computer models illustrating the principles of the three body motion in the classical mechanics.
Keywords	Three body, Lagrange points, Easy Java Simulation, Arenstorf orbit, Hill sphere
Number of pages	44
Number of appendices	0
Language	Czech

Obsah

Úvod	6
1. Historie nebeské mechaniky	7
2. Omezený problém tří těles	8
2.1 Pohybová rovnice tělesa zanedbatelné hmotnosti	9
2.2 Hillova plocha.....	11
2.3 Librační body.....	13
3. Program Easy Java Simulation	17
3.1 První kontakt s Easy Java Simulation.....	17
3.2 Příklad tělesa na pružině v programu Easy Java Simulation.....	19
4. Omezený problém tří těles v <i>EJS</i>	26
4.1 Simulace: Lagrangeovy stabilní a nestabilní body	26
4.2 Simulace: Jacobiho konstanta.....	29
4.3 Simulace: Arenstorfova orbita.....	30
4.4 Vytvoření simulace omezeného problému tří těles v <i>EJS</i>	32
5. Obecný problém tří těles	36
5.1 Obecný problém tří těles v rovině	36
5.2 Obecný problém tří těles v prostoru	38
5.3 Simulace: Gravitační asistence	41
Závěr.....	43
Seznam použitých pramenů	44

Úvod

Cílem bakalářské práce je vytvořit dynamický model problému tří těles v klasické mechanice za použití některého volně šiřitelného softwaru. K vypracování práce jsem si vybral software *Easy Java Simulation*. Program, jenž zatím u nás není příliš rozšířený, jsem si zvolil pro to, že má velice intuitivní ovládání a byl vyvinut primárně právě k vytváření fyzikálních simulací.

Vzniklé simulace by měly sloužit jako studijní materiál studentům kurzu „Přehledu relativistické astrofyziky a kosmologie“ na Přírodovědecké fakultě university Palackého, ale i studentům a ostatním zajímavajícím se o problematiku pohybu kosmických těles. Součástí práce je rovněž teoretický úvod ke studiu problému tří těles a to především tzv. omezeného problému tří těles. Uvedená problematika je rozpracována v řadě pramenů (namátkou uvedme např. [2], [4], [5], [9]), základy byly formulovány již v 18. století *Josephem Louisem Lagrangem*. Záměrem práce bylo proto přehledně shrnout některé důležité pojmy, jakými jsou například Lagrangeovy librační body, Hillova plocha či Jacobiho konstanta, a zejména ilustrovat jejich význam a zákonitosti pohybu těles pod vlivem vzájemné gravitace prostřednictvím vlastních vytvořených simulací.

Jedním z hlavních důvodů, proč jsem si toto téma vybral, je právě dynamické modelování, jehož význam i využití ve fyzice, astrofyzice i kosmonautice jde ruku v ruce se snadnější dostupností a zvyšováním výkonu výpočetní techniky, který stále roste. V případě modelování pohybu kosmických těles uvedených v bakalářské práci tak lze velmi názorně demonstrovat zákony klasické mechaniky, rozdíl mezi pohledem z různých vztažných soustav i některé typické a fyzikálně významné typy pohybů.

Chtěl bych zde poděkovat mému vedoucímu práce Mgr. Lukáši Richterekovi, Ph.D. Jednak za seznámení s programem *EJS*, ale také za nespočet rad, které vedly ke zkvalitnění celé práce, a v neposlední řadě i za množství času, které mě věnoval. Dále pak prof. RNDr. Jiřímu Bajerovi, Csc za přečtení pracovní verze a upozornění na některé nepřesnosti.

1. Historie nebeské mechaniky

Nebeská mechanika je vědní obor ležící na rozhraní mezi astronomií a teoretickou mechanikou zabývající se popisem pohybu kosmických těles a určováním jejich drah. Samotné pojmenování tohoto oboru odráží dobu jejího největšího rozvoje v 17. a 18. Století, z hlediska fyziky jde o aplikaci Newtonových pohybových zákonů a gravitačního zákona na pohyb těles Sluneční soustavy. Historie pozorování vesmírných objektů sahá až do pravěku. Důkazem je prehistorická památka Stonehenge v jižní Anglii, jež měla ve své době (uvádí se, že pochází z období kolem roku 1700 př. n. l., podle [2]), pravděpodobně sloužit k sledování pohybů Měsíce a Slunce. Vzpomeňme na staré Egyptany, kteří podle některých hvězd orientovali své pyramidy. Dalšími průkopníky astronomie byli bezesporu antičtí Řekové. Za všechny jmenujme alespoň *Platóna*, který vytvořil tzv. *ideu pravé astronomie*, jejímž úkolem mělo být vysvětlit pohyb nebeských těles pomocí rovnoměrných kruhových pohybů ([2]). Tato myšlenka se stala na téměř dvě tisíciletí převládajícím názorem. Bylo to právě v době antiky, kdy se poprvé zjistilo, že Slunce je mnohokrát větší než Země a tak se dospělo k heliocentrickému názoru. V období renesance nastává obrovský posun v představách o Vesmíru a v nebeské mechanice. Toto období je spojeno se jmény *Tycho de Brahe*, *Johannes Kepler*, *Galileo Galilei*, *Sir Isaac Newton*, *Mikuláš Koperník* a další. Jedním z velkých objevů té doby bylo sepsání *Keplerových zákonů*.

Období 18. - 19. můžeme nazvat zlatou érou nebeských mechaniků, jakými byli *Jean le Rond d'Allembert*, *Leonhard Euler*, *Pierre Simon de Laplace* a především *Joseph Louis Lagrange* (1736 – 1843). Kromě toho, že odvodil základní rovnice analytické mechaniky a rozpracoval metodu variace konstant, vyřešil i speciální případ problému tří těles, jimž se budeme v tomto textu také podrobněji zabývat.

I dnes, v éře kosmických letů, je velmi důležité umět předpovídat polohu těles sluneční soustavy, plánovat pohyb raket a umělých družic vyslaných k planetám a jejich měsícům nebo rozmísťovaných v okolí Země. Zajímavou kapitolou je sledování tzv. nera-Earth objects (NEO), u nichž dochází k významnému přiblížení k Zemi, je proto nutné pečlivě stanovit jejich trajektorie a vyhodnocovat pravděpodobnost případné srážky. I když jde o problémy, jež lze popsat v rámci klasické fyziky, jsou nejenom zajímavé, ale důležité z praktického hlediska.

2. Omezený problém tří těles

Nejjednodušším úkolem nebeské mechaniky je problém dvou těles, který má analytické řešení. Tento problém omezující se na dvě kulovitě symetrická tělesa je dobrým modelem interakce dvou těles (samozřejmě ve vesmíru se izolovaná soustava dvou těles nevyskytuje, ale přesto je tento model použitelný, je-li vliv ostatních těles na soustavu nepatrný a lze v uvažovaném kontextu ho zanedbat). Problém n -těles (úloha o pohybu vzájemně interagujících n -hmotných bodů) je reálný, analyticky řešitelný není. Stejně tak jako obecný problém tří těles. Zde nastupuje moderní výpočetní technika a numerické metody matematiky, často s využitím nejvýkonnějších počítačů. Omezený problém tří těles (též nazývaný Lagrangeův problém tří těles) je speciální případ problému tří těles např. pro soustavu Slunce - Jupiter - kometa. Vychází z představy, že jedno z těles má zanedbatelnou hmotnosti oproti zbývajícím dvěma tělesům. Třetí lehké těleso může představovat již výše zmíněnou kometu, planetku nebo umělý satelit. O dvou těžších tělesech předpokládáme, že obíhají kolem společného těžiště po kruhových dráhách. Souřadné osy volíme tak, že se „velká“ tělesa pohybují v rovině xy .

Máme najít rychlost rotace soustavy těžších těles, které je o hmotnostech m_1 , m_2 , obíhající kolem společného těžiště. Vzdálenost obou těles je a . Z definice těžiště je vzdálenost tělesa m_1 od těžiště

$$a_1 = \frac{am_2}{(m_1 + m_2)}$$

Tělesa se přitahují gravitační silou

$$F_g = \kappa \frac{m_1 m_2}{a^2}$$

ta se musí rovnat dostředivé síle. Řešíme rovnici

$$m_1 \omega^2 a_1 = \kappa \frac{m_1 m_2}{a^2}$$

a po jednoduché algebraické úpravě dostáváme vyjádření 3. Keplerova zákona ve tvaru

$$\omega^2 = \kappa \frac{(m_1 + m_2)}{a^3}$$

kde

ω ... úhlová rychlost soustavy

κ ... gravitační konstanta.

2.1 Pohybová rovnice tělesa zanedbatelné hmotnosti

Studujme nyní pohyb třetího tělesa zanedbatelné hmotnosti. Chceme zjistit zrychlení tělesa \mathbf{a} , známe-li úhlovou rychlost soustavy $\boldsymbol{\omega}$ a rychlost tělesa vůči rotující soustavě \mathbf{v} . Pohybová rovnice pro toto těleso v soustavě spojené s rotující soustavou (její odvození nalezneme v každé učebnici mechaniky např. [3]) je

$$\mathbf{a} = -2\boldsymbol{\omega} \times \mathbf{v} + \omega^2 \mathbf{r} - \frac{\kappa m_1}{r_1^3} \mathbf{r}_1 - \frac{\kappa m_2}{r_2^3} \mathbf{r}_2 \quad (2.1.1)$$

kde

\mathbf{r} ... polohový vektor tělesa

r_1 ... vzdálenost m od m_1

r_2 ... vzdálenost m od m_2

m ... hmotnost tělesa, kde $m \ll m_1, m_2$

Rovnici (2.1.1) lze přepsat do tvaru

$$\mathbf{a} = -2\boldsymbol{\omega} \times \mathbf{v} - \nabla \Omega \quad (2.1.2)$$

kde

$$\Omega = -\frac{1}{2} \omega^2 (x^2 + y^2) - \frac{\kappa m_1}{r_1} - \frac{\kappa m_2}{r_2} \quad (2.1.3)$$

představuje tzv. efektivní potenciál (zde dodržujeme znaménkovou konvenci podle [4], v astronomii je obvyklé definovat efektivní potenciál Ω s opačným znaménkem, jako v [5] popřípadě [9]).

Nyní dokážeme, že (2.1.2) je ekvivalentní s (2.1.1). Přitom předpokládejme, že se zkoumané těleso pohybuje v rovině $z = 0$.

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{1}{r_1} \right) &= -\frac{1}{r_1^2} \frac{1}{\sqrt{(x-x_1)^2 + (y-y_1)^2}} \cdot (x-x_1) = -\frac{x-x_1}{r_1^3} \\ \frac{\partial}{\partial x} \left(\frac{1}{r_2} \right) &= -\frac{1}{r_2^2} \frac{1}{\sqrt{(x-x_2)^2 + (y-y_2)^2}} \cdot (x-x_1) = -\frac{x-x_2}{r_2^3} \\ \frac{\partial}{\partial x} \left(-\frac{1}{2} \omega^2 (x^2 + y^2) \right) &= -\omega^2 x \end{aligned}$$

Parciální derivace podle y analogicky

$$\frac{\partial}{\partial y} \left(\frac{1}{r_1} \right) = -\frac{y - y_1}{r_1^3}$$

$$\frac{\partial}{\partial y} \left(\frac{1}{r_2} \right) = -\frac{y - y_2}{r_2^3}$$

$$\frac{\partial}{\partial y} \left(-\frac{1}{2} \omega^2 (x^2 + y^2) \right) = -\omega^2 y$$

Ted' již můžeme zjistit jednotlivé složky $\mathbf{F} = -(2m\boldsymbol{\omega} \times \mathbf{v} + m\nabla\Omega)$

$$F_x \mathbf{e}_x = -\frac{\partial \Omega}{\partial x} \mathbf{e}_x = -2m\omega v_y \mathbf{e}_x + m\omega^2 x \mathbf{e}_x - \frac{\kappa m_1 m}{r_1^3} (x - x_1) \mathbf{e}_x - \frac{\kappa m_2 m}{r_2^3} (x - x_2) \mathbf{e}_x$$

$$F_y \mathbf{e}_y = -\frac{\partial \Omega}{\partial y} \mathbf{e}_y = 2m\omega v_x \mathbf{e}_y + \omega^2 y \mathbf{e}_y - \frac{\kappa m_1 m}{r_1^3} (y - y_1) \mathbf{e}_y - \frac{\kappa m_2 m}{r_2^3} (y - y_2) \mathbf{e}_y$$

$$\mathbf{F} = F_x \mathbf{e}_x + F_y \mathbf{e}_y = -2m\boldsymbol{\omega} \times \mathbf{v} + \omega^2 \mathbf{r} - \frac{\kappa m_1 m}{r_1^3} \mathbf{r}_1 - \frac{\kappa m_2 m}{r_2^3} \mathbf{r}_2$$

Užitím druhého Newtonova zákona dostáváme vztah (2.1.1).

Připomeňme, že rovina, ve které se velká tělesa pohybují, má rovnici $z = 0$. Osy x a y se otáčejí kolem osy z , přičemž osa x je zvolena tak, aby těleso m_1 mělo souřadnice $(-a_1, 0, 0)$ a m_2 $(a_2, 0, 0)$. Počátek volíme v hmotném středu soustavy těles m_1 a m_2 . Hmotnosti těles volíme $m_1 > m_2 \gg m$. Toto „nastavení“ základních veličin nás bude provázet celou kapitolou 2 a kapitolou 4. Kromě již výše zmíněného budeme volit jednotky tak, aby se vztahy mezi základními veličinami zjednodušili na tvar

$$m_1 + m_2 = 1$$

$$a_1 + a_2 = a = 1$$

$$\mu = \frac{m_2}{m_1 + m_2}$$

úpravou předešlých vztahů vyjádříme veličiny hmotnosti a vzdálenosti

$$m_1 = 1 - \mu$$

$$m_2 = \mu$$

$$a_1 = \mu$$

$$a_2 = 1 - \mu$$

dále pak položíme $\kappa = 1$, $\omega = 1$.

2.2 Hillova plocha

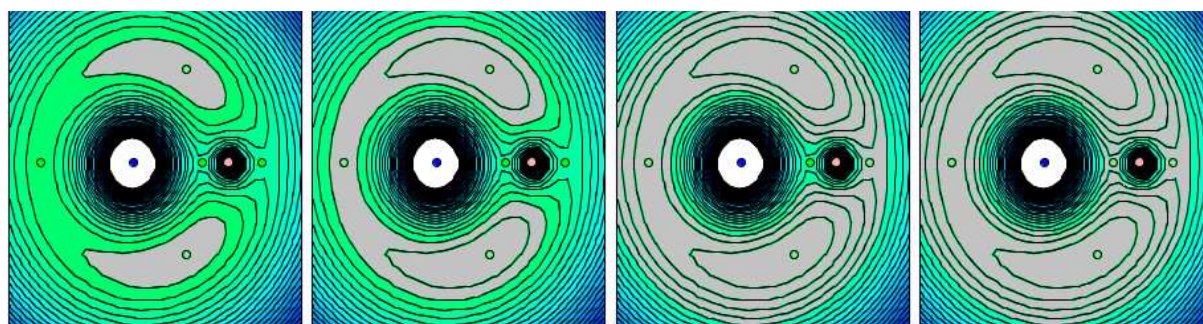
Z definice efektivního potenciálu (2.1.3) a pohybové rovnice (2.1.1) je možno vyjádřit kvadrát rychlosti tělesa zanedbatelné hmotnosti

$$v^2 = C - 2\Omega > 0 \quad (2.2.1)$$

Kde C je Jacobiho konstanta a můžeme jí považovat za základní charakteristiku pohybu daného tělesa. Čtverec relativní rychlosti na levé straně rovnice (2.2.1) je nezáporný! Proto také $C - 2\Omega$ je nezáporné. Plocha $C - 2\Omega = 0$ (2.2.2) představuje hranici části prostoru, v níž je pohyb třetího tělesa za daných podmínek myslitelný. Nazýváme ji **Hillovou plochou**.

Pohyb je povolen pouze v místech, kde $C > 2\Omega$. Pro určitou dráhu tělesa (např. komety) odpovídají „vrstevnice“ potenciálu Ω křivkám nulové rychlosti. Přiblíží-li se těleso s danými parametry k této křivce, tak se od ní „odrazí“.

V grafu funkce 2Ω se nachází pět bodů, které si označme L_1 až L_5 (jedná se o tzv. Lagrangeovy body, viz níže). Podle hodnot Jacobiho konstanty C můžeme rozlišit pět případů, jak vypadají dovolené a zakázané oblasti. Konstantu C snadno vypočítáme rozepsáním vztahu (2.2.1). Následující rozbor je upraven podle [5].



a,

b,

c,

d,

Obr. 1: Program EJS. Tvar dovolených (zelené) a zakázaných (šedé) oblastí pro pohyb zkoumaného tělesa.

Je-li $C > 2\Omega$ (v bodě L_4) = $2\Omega(L_5)$, pak je pohyb dovolen v celé rovině xy . Což lze předpokládat, protože udělíme-li tělesu velkou rychlost, bude také C velké. Připomeňme že $C = 2\Omega + v^2$.

Další případy hodnot Jacobiho konstanty:

a, $2\Omega(L_4) = 2\Omega(L_5) > C > 2\Omega(L_3)$ dvě zakázané oblasti jsou v okolí bodů L_4 a L_5 (obr. 1a).

b, $2\Omega(L_3) > C > 2\Omega(L_2)$ zakázané jsou okolí bodů L_3, L_4, L_5 . Tyto okolí vytvářejí zakázanou oblast ve tvaru podkovy (obr. 1b).

c, $2\Omega(L_2) > C > 2\Omega(L_1)$ (obr. 1c).

d, $C < 2\Omega(L_1)$ se zkoumané těleso může pohybovat v okolí „velkých těles“ a to buď v okolí jednoho, nebo druhého, popřípadě ve velké vzdálenosti od těchto těles. V praxi to znamená, že dostane-li se např. kometa či planetka o hmotnosti m do gravitačního působení jednoho z těles, tak nemůže přejít do gravitačního působení druhého tělesa (obr. 1d).

Dva zvláštní tvary drah dostaly své názvy. Jde o případ **a** – jedná se o orbit typu *pulec* (v soustavě Slunce-Jupiter jsou to shluky planetek Řeků a Trojanů) a případ **b** - *podkova* (koorbitální satelity Saturnu, Janus a Epimetheus).

Položme si otázku, jak Hillova plocha vlastně vypadá? Připomeňme, že Hillova plocha je hranicí nulové rychlosti. Tvar Hillovy plochy dostaneme pomocí řezů souřadnými rovinami. Rovnice Hillovy plochy bude (po zavedení vztahu mezi základními veličinami na konci předešlé kapitoly) podle (2.2.2)

$$x^2 + y^2 + 2\frac{m_1}{r_1} + 2\frac{m_2}{r_2} = C$$

Proveďme řez rovinou $z = 0$. Dostáváme tři geometrické útvary, přibližně kružnice

$$\mathbf{a}, x^2 + y^2 = C - b_1$$

r_1, r_2 jsou velké

$$\mathbf{b}, (x + a_1)^2 + y^2 = \left(\frac{2m_1}{C - b_2}\right)^2$$

r_1 je velmi malé

$$\mathbf{c}, (x - a_2)^2 + y^2 = \left(\frac{2m_2}{C - b_3}\right)^2$$

r_2 je velmi malé

kde b_1 , b_2 a b_3 jsou malé veličiny.

Zjištěné výsledky odpovídají případu, kdy dvě menší kružnice (\mathbf{b} , \mathbf{c}) se nacházejí uvnitř velké kružnice (\mathbf{a}). Středů kružnic \mathbf{b} a \mathbf{c} splývají se souřadnicemi těles m_1 a m_2 . Zbylé řezy rovinami $x=0$ a $y=0$ může čtenář nalézt v [2].

2.3 Librační body

Body, ve kterých se zkoumané těleso m nepohybuje vůči soustavě velkých těles, nazýváme librační body. Jsou to tedy body, v nichž na těleso m nepůsobí žádná síla. Při zjišťování bodů vyjdeme z pohybové rovnice (2.1.1) a dosadíme za $\mathbf{v} = 0$, $\mathbf{a} = 0$. Dostaneme tak dvě rovnice

$$a_x = \omega^2 x - \frac{\kappa m_1}{r_1^3}(x - a_1) - \frac{\kappa m_2}{r_2^3}(x + a_2) = 0 \quad (2.3.1)$$

$$a_y = \omega^2 y - \frac{\kappa m_1}{r_1^3}y - \frac{\kappa m_2}{r_2^3}y = 0 \quad (2.3.2)$$

Z rovnice (2.3.2) ihned dostáváme řešení a to $y = 0$ nebo

$$\omega^2 = \frac{\kappa m_1}{r_1^3} + \frac{\kappa m_2}{r_2^3} \quad (2.3.3)$$

Bude-li se zabývat případem kdy $y \neq 0$, můžou být rovnosti (2.3.1) a (2.3.3) splněny pouze za předpokladu, že $r_1 = r_2 = a$. Tak dostaneme dva z pěti libračních bodů, které se označují jako L_4, L_5 . Tyto body nalezneme jako vrcholy rovnostranných trojúhelníků se základnou a .

Ukazuje se, že Librační body L_4 a L_5 jsou stabilní, je-li poměr hmotností

$$\mu < 0,0385$$

(převzato z [4]).

Znamená to, že tělesa v těchto bodech mohou setrvat velmi dlouhou dobu. Důkazem jsou, již výše zmíněná poměrně početná populace Trojanů a Řeků v Lagrangeových bodech příslušející k Jupiteru. (k datu 26. 8. 2008 bylo známo 1274 asteroidů v L_4 a 1272 v L_5 podle [4]).

Zbylé librační body musí splňovat podmínku $y = 0$, to znamená, že body leží na ose x . V rovnici (2.3.1) dosadíme podmínku $y = 0$ a dostaneme rovnici

$$\omega^2 x - \frac{\kappa m_1}{|x - a_1|^3} (x - a_1) - \frac{\kappa m_2}{|x + a_2|^3} (x + a_2) = 0 \quad (2.3.4)$$

Přibližné řešení rovnice (2.3.4) lze nalézt v publikaci [9]

$$\begin{aligned} L_{x1} &\approx 1 - \mu - \beta + \frac{\beta^2}{3} + \frac{\beta^3}{9} + 23 \frac{\beta^4}{81} \\ L_{x2} &\approx 1 - \mu + \beta + \frac{\beta^2}{3} - \frac{\beta^3}{9} - 31 \frac{\beta^4}{81} \\ L_{x3} &\approx -1 - \mu + \frac{7}{12} \frac{m_2}{m_1} - \frac{7}{12} \left(\frac{m_2}{m_1}\right)^2 + \frac{13223}{20736} \left(\frac{m_2}{m_1}\right)^3 \end{aligned}$$

kde

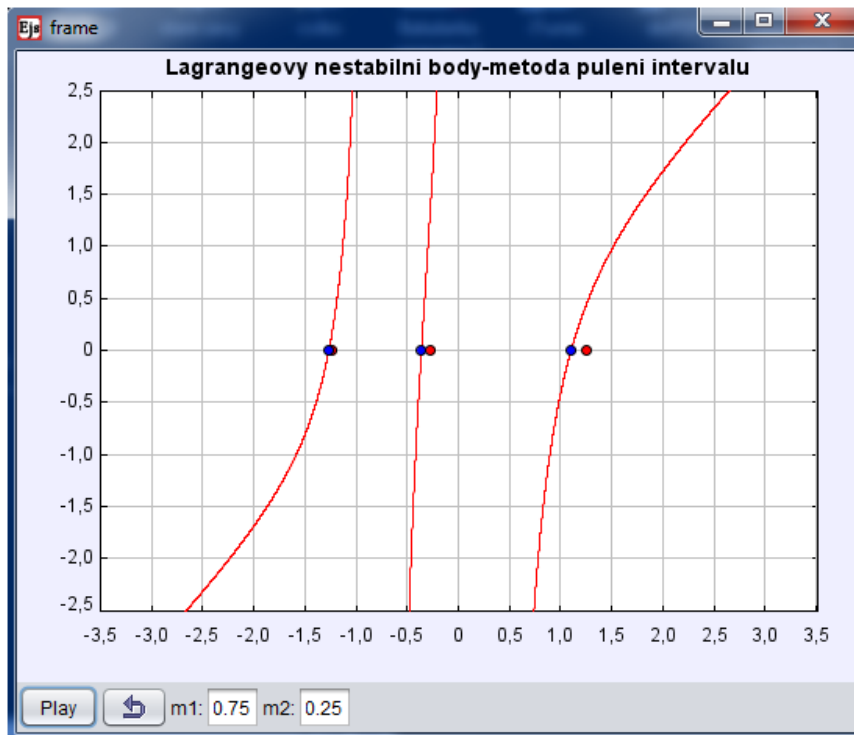
$$\beta = \sqrt[3]{\left(\frac{\mu}{3(1-\mu)}\right)}$$

$L_{x1} \dots$ je x -složka Lagrangeova bodu L_1

$L_{x2} \dots$ je x -složka Lagrangeova bodu L_2

$L_{x3} \dots$ je x -složka Lagrangeova bodu L_3

Takto dostaneme zbylé librační body L_1, L_2, L_3 . Mějme na paměti, že tento výsledek je pouze přibližný. Pro přesnější výpočet musíme požit některou numerickou metodu, např. *půlení intervalu* (EJS nám umožňuje pracovat s přesností 10^{-17}). Na obr. 2 dostáváme grafické řešení rovnice (2.3.4) v programu EJS. Kde červené body jsou přibližné výpočty libračních bodů podle výše uvedených vztahů a modré body jsou získány pomocí metody *půlení intervalu*.



Obr. 2: Simulace v EJS-výpočet nestabilních Lagrangeových bodů pro $\mu = 0.25$

Body L_1, L_2, L_3 jsou nestabilní a odpovídají sedlovým bodům efektivního potenciálu. Těleso se v nich dlouhodobě neudrží, přesto těchto bodů využíváme. Pro systém Slunce-Země mohou tělesa zůstat v okolí Lagrangeova bodu řádově měsíc. Příkladem mohou být kosmické sondy SOHO a WMAP, které se nachází poblíž L_1 a L_2 .

Vraťme se k libračním bodům L_4, L_5 v soustavě Slunce-Jupiter. Názvy skupin asteroidů nacházející se v těchto bodech jsou inspirovány antickým bojem mezi Řeky a Trojany. Pojmenování samotných asteroidů taktéž pochází z řecké mytologie, několik těchto asteroidů uvádím v tab. 1, spolu se základními parametry. Dvěma parametrům budeme věnovat větší pozornost, jsou jimi *absolutní jasnost* a *sklon dráhy*. *Zdánlivá jasnost*, též *zdánlivá magnituda* nebo *zdánlivá hvězdná velikost*, představuje fyzikální veličinu, používanou hojně

v astronomii, která udává jasnost nebeského tělesa. Jednotkou *zdánlivé jasnosti* je jedna magnituda (značeno m). *Absolutní jasnost* je potom *zdánlivá jasnost* tělesa ve vzdálenosti 10 parseků¹. Jako zajímavost uvádím vztah pro výpočet *zdánlivé jasnosti*

$$h = -2.5 \cdot \log_{10} I/I_0$$

kde

I_0 ... hustota světelného toku hvězdy, které je přiřazena *zdánlivá jasnost* 0^m

I ... je hustota světelného toku dopadající na oči pozorovatele.

Označení	Jméno	Oblak	Absolutní jasnost[m]	Sklon dráhy[°]	Průměr[km]
624	Hektor	L4	7,49	18,2	191
911	Agamemnon	L4	7,89	21,8	158
1143	Odysseus	L4	7,93	3,1	155
3451	Mentor	L5	8,10	24,7	143
617	Patroclus	L5	8,19	22,0	137
3317	Paris	L5	8,30	27,9	130
1437	Diomedes	L4	8,30	20,5	130
1867	Deiphobus	L5	8,30	26,9	130
1172	Aneas	L5	8,33	16,7	128
2797	Teucer	L4	8,40	22,4	125
1583	Antilochus	L4	8,60	28,5	113
3063	Makhaon	L4	8,60	12,2	113
4063	Euforbo	L4	8,60	18,9	113
2241	Alcathous	L5	8,64	16,6	111
588	Achilles	L4	8,67	10,6	110

Tab. 1: Vybraní Trojaní a Řeci o průměru větším než 100 km, zdroj [8].

Je potřeba poznamenat, že čím je *absolutní magnituda* menší, tím je jasnost tělesa větší a naopak.

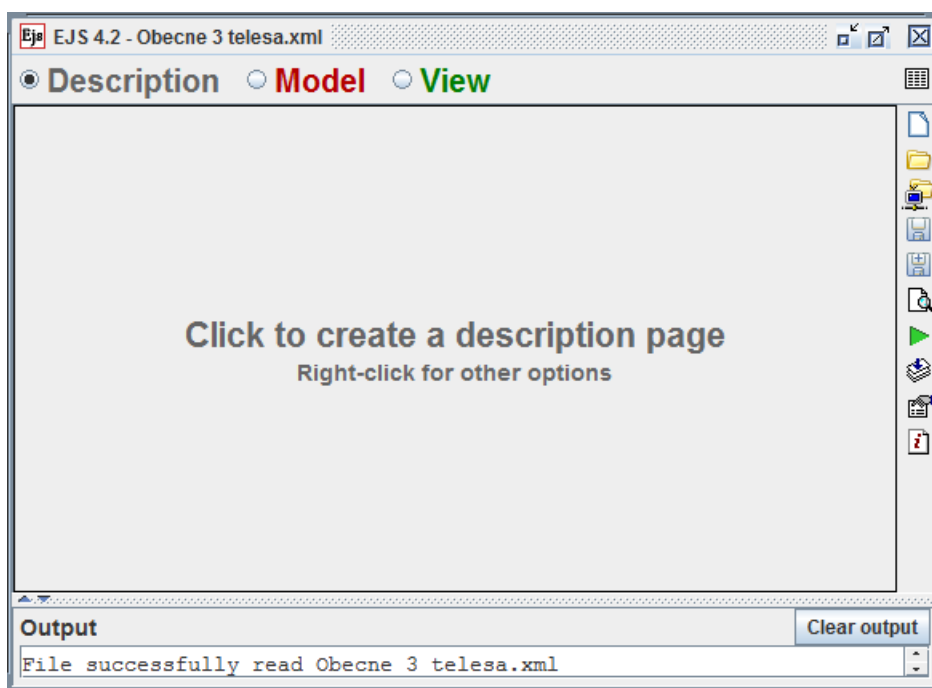
Sklon dráhy vyjadřuje úhel mezi rovinou dráhy tělesa a rovinou xy příslušné souřadné soustavy. Obvykle se udává ve stupních, zřídka v radiánech. Všimněme si velkého sklonu dráhy Trojanů a Řeků (Tab. 1).

¹ 1 parsek $\approx 3,086 \cdot 10^{16}$ m

3. Program Easy Java Simulation


Náš obecný problém tří těles budeme simulovat v programu Easy Java Simulation (dále jen *EJS*). Program *EJS* byl vytvořen tak, aby jsme pomocí jednoduchých nástrojů mohli vytvářet interaktivní simulace v jazyku Java. Pomocí *EJS* můžeme tedy vytvořit simulaci téměř bez předešlé zkušenosti s programováním.


3.1 První kontakt s Easy Java Simulation





Obr. 3: Hlavní panel programu Easy Java Simulation

Obrázek tři nám ukazuje rozhraní *EJS*. Na první pohled může mnohé zaskočit množství ikon na hlavním panelu, ale jak sám název programu napovídá, nejedná se o nic složitého. Popíšeme nejdůležitější ikony na pravé straně hlavního panelu, které budeme při své práci často používat.


 „New“. Tato ikona nám ukončí aktuální simulaci (zda-li je nějaká spuštěná) a otevře novou.


 „Open“. Tato ikona nám nahraje existující simulaci.

 „Save“. Tato ikona uloží současnou simulaci pod stejným názvem, jakým byla simulace nahrána.

 „Save as“. Tato ikona uloží současnou simulaci pod námi zadaným názvem.

 „Run“. Tato ikona spouští simulaci.

 „Options“. Tato ikona nám dovoluje změnit nastavení, vzhled a chování *EJS*.

 „Information“. Tato ikona nás posílá na webové stránky programu *EJS*, poskytující základní informace o tomto programu.

Po představení základních ikon nás teď čeká objasnění významu záložek „Description“, „Model“ a „View“. Záložka „Description“ je po spuštění programu otevřená a její jediná funkce je popis simulace. Popisem simulace rozumíme vložení základních informací. O simulaci zde můžeme na příklad uvést, o jaký fyzikální jev se jedná a jaké fyzikální vztahy a zákony použijeme. Jednoduše v záložce „Description“ představíme celý náš projekt. Záložka „Model“ bude pro naši práci daleko podstatnější. V této záložce budeme do programu zadávat rovnice a proměnné, kterými se bude naše simulace řídit. A konečně v záložce „View“ budeme naši simulaci upravovat výstupy a dávat jí konečnou grafiku. Podrobněji se budeme těmto záložkám věnovat v další kapitole, kde si mnohé objasníme na konkrétním jednoduchém příkladu.

3.2 Příklad tělesa na pružině v programu Easy Java Simulation

Jako ukázkou, která by nám měla pomoci se lépe seznámit s prostředím *EJS*, jsem zvolil jednoduchou simulaci tělesa na pružině. Pro pohyb tělesa na pružině použijeme Hookova zákona a druhého Newtonova zákona. Dostaneme vztah

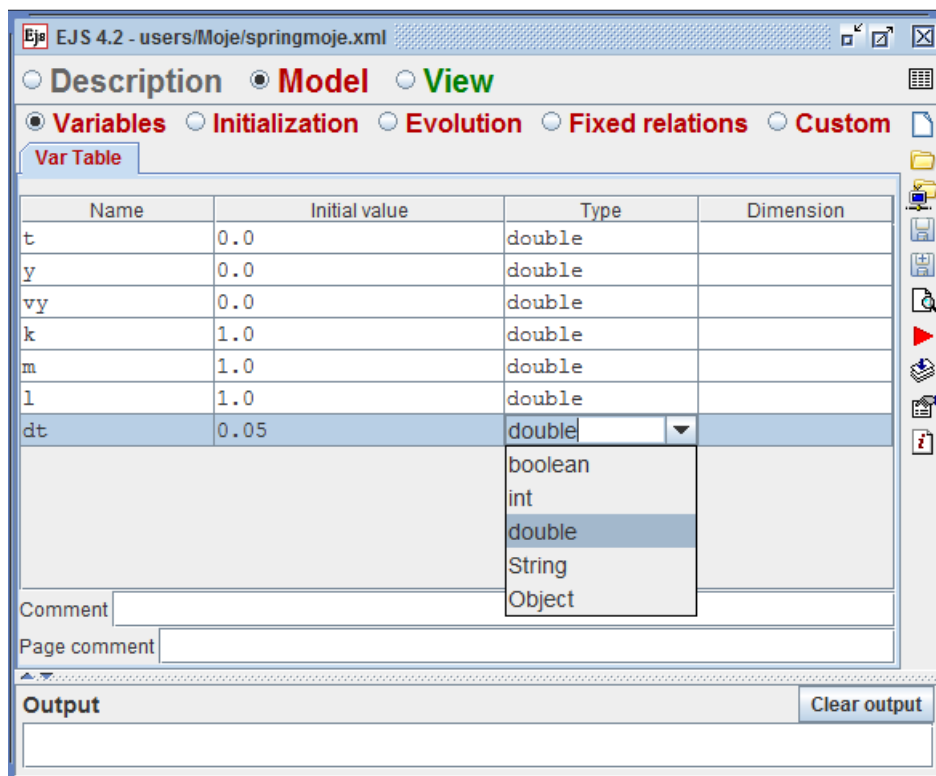
$$y'' = -\frac{k}{m}(y - l)$$

Kde k je tuhost pružiny, l délka pružiny v rovnovážné poloze a y poloha závaží.

První důležitou věcí bude zadání proměnných a konstant do *EJS*. Zadáme záložku *Model* a podzáložku *Variables*, tak jak je vidět na obr. 4. V podzáložce *Variables* definujeme konstanty a proměnné a zároveň zvolíme typ proměnných. Typy proměnných jsou v *EJS* následující

- *boolean* – proměnné typu boolean mají pouze dvě hodnoty a to true nebo false
- *byte*, *short*, *int* a *long* – celočíselné proměnné
- *float* a *double* – proměnné jsou reálná čísla
- *char* a *string* – pro znaky a texty

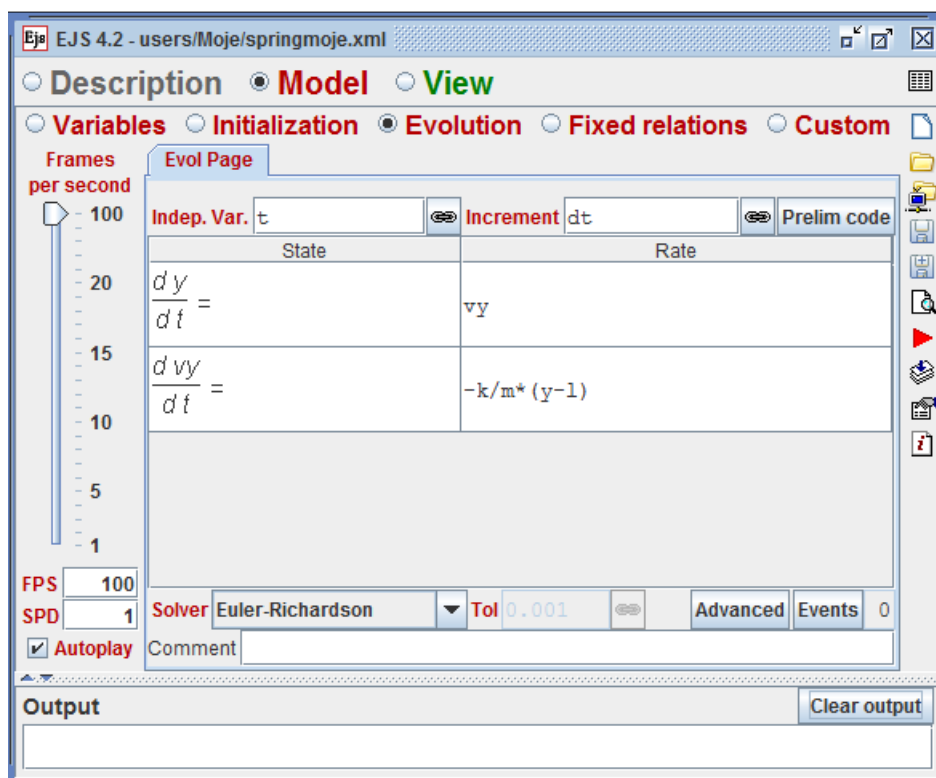
V našem příkladu budeme pracovat s pouze proměnnými typu *double*.



Obr. 4: Zadané proměnné pro případ tělesa na pružině i se svými počátečními hodnotami (initial value).

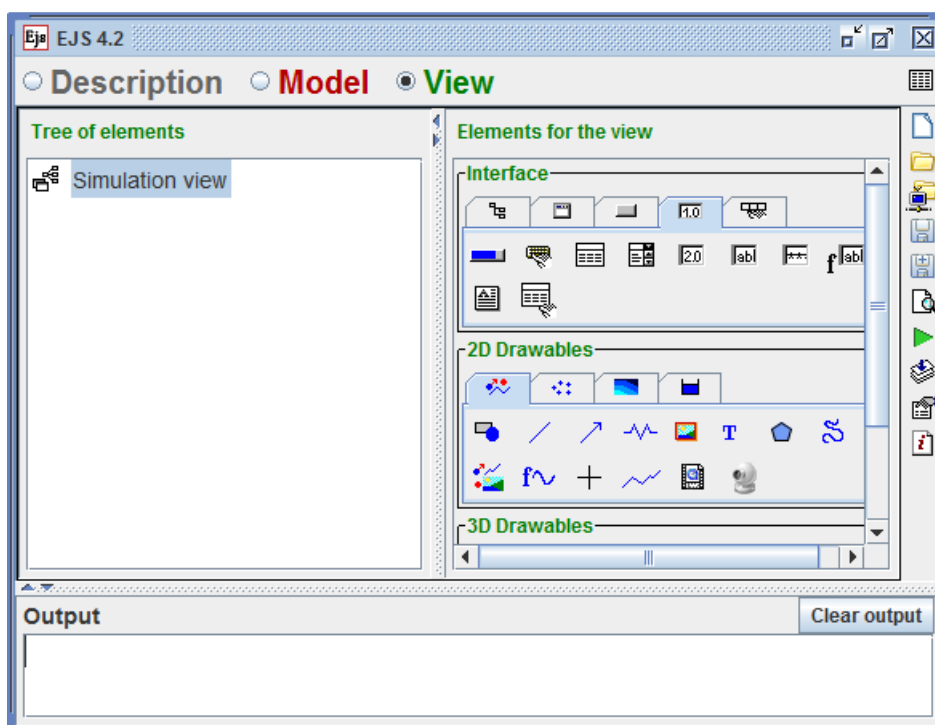
Každá naše proměnná je zadána i se svou počáteční hodnotou. Budeme-li potřebovat komplexnější inicializaci, můžeme použít druhé podzáložky *Initialization*. V našem modelovém případě to nebude nutné, proto zůstane tato podzáložka prázdná.

Nyní se zaměříme na další podzáložku *Evolution*, kde budeme zadávat rovnice, kterými se bude simulace řídit. Použijeme výše zmíněnou obyčejnou diferenciální rovnici druhého řádu, která vznikne spojením druhého Newtonova a Hookova zákona. Program *EJS* nám nedovolí zadat diferenciální rovnice druhého a vyššího řádu. Zdálo by se, že nám to práci na simulaci tělesa na pružině může značně zkomplikovat, ale není tomu tak. Tento problém vyřešíme a to tak, že místo jedné diferenciální rovnice druhého řádu, zadáme do programu dvě diferenciální rovnice prvního řádu. V našem konkrétním příkladě využijeme vztahu (substituce) $\dot{y} = v_y$, což bude naše první diferenciální rovnice prvního řádu. Mějme na paměti, že v podzáložce *Variables* jsme v_y zadali jako proměnnou. Druhá diferenciální rovnice vznikne nahrazením $\ddot{y} = \dot{v}_y$. Situace je znázorněna na obr. 5.



Obr. 5: Okno podzáložky *Evolution* s dvěma diferenciálními rovnicemi prvního řádu.

V okně podzáložky *Evolution* ještě chvíli zůstaneme. Krom rovnic jsme zadali i nezávislou proměnnou t a v kolonce *Increment* jsme zvolili přírůstek dt , kterému jsme v podzáložce *Variables* přiřadili hodnotu 0.05. Dále jsme si zvolili numerickou metodu výpočtu a to



Obr. 6: Okno záložky View.

Euler-Richardsonovu. V *EJS* jsou algebraické operace jako sčítání, odčítání, násobení a dělení reprezentovány znaky „+ , - , * , /“.

Jiné rovnice obsahující složitější výrazy se zapisují do *Evolution* pomocí přípony *Math*. Na příklad obsahuje-li naše rovnice výraz x^2 , do systému musíme zadat *Math.pow(x, 2)*, kde *pow(x, y)* překládá program jako funkci x^y . Podrobný seznam výrazů a jejich zápisu do *EJS* lze nalézt v [7].


Poslední dvě záložky *Fixed relations* a *Custom* zůstanou v našem příkladě prázdné. *Fixed relations* je psán v Java kódu a používáme jej v případě, kdy naše proměnné mají mezi sebou pevný vztah.

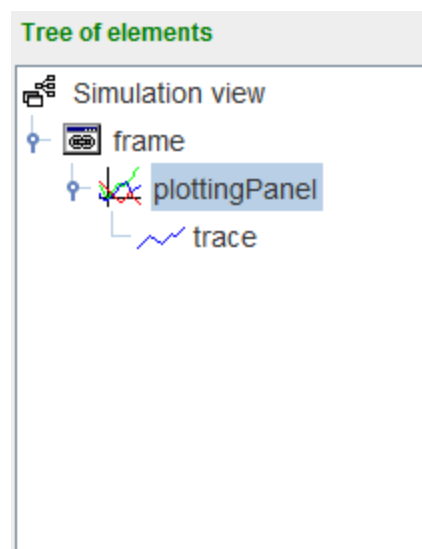
Proměnné a rovnice máme zadány, zbývá nám tedy simulaci vytvořit grafickou podobu. Otevřeme záložku *View*. Jak lze vidět z obr. 6, záložku máme rozdělenou na dvě poloviny. V pravé polovině máme seznam prvků pro vytváření simulace (*Elements for the view*) a v levé polovině tyto prvky dáváme dohromady (*Tree of the elements*).

Prvky (*elements*) máme rozděleny do tří skupin

- *interface* – zde nalezneme základní prvky, jako jsou rámy (*frame*), panely (*plotting panel*, *drawing panel*), tlačítka (*buttons*), číselná pole (*field*)
- *2D Drawables* – prvky o dvou souřadnicích
- *3D Drawables* – prvky o třech souřadnicích

Náš modelový případ, tělesa na pružině, nasimulujeme ve dvou-dimenzionálním prostoru. Budeme tedy využívat pouze prvky ze skupin *interface* a *2D Drawables*.

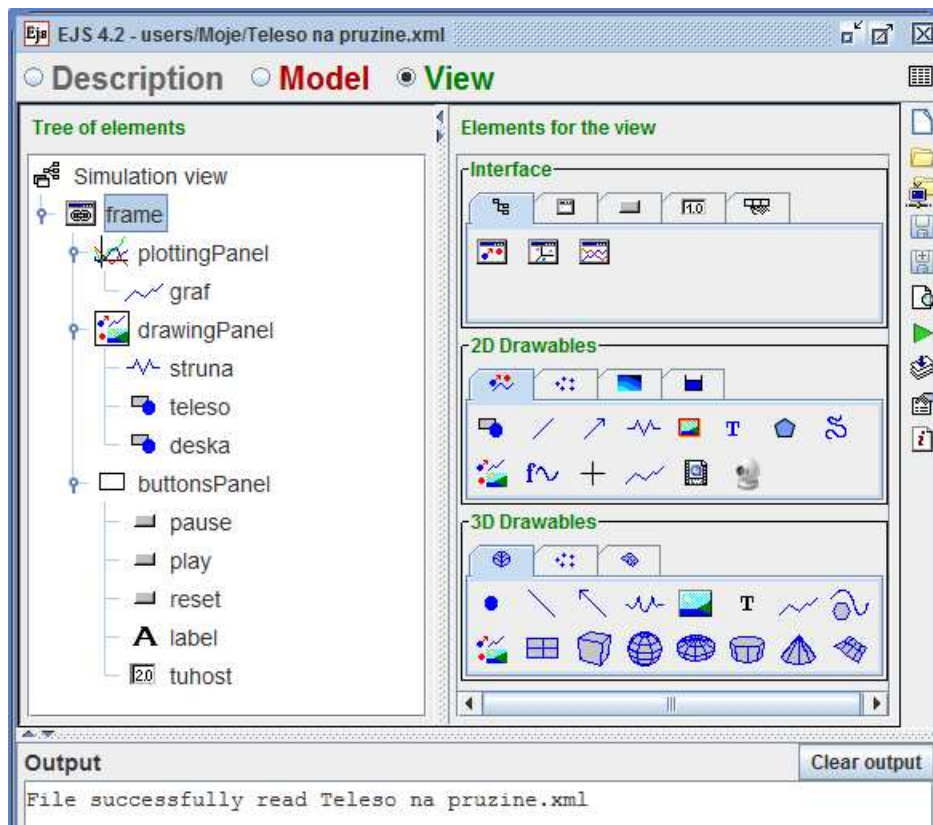
Chceme-li vytvořit simulaci dle našich představ, musíme si zvolit vhodné prvky a tyto prvky pak přesunout do levé poloviny okna *Wiev*. Samotný přesun je realizován „kouzelnou hůlkou“ , která jednotlivé prvky spojí a vytvoří tak „strom elementů“. Ten musí mít logickou navaznost, není možné vytvořit graf bez pozadí, na kterém by se zobrazoval. Proto nejprve musíme vytvořit rám (*frame*), poté přidat panel (*plotting panel*) a na konec graf (*trace*).



Obr. 7: „strom elementů“

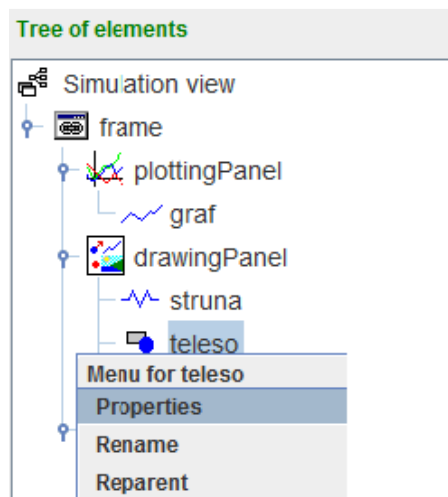
Takto poskládaný strom elementů nevytvoří žádnou simulaci. Je nutné propojit prvky s proměnnými nebo rovnicemi. K tomu nám slouží vlasnosti (*properties*) jednotlivých prvků. *Properties* patří mezi nejdůležitější nástroje záložky *View*. V *properties* lze také měnit vzhled prvků a přizpůsobit si prvek tak jak jej potřebujeme. Označením prvku a kliknutím pravým tlačítkem myši se dostaneme do *properties* prvku.

Vraťme se k našemu příkladu tělesa na pružině. Na obr.8 můžeme vidět již vytvořený strom elementů.

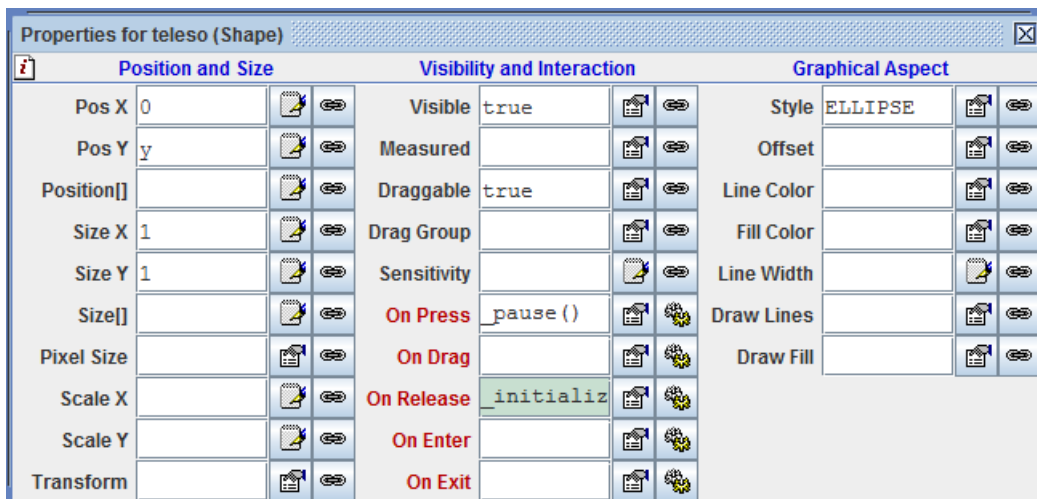


Obr. 8: Okno *View* tělesa na pružině.

Označme prvek „teleso“ tak jak je znázorněno na obr. 9 a dostaneme se do *properties* tohoto prvku. obr. 10 ukazuje prostředí *properties*.




Obr. 9: Strom elementů tělesa na pružině.



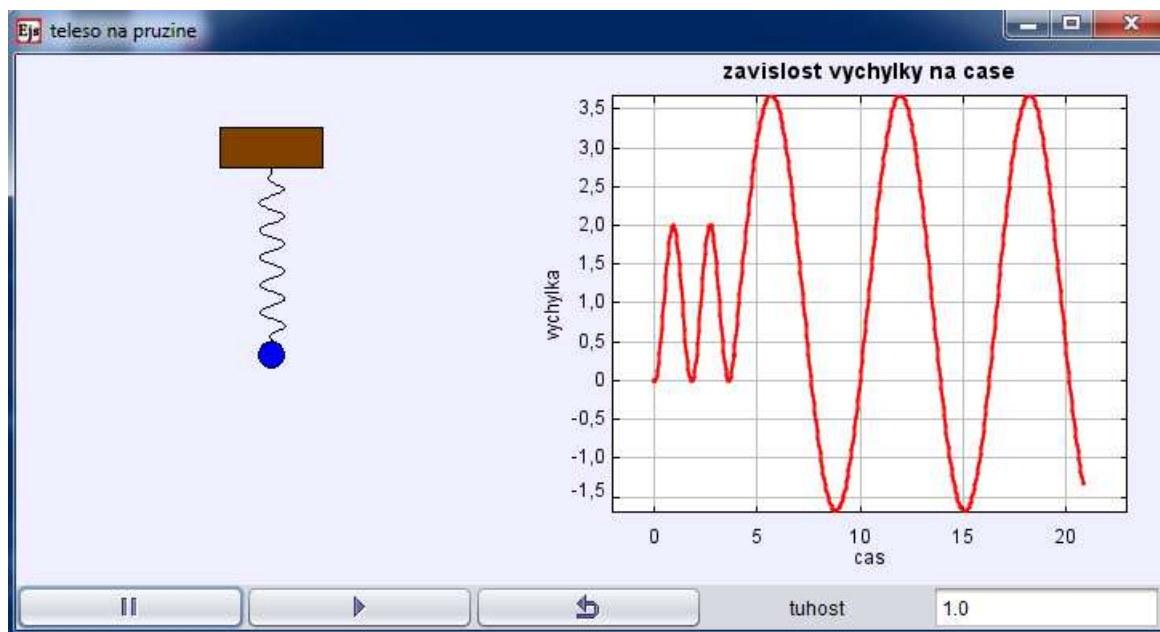
Obr. 10: *Properties* prvku „teleso“

Ve vlastnostech „telesa“ na obr. 10 máme zadány a nadefinovaný tvar tělesa, velikost tělesa, interakci pozorovatele s tělesem a nakonec v kolonce *Pos Y* je prvek spojen s rovnicemi přes proměnnou *y*. Červeně zvýrazněné kolonky jsou tak zvané *action properties*, zde si můžeme nastavit jak se simulace bude chovat při interakci s pozorovatelem. V našem případě můžeme těleso přetahovat (*Draggable*) a v *action properties* jsme nadefinovali co se bude dít po uvolnění (*On Release*)

```
_initialize();
_view.resetTraces();
```

Tento kód říká, že program přijme nové počáteční hodnoty proměnných (při natažení změníme počáteční výchylku) a smaže stopu (*trace*) v našem případě graf, který vznikne při natažení. Tyto základní kódy si nemusíme pamatovat, jejich seznam (včetně významu jednotlivých kódů) dostaneme kliknutím na ikonu .

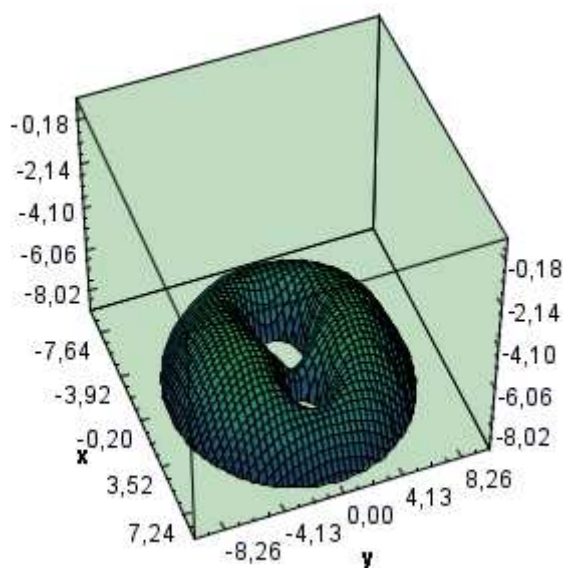
Takto si upravíme vlastnosti všech prvků vyskytující se v našem stromě elementů. Výsledná simulace je znázorněná na obr. 11. Je zde také graf závislosti výchylky na čase. V průběhu simulace jsme změnili *tuhost* z hodnoty 2 na 1.



Obr.11: Simulace tělesa na pružině. Graf ukazuje změnu výchylky v čase při rozdílné *tuhosti*.

V následující kapitole se opět vrátíme k problému tří těles.

4. Omezený problém tří těles v *EJS*



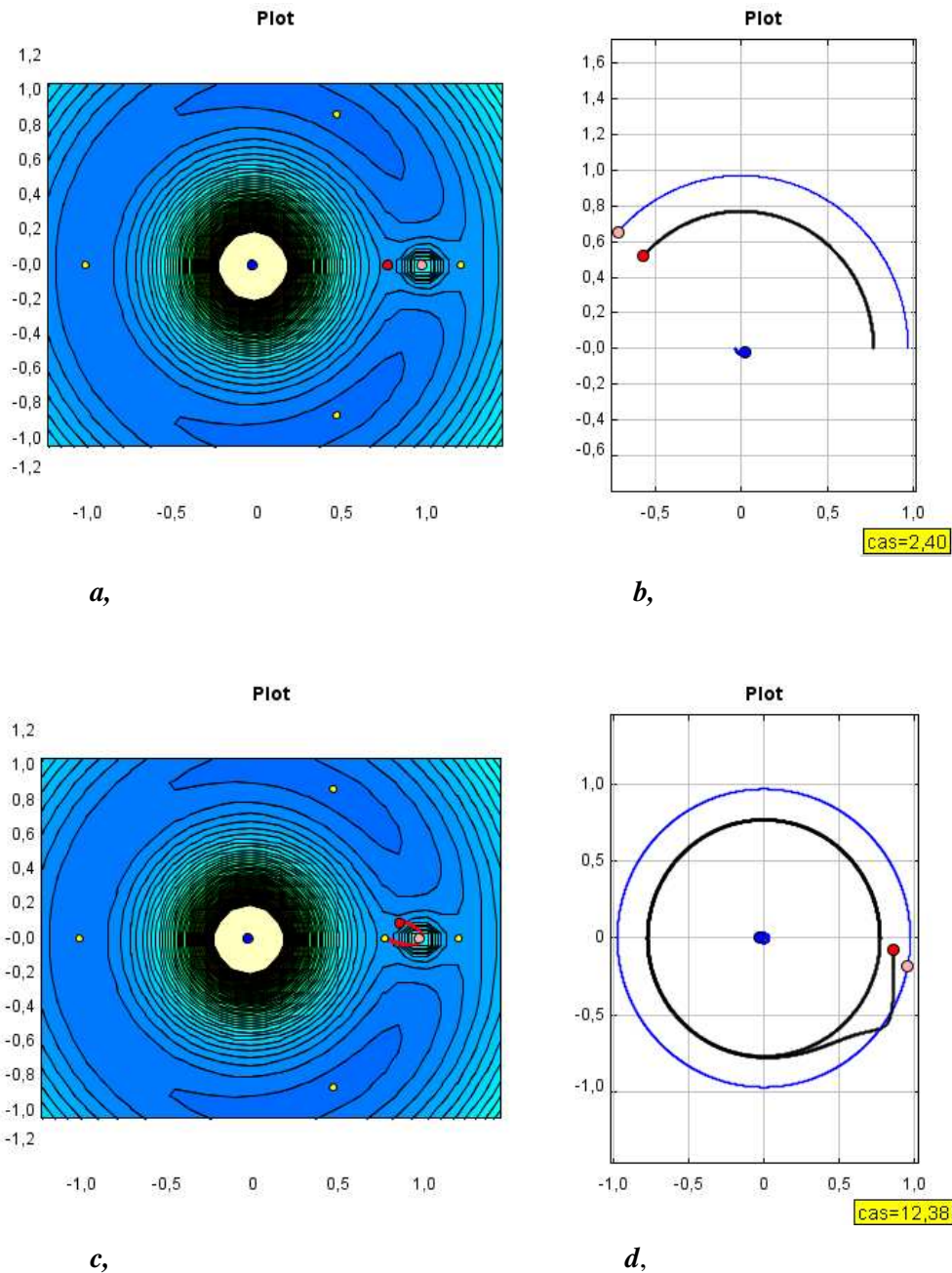
Obr. 12: Efektivní potenciál omezeného problému tří těles v *EJS*.

Úkolem této kapitoly je snaha seznámit čtenáře se simulací speciálního problému tří těles a s aplikací této simulace jak na konkrétní příklady, tak na definované pojmy a situace v první kapitole. V závěru této kapitoly je čtenář seznámen se stručným popisem vzniku simulace v programu *EJS*.

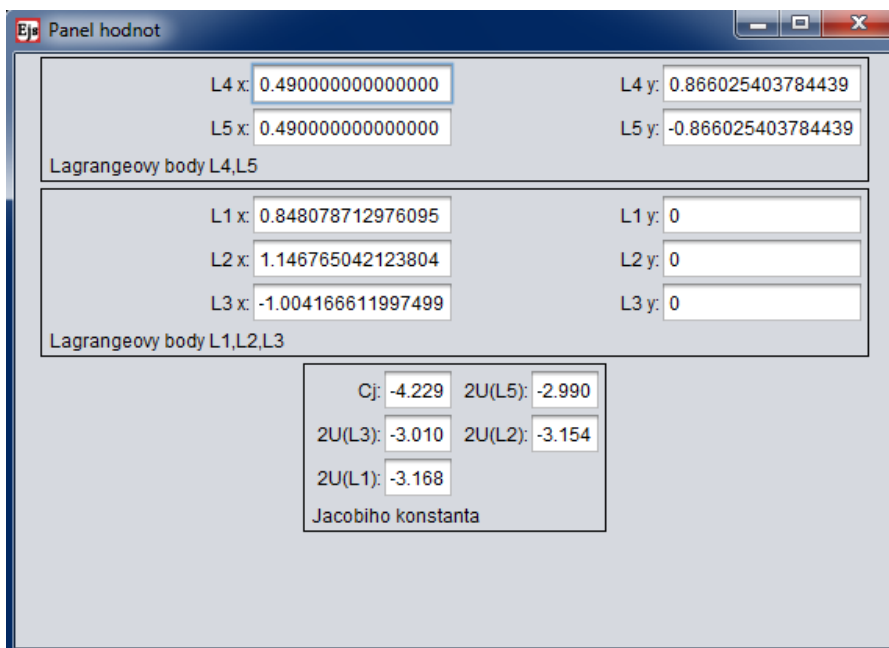
4.1 Simulace: Lagrangeovy stabilní a nestabilní body

Na obr. 12 je znázorněn graf efektivního potenciálu Ω . Pro naše účely položíme $z = 0$, dostaneme tak efektivního potenciál v rovině xy , tak jak lze vidět na obr. 13a a 13c. Umístíme těleso zanedbatelné hmotnosti do okolí nestabilního Lagrangeova bodu např. L_1 a můžeme pozorovat chování tělesa, které se jistý čas nebude pohybovat vzhledem k oběma větším tělesům, obr. 13a. Analogická situace je znázorněna na obr. 13b zde ovšem je problém pozorován z pohledu pozorovatele nacházejícího se v inerciální vztažné soustavě. Trajektorie pohybu tělesa zanedbatelné hmotnosti je na simulaci znázorněna červenou stopou, samotné těleso pak jako červený bod. Na obr. 13c je simulace v pokročilejším čase, zkoumané těleso

se nedokáže v sedlovém nestabilním bodě udržet a začíná se pohybovat vůči těžším tělesům. Počáteční podmínky pro tuto simulaci dostaneme numerickým výpočtem vztahu (2.3.4.) V programu *EJS* jsme naprogramovali vygenerování počátečních podmínek pro Lagrangeovy body, obr. 14. V podkapitolách 4.1 a 4.2 volíme $\mu = 0,030$.



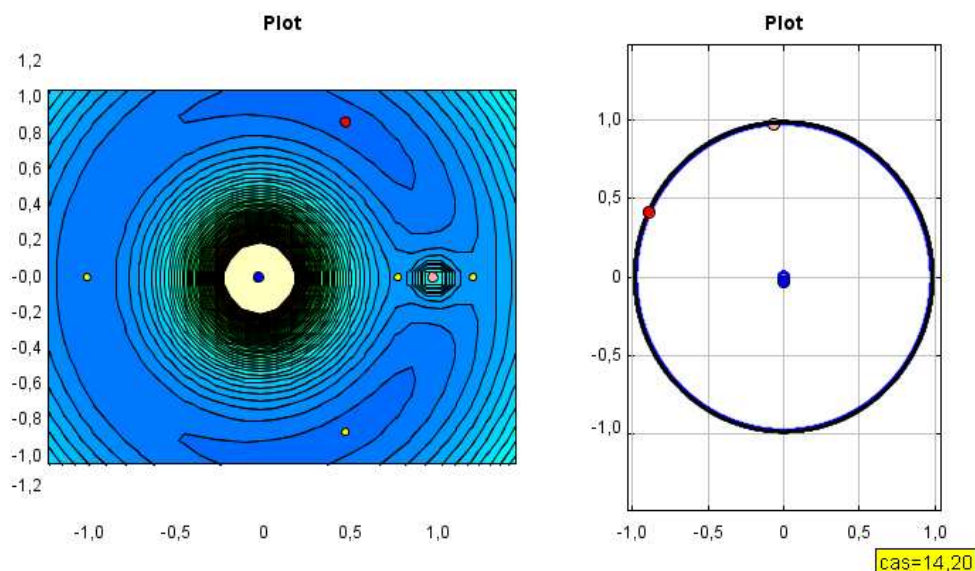
Obr. 13: Simulace tělesa v Lagrangově bodě L_1 . Na obrázcích *a*, *b* se těleso nachází v čase 2,40, na zbylých dvou, *c*, *d* v čase 12,38.



Obr. 14: Souřadnice Lagrangeových bodů a Jacobiho konstanta.

Posledním úkolem této podkapitoly je simulace pohybu tělesa ve stabilním Lagrangeovém bodě. Jak z teorie víme, tyto body jsou dva, L_4 a L_5 . Těleso v těchto bodech se nepohybuje vzhledem k dvěma větším tělesům. Počáteční podmínky opět vyčteme z obr. 14.

Volíme bod L_4 .

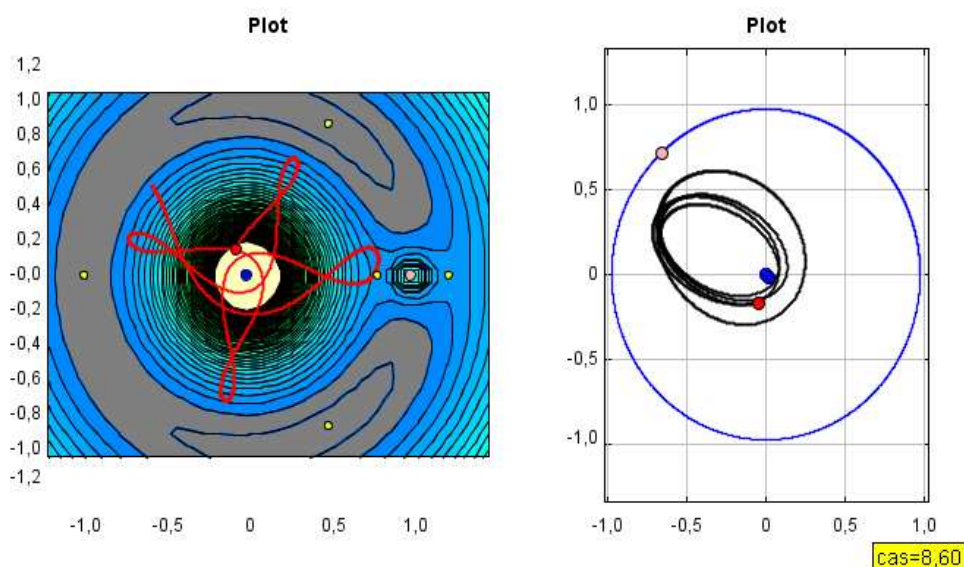


Obr. 15: Těleso ve stabilním Lagrangeovém bodě.

Na obr. 15 lze vidět, že se těleso vůči svým větším tělesům nepohybuje, což je přesně to, co očekáváme.

4.2 Simulace: Jacobiho konstanta

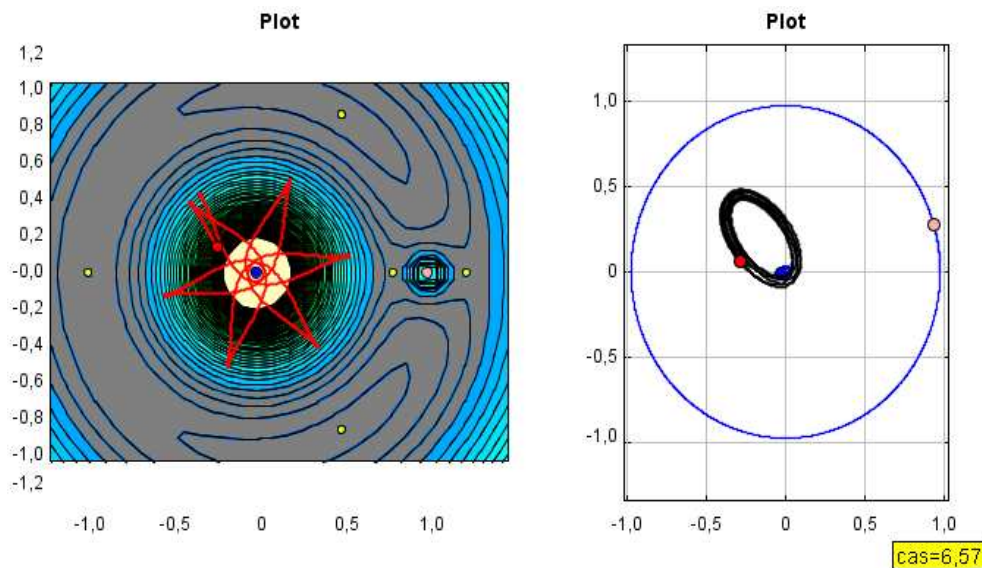
Cílem této kapitoly je objasnit čtenáři význam Jacobiho konstanty pomocí simulace v *EJS*. Z teorie víme, že konkrétní volbou počátečních podmínek dostáváme tzv. Jacobiho konstantu, a z hodnoty Jacobiho konstanty jsme schopni určit oblasti, ve kterých se těleso může pohybovat a ve kterých nikoli. V simulaci lze určit Jacobiho konstantu z počátečních podmínek, tak jak lze vidět ve spodní části na obr. 14. Vyčteme zde také hodnoty dvojnásobku efektivního potenciálu v Lagrangeových bodech. Tyto hodnoty jsou pro naše účely velice důležité a slouží jako „orientační body“. V naší simulaci se budeme zabývat dvěma případy. Prvním z nich bude situace, která se v literatuře označuje jako „podkova“ (viz podkapitola 2.2). Ve druhém případě zvolíme Jacobiho konstantu tak, že se zkoumané těleso bude pohybovat pouze v okolí tělesa m_1 .



Obr. 16: Situace „podkova“, těleso se vyhýba zakázaným šedým oblastem.

Obr. 16 nám ukazuje trajektorii (červená stopa) tělesa s Jacobiho konstantou $C = -3,210$. Pro kterou platí $2\Omega(L_3) = -3,030 > C > 2\Omega(L_2) = -3,278$. Šedá oblast je pro těleso zakázaná oblast a vytváří tzv. podkovu. Počáteční podmínky pro tuto situaci jsou:

$$x = -0,59587 \quad y = 0,50042$$



Obr. 17: Těleso se může pohybovat pouze v okolí velkých těles.

V druhém případě volíme Jacobiho konstantu $C = -3,930$, obr. 17. Počáteční podmínky:

$$x = -0,43767 \quad y = 0,35995$$

Rychlost tělesa byla v obou případech nulová.

4.3 Simulace: Arenstorfova orbita

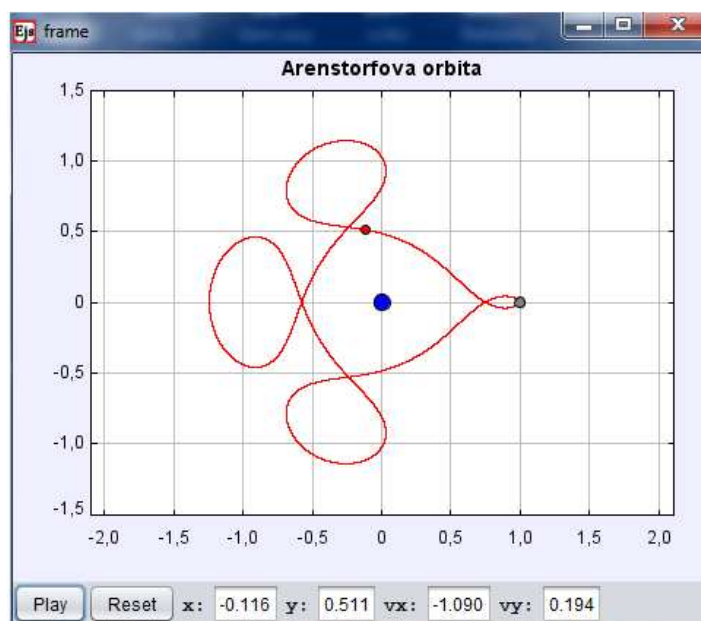
Arenstorfova orbitu objevil americký matematik *Richard F. Arenstorf* v době, kdy NASA pracovala na projektu Appolo. Zadání bylo jednoduché, najít periodickou dráhu v soustavě Země-Měsíc, tak aby astronautům byl umožněn bezpečný návrat z Měsíce zpátky na Zem a to bez použití motorů. Nespornou výhodou takového zadání je, že vesmírná loď má oproti Zemi a Měsíci zanedbatelnou hmotnost a proto zde lze rovnou aplikovat omezený problém tří těles. Vědci z NASA šli ještě dál, chtěli využívat Arenstorfovu orbitu jako cestu pro tzv. lunární autobus. Tento „autobus“ by mohl zásobovat stálou posádku kolonistů na Měsíci. Výhodou ideji toho vesmírného autobusu bylo, že nespotřebovává palivo, veškerou energii získává z gravitačního pole soustavy Země-Měsíc. Dnes je již projekt lunárního autobusu delší dobu zmražen z důvodů ukončení projektu Appolo v roce 1972 ([11]).

Arenstorfovy orbity se dělí podle počtu smyček. Nejzákladnější orbitou je typ „8“, další typy jsou tzv. dvousmyčková a třismyčková Arenstorfova orbita. Nehledě na to o jaký typ se jedná,

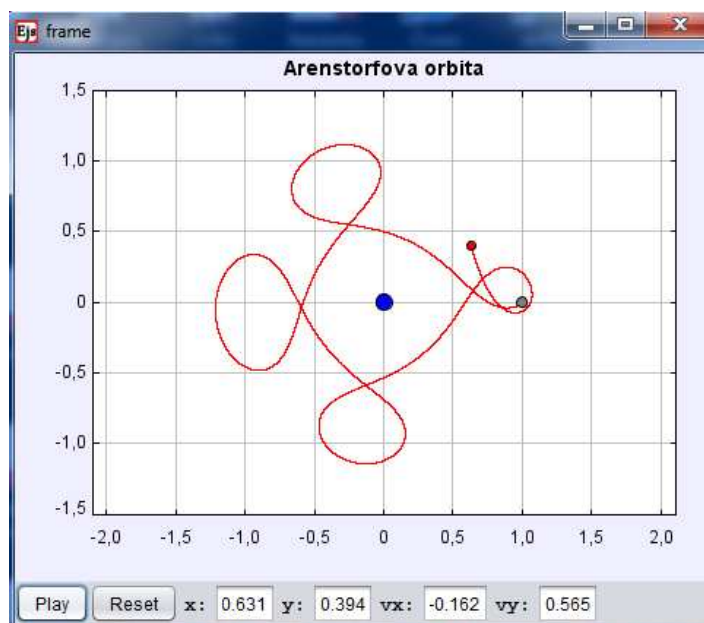
vždy vzniká Arenstorfova orbita v soustavě Země-Měsíc. Počáteční podmínky pro simulaci třísmyčkové orbity jsou (převzato z [6]):

$$\begin{aligned} x &= 0.994 & \dot{x} &= 0 \\ y &= 0 & \dot{y} &= -2.0015851063791 \end{aligned}$$

V soustavě Země-Měsíc volíme $\mu = 0.012277471144$



Obr. 18: Třísmyčková Arenstorfova orbita. Numerická metoda *Fehlberg 8(7)* při toleranci 10^{-11} .



Obr. 19: Třísmyčková Arenstorfova orbita. Numerická metoda *Euler-Richardsonova*.

Arenstorfova orbita má ještě jedno důležité využití. Je totiž velice citlivá na numerickou chybu, a proto jsme s její pomocí schopni ilustrovat přesnost numerické metody. Na obr. 19 byly použity ty stejné počáteční podmínky, byla pouze změněna numerická metoda. Chyba *Euler-Richardsonovy* metody zapříčiní neperiodické numerické řešení. Naštěstí program *EJS* nám umožňuje použití i jiných numerických metod. Na obr. 18 je použito metody *Fehlberg 8(7)* s absolutní tolerancí 10^{-11} . Při použití samotné *Eulerovy* numerické metody ani nedojde k vytvoření třísmýčkové orbity.

4.4 Vytvoření simulace omezeného problému tří těles v EJS

Při tomto popisu vytvoření simulace omezeného problému tří těles se zaměříme na záložky *Evolution*, *Fixed Relations* a *Initialization*, které můžeme právem nazývat „srdcem“ každé simulace vytvořené programem *EJS*. Nejprve bude potřeba upravit vztah (2.1.1) popřípadě vztah (2.1.2) do souřadnicových složek.

$$\begin{aligned}\frac{d^2x}{dt^2} - 2\omega \frac{dy}{dt} &= \frac{\partial\Omega}{\partial x} \\ \frac{d^2y}{dt^2} + 2\omega \frac{dx}{dt} &= \frac{\partial\Omega}{\partial y}\end{aligned}$$

Nyní zvolme jednotky délky a hmoty stejným způsobem jakým jsme je zavedli v první kapitole. Předešlé rovnice přejdou na tvar

$$\frac{d^2x}{dt^2} = x + 2\frac{dy}{dt} - \bar{\mu}\frac{x+\mu}{R_1} - \mu\frac{x-\bar{\mu}}{R_2} \quad (4.4.1a)$$

$$\frac{d^2y}{dt^2} = y - 2\frac{dx}{dt} - \bar{\mu}\frac{y}{R_1} - \mu\frac{y}{R_2} \quad (4.4.1b)$$

kde

$$\bar{\mu} = 1 - \mu$$

$$R_1 = ((x + \mu)^2 + y^2)^{3/2}$$

$$R_2 = ((x - \bar{\mu})^2 + y^2)^{3/2}$$

V jazyku Java jsou tyto diferenciální rovnice zapsány takto

$$\begin{aligned} dx/dt &= vx \\ dy/dt &= vy \\ dvx/dt &= x+2*vy-(1-\mu)*(x+\mu)/R1(x,y)-\mu*(x-1+\mu)/R2(x,y) \\ dvy/dt &= y+2*vx-(1-\mu)*y/R1(x,y)-\mu*y/R2(x,y) \end{aligned}$$

Kromě výše zmíněných třech podzáložek, budeme využívat podzáložku *Custom*. V *Custom* můžeme naprogramovat vlastní, programem *EJS* nepředdefinované, objekty a funkce, samozřejmě pomocí programovacího jazyku Java. Typickým příkladem je nadefinování funkcí $R1(x,y)$ a $R2(x,y)$. Funkci $R1(x,y)$ v *Custom* nadefinuju takto

```
public double R1 (double a1,double a2) {
return Math.pow((a1+mu)* (a1+mu)+a2*a2,1.5);}
```

Obdobně lze zavést funkci $R2(x,y)$. Rovnice (4.4.1) v java kódu tvoří základ podzáložky *Evolution*. Dále je potřeba v *Evolution* zvolit numerickou metodu, absolutní toleranci, nezávislou proměnnou, přírůstek nezávislé proměnné a v neposlední řadě se dá v *EJS* nastavit rychlost výpočtu. Postupně jsme volili *Fehlberg 8(7)*, $Tol=1e-15$, t , $dt=0.005$ a rychlost výpočtu $SPD=20$.

Podzáložka *Initialization* je rozdělena na dvě okna. V prvním okně zavádíme základní vztahy mezi hmotnostmi a vzdálenostmi. Další součástí tohoto okna jsou stabilní lagrangeovy body a přibližný výpočet nestabilních lagrangeových bodů, viz obr. 20. V druhém okně se nachází přesný výpočet nestabilních lagrangeových bodů a to pomocí numerické metody *půlení intervalů*. Aby tento výpočet byl obecný, to znamená, aby platil pro libovolný poměr hmotností a s libovolnou přesností, musíme použít cyklus *while*. Zmíníme zde konkrétní cyklus, slouží k výpočtu nestabilního bodu L_1 . Najít body L_1 , L_2 a L_3 znamená převést rovnici (2.3.4) na funkci a hledat průsečíky s osou y , obr. 2. Tato funkce

$$f(x) = \omega^2 x - \frac{\kappa m_1}{|x - a_1|^3} (x - a_1) - \frac{\kappa m_2}{|x + a_2|^3} (x + a_2)$$

je zapsána v *Custom* takto

```
public double funkce (double x1) {
return x1-(m1*Math.pow(Math.abs(x1+r1),-3))*(x1+r1)-
(m2*Math.pow(Math.abs(x1-r2),-3))*(x1-r2);}

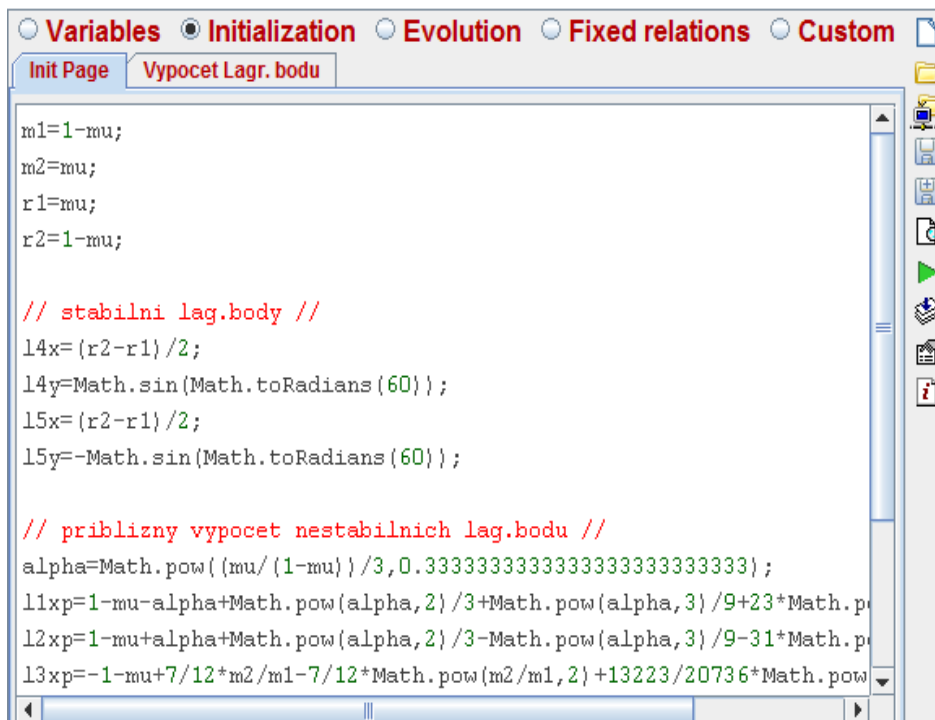
```

Konstrukce metody půlení intervalů v javě je

```
while (Math.abs(b-a)>=presnost) {
    l1x=(a+b)/2;
    if (funkce(l1x)*funkce(a)>0){a=l1x;}
    else { b=l1x }}

```

Příkaz *while* znamená, že cyklus poběží do té doby, než bude splněna podmínka v kulatých závorkách. Podmínka *if - else* přikazuje programu, aby funkční hodnoty krajních bodů měly různá znaménka. Proměnné *a* a *b* jsou krajními body jednotlivých intervalů. Počáteční hodnoty těchto krajních bodů jsou pro každý Lagrangeův bod různé a k jejich lokalizaci se s výhodou využívá znalosti přibližných hodnot poloh těchto bodů.



```

Variables Initialization Evolution Fixed relations Custom
Init Page Vypocet Lagr. bodu
m1=1-mu;
m2=mu;
r1=mu;
r2=1-mu;

// stabilni lag.body //
l4x=(r2-r1)/2;
l4y=Math.sin(Math.toRadians(60));
l5x=(r2-r1)/2;
l5y=-Math.sin(Math.toRadians(60));

// priblizny vypocet nestabilnich lag.bodu //
alpha=Math.pow((mu/(1-mu))/3,0.3333333333333333333333333333);
l1xp=1-mu-alpha+Math.pow(alpha,2)/3+Math.pow(alpha,3)/9+23*Math.p
l2xp=1-mu+alpha+Math.pow(alpha,2)/3-Math.pow(alpha,3)/9-31*Math.p
l3xp=-1-mu+7/12*m2/m1-7/12*Math.pow(m2/m1,2)+13223/20736*Math.pow

```

Obr. 20: První okno *Init Page* podzáložky *Initialization*

Poslední podzáložku, které se budeme věnovat je *Fixed relations*. V simulaci omezeného problému tří těles, které se tady věnujeme, je pohyb těles možno pozorovat jak v neinerciální vztažné soustavě, spojené s rotující soustavou, tak z pohledu pozorovatele v inerciální vztažné soustavě. Právě tento převod z jedné soustavy do druhé je zapsán v podzáložce *Fixed relations*. Označíme-li x a y jako souřadnice tělesa v inerciální soustavě, pak pro transformaci souřadnic platí

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos \omega t - y \sin \omega t \\ x \sin \omega t + y \cos \omega t \end{pmatrix}$$

Transformační rovnice jsou ve *Fixed relations* zapsány takto

```
x1 = x*Math.cos(omega*t)-y*Math.sin(omega*t);
y1 = y*Math.cos(omega*t)+x*Math.sin(omega*t);
```

Aby tato simulace vypadala co nejvíce realisticky, je potřeba vyřešit problém přiblížení těles a jejich následnou kolizi. Bylo by docela složité nasimulovat v *EJS* skutečnou srážku, na příklad dvou planet. Proto se omezíme, v případě střetu dvou těles, pouze na zastavení simulace. Tento problém budeme řešit pomocí příkazu *if*. Konstrukce podmínky zastavující simulaci není složitá

```
if ( /* zde napíšeme podmínku*/ ) {_pause();};
```

Za *if* napíšeme podmínku v java kódu, dříve než tak učiníme, je potřeba znát základní logické spojky v java jazyku. Obecně vypadají logické operátory konjunkce a disjunkce takto $\&$ a $|$. My bude používat tzv. podmíněné operátory $\&\&$ a $||$. Rozdíl mezi podmíněnými a nepodmíněnými operátory je v tom, že v případě podmíněných operátorů je výraz ukončen v momentě, kdy již není možno výsledek operace změnit. Objeví-li se na levé straně operátoru $\&\&$ pravdivostní hodnota *false*, operátor tuto hodnotu ihned vrací zpět, bez ohledu na to jaká pravdivostní hodnota se nachází na pravé straně. Analogická situace nastává pro operátor $||$ a pravdivostní hodnotu *true*. Slibovaná podmínka pro kolizi těles

```
(Math.abs(x-r2)<=0.01 && Math.abs(y)<=0.01) || (Math.abs(x+r1)<=0.01
&& Math.abs(y)<=0.01)
```

5. Obecný problém tří těles

Obecným problémem tří těles rozumíme pohyb tří hmotných bodů podrobených vzájemným přitažlivým silám. Na rozdíl od problému dvou těles, není analyticky řešitelný. Proto v následující kapitole bude tento problém řešen numericky a to jak v rovině, tak v prostoru.

5.1 Obecný problém tří těles v rovině

K vytvoření simulace problému tří těles vystačíme s Newtonovým gravitačním zákonem. Pohybové rovnice pro tento problém jsou

$$\begin{aligned}\ddot{x}_1 &= \kappa \cdot m_2 \cdot \frac{x_2 - x_1}{R_{12}^3} + \kappa \cdot m_3 \cdot \frac{x_3 - x_1}{R_{13}^3} \\ \ddot{y}_1 &= \kappa \cdot m_2 \cdot \frac{y_2 - y_1}{R_{12}^3} + \kappa \cdot m_3 \cdot \frac{y_3 - y_1}{R_{13}^3} \\ \ddot{x}_2 &= \kappa \cdot m_1 \cdot \frac{x_2 - x_1}{R_{12}^3} + \kappa \cdot m_3 \cdot \frac{x_2 - x_3}{R_{23}^3} \\ \ddot{y}_2 &= \kappa \cdot m_1 \cdot \frac{y_1 - y_2}{R_{12}^3} + \kappa \cdot m_3 \cdot \frac{y_3 - y_2}{R_{23}^3} \\ \ddot{x}_3 &= \kappa \cdot m_1 \cdot \frac{x_1 - x_3}{R_{13}^3} + \kappa \cdot m_2 \cdot \frac{x_2 - x_3}{R_{23}^3} \\ \ddot{y}_3 &= \kappa \cdot m_1 \cdot \frac{y_1 - y_3}{R_{13}^3} + \kappa \cdot m_2 \cdot \frac{y_2 - y_3}{R_{23}^3}\end{aligned}$$

kde

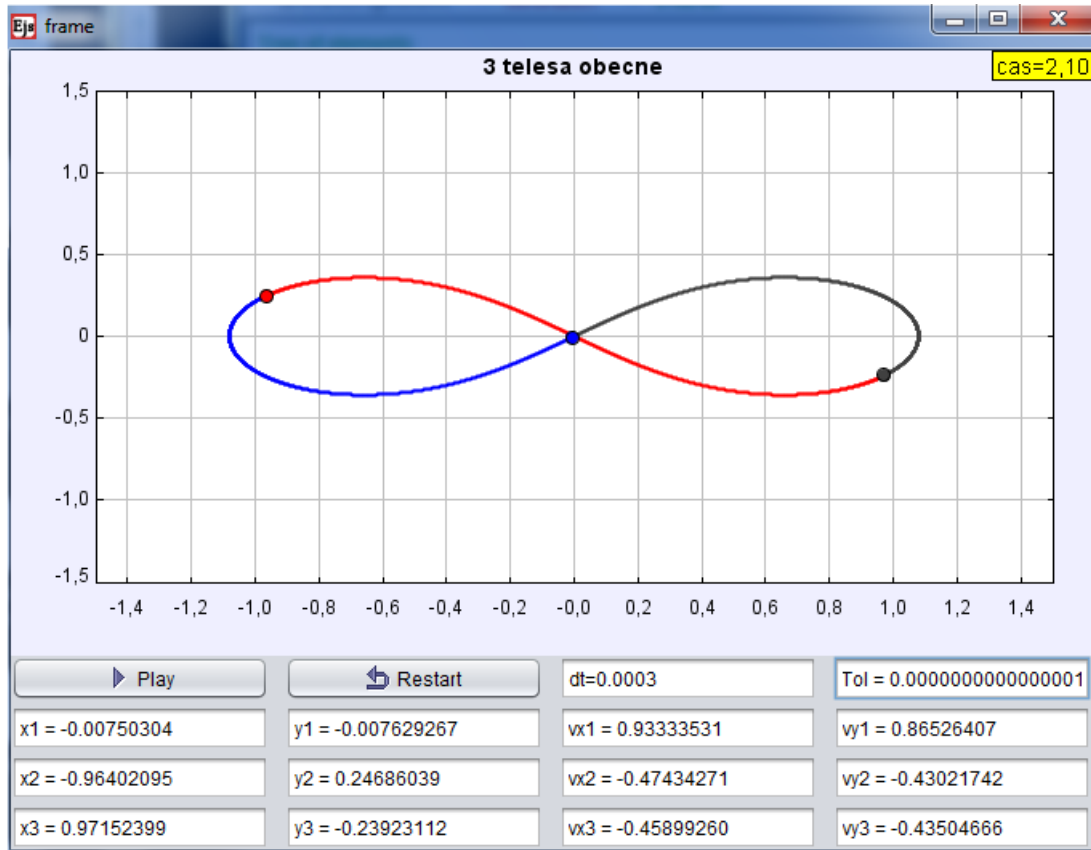
$x_k, y_k \dots$ je poloha k -tého tělesa, pro $k = 1, 2, 3$

$m_k \dots$ hmotnost k -tého tělesa

$R_{kj} \dots$ vzdálenost k -tého tělesa od j -tého tělesa, pro $j, k = 1, 2, 3, j \neq k$

Výše zmíněné pohybové rovnice získáme rozepsáním soustavy vektorových rovnic, které získáme z druhého Newtonova zákona a Gravitačního zákona.

Zajímavou simulací a současně dobrou zkouškou funkčnosti modelu obecného problému tří těles v rovině je simulace „osmičky“. Jedná se o simulaci tří těles o stejné hmotnosti, jejichž trajektorie má tvar ležaté osmičky, při vhodných počátečních podmínkách.

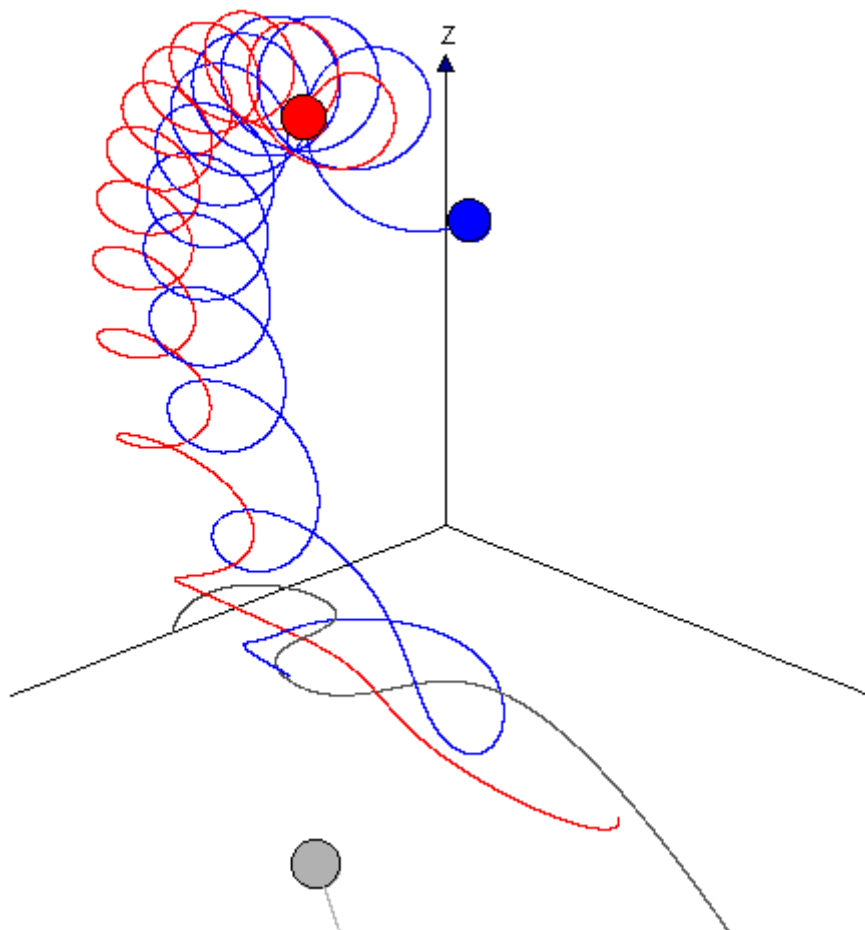


Obr. 21: Simulace „osmičky“.

Počáteční hodnoty pro tuto simulaci jsou (převzato z [10])

$$\begin{array}{ll}
 x_1 = -0.97000436 & \dot{x}_1 = -0.46620368 \\
 y_1 = 0.243087530 & \dot{y}_1 = -0.43236573 \\
 x_2 = 0.97000436 & \dot{x}_2 = -0.46620368 \\
 y_2 = -0.24308753 & \dot{y}_2 = -0.43236573 \\
 \\ \\
 x_3 = 0 & \dot{x}_3 = 0.93240737 \\
 y_3 = 0 & \dot{y}_3 = 0.86473146
 \end{array}$$

5.2 Obecný problém tří těles v prostoru



Obr. 22: Chaotický pohyb tří hmotných těles v prostoru.

V této podkapitole se budeme věnovat popisu programu simulace obecného problému tří těles v prostoru. Zároveň se zmíním, jak program vznikal. Záměrně jsem vynechal vznik simulace v rovině, protože v rovině je situace totožná, jen je o jednu souřadnici méně.

Začneme rozбором podzáložky *Evolution*, jádrem programu jsou pohybové rovnice, pohybové rovnice jsou stejné jako u problému v rovině (předešlá podkapitola 4.1), jen jejich počet je jiný, není jich šest, ale devět, každé těleso má ještě třetí souřadnici z . Jak jsem již výše zmínil, v *EJS* nemůžeme zadávat diferenciální rovnice druhého a vyšších řádů, proto pomocí substituce (kapitola 3), převedeme devět diferenciálních rovnic druhého řádu na dvojnásobný počet diferenciálních rovnic prvního řádu.

Výstup poté vypadá takto

$$dx1/dt = vx1$$

$$dy1/dt = vy1$$

$$dz1/dt = vz1$$

$$dx2/dt = vx2$$

$$dy2/dt = vy2$$

$$dz2/dt = vz2$$

$$dx3/dt = vx3$$

$$dy3/dt = vy3$$

$$dz3/dt = vz3$$

$$dvx1/dt = m2*(1/R12)*(x2-x1)+m3*(1/R13)*(x3-x1)$$

$$dvy1/dt = m2*(1/R12)*(y2-y1)+m3*(1/R13)*(y3-y1)$$

$$dvz1/dt = m2*(1/R12)*(z2-z1)+m3*(1/R13)*(z3-z1)$$

$$dvx2/dt = m1*(1/R12)*(x1-x2)+m3*(1/R23)*(x3-x2)$$

$$dvy2/dt = m1*(1/R12)*(y1-y2)+m3*(1/R23)*(y3-y2)$$

$$dvz2/dt = m1*(1/R12)*(z1-z2)+m3*(1/R23)*(z3-z2)$$

$$dvx3/dt = m1*(1/R13)*(x1-x3)+m2*(1/R23)*(x2-x3)$$

$$dvy3/dt = m1*(1/R13)*(y1-y3)+m2*(1/R23)*(y2-y3)$$

$$dvz3/dt = m1*(1/R13)*(z1-z3)+m2*(1/R23)*(z2-z3)$$

Stejně jak v simulaci omezeného problému tří těles v podzáložce *Custom* vytvoříme nové funkce *X*, *Y* a *Z*. Pomocí těchto funkcí pak v *Fixed relations* nadefinujeme *R12*, *R13* a *R23*. Toto nadefinování je velice praktické, jednak pro to, že nebudeme muset *R12*, *R13* a *R23* dlouze vypisovat v *Evolution*, ale také program *EJS* bude při numerickém výpočtu méně zatěžován a to povede k přesnějším výsledkům (na příklad simulace *Arenstorfovy orbity* by bez podobného zjednodušujícího zápisu nevznikla). Výstup podzáložky *Custom*

```
public double X (double X1, double X2) {
    return Math.pow(X1-X2,2);
}
public double Y (double Y1, double Y2) {
    return Math.pow(Y1-Y2,2);
}

public double Z (double Z1, double Z2) {
```

```

return Math.pow(Z1-Z2, 2);
}

```

Nadefinování $R12$, $R13$ a $R23$ ve *Fixed relations* pomocí funkcí X , Y , Z

```

R12=Math.pow(X(x1,x2)+Y(y1,y2)+Z(z1,z2),1.5);
R13=Math.pow(X(x1,x3)+Y(y1,y3)+Z(z1,z3),1.5);
R23=Math.pow(X(x2,x3)+Y(y2,y3)+Z(z2,z3),1.5);

```

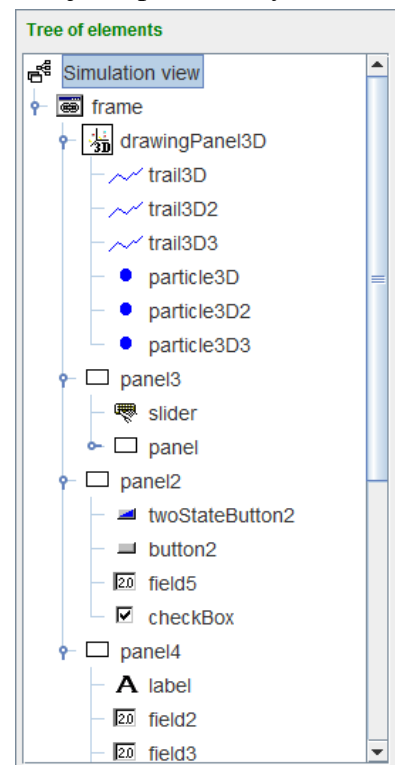
Funkce $R12$, $R13$ a $R23$ mají fyzikální význam jako třetí mocniny vzdálenosti. Tohoto využijeme při sestavení podmínky při kolizi těles.

```

if ((R12<=0.005) || (R13<=0.005) || (R23<=0.005)) {_pause();};

```

Tímto jsme zjednodušeně popsali vznik této simulace, vynechali jsme podzáložky *Variables* a *View* (obr. 23). Ve *Variables* jsou nadefinovány všechny proměnné, které vystupují ve výše zmíněných kódech. Za zmínku stojí proměnná *trace*, která se nevyskytuje v žádném z kódů. Je to proměnná typu *boolean*, to znamená, že nabývá pouze dvou hodnot a to *true* nebo *false*. Tato proměnná v simulaci dokáže schovávat (*false*) nebo odkrývat (*true*) trajektorii těles. Dalším zajímavým prvkem simulace je možnost přiblížení a oddálení. Opět jsme ve *Variables* nadefinovali novou proměnnou-*zoom*. Tato proměnná, typu *double*, je propojená s prvkem z *drawingPanel3D* podzáložky *View* (konkrétně s prvkem *Screen At*).



Obr. 23: Strom elementů, podzáložka *View*

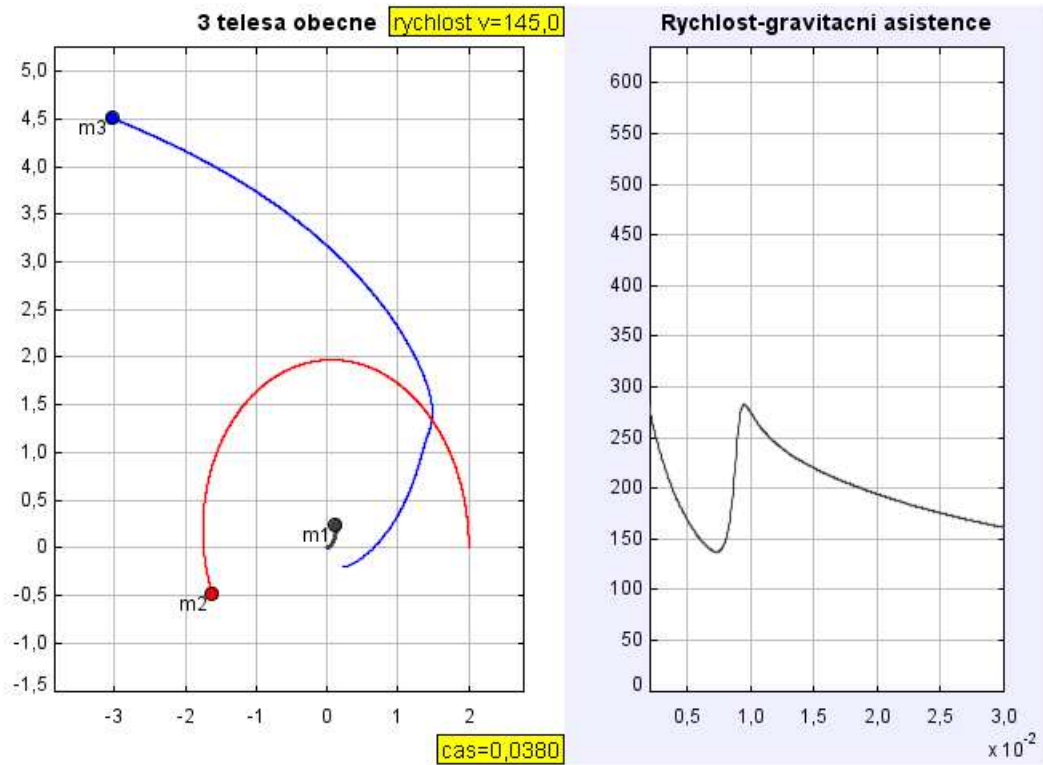
5.3 Simulace: Gravitační asistence

Závěr této kapitoly je věnován fenoménu nazvanému gravitační asistence. Jako příklad gravitační asistence můžeme uvést urychlení vesmírných sond *Pioneer* a *Voyager* „pomocí“ Jupiteru. Zatímco velikost rychlosti vzhledem k planetě před průletem a po průletu se ve velké vzdálenosti od planety nemění, rychlost sondy vůči Slunci se změní. Platnost zákona zachování energie je přitom zajištěna změnou rychlosti planety, která je ovšem vzhledem k její hmotnosti zanedbatelná. Tento manévr se často označuje jako *gravitační prak* (v odborné literatuře spíše jako *gravitační manévr*). Jev pro kosmonautiku a výzkum vesmíru velmi důležitý, protože výrazně šetří palivo a tím nákladné vesmírné projekty zlevňuje. Dalším případem je tzv. „gravitační zpomalení tělesa“. Tyto jevy jsou pro nás zajímavé, protože se jedná o problém tří těles. Analytický rozbor, včetně zajímavých příkladů, nalezneme čtenář např. v učebnici [4].

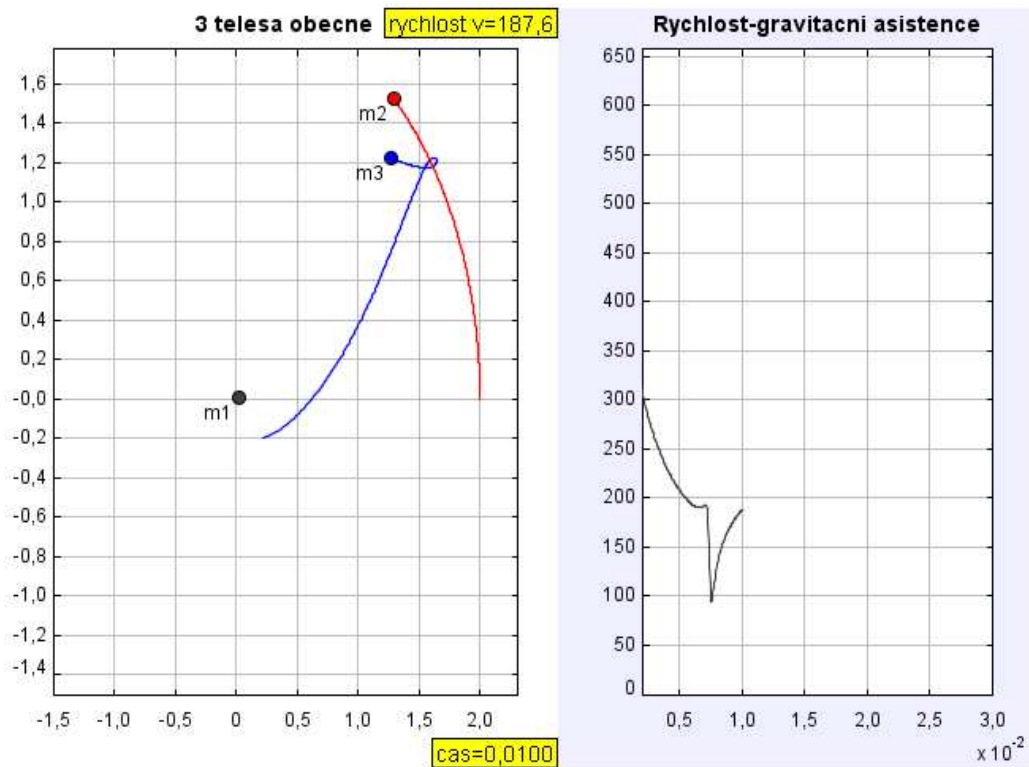
Všude bereme v úvahu, že $m_1 > m_2 > m_3$, $\kappa = 1$, a zároveň těleso m_1 se v čase $t = 0$ nachází v počátku soustavy souřadnic. Nebude-li uvedeno jinak, tak vše co je níže popsáno, platí jak pro *gravitační prak* (obr. 24), tak pro *gravitační zpomalení* (obr. 25). Zvolíme si jedno těžké těleso m_1 a druhé těleso m_2 kolem něj necháme obíhat po kruhové dráze. Zbylé těleso m_3 může představovat vesmírnou loď nebo sondu, jeho hmotnost bude tedy výrazně menší než hmotnosti předešlých těles m_1 a m_2 . Aby k výraznějšímu urychlení (zpomalení u *gravitačního zpomalení*) tělesa m_3 došlo, je potřeba nastavit počáteční podmínky tak, že těleso m_3 bude těsně míjet těleso m_2 a zároveň těleso m_3 proletí „za“ (pro *gravitační zpomalení* „před“) tělesem m_2 .

Počáteční podmínky pro tuto simulaci je možné nalézt jednoduchou úvahou. Tělesu m_3 , které vystřelíme z blízkosti tělesa m_1 , udělíme o něco menší rychlost v_3 než je *úniková rychlost* $v_{II} = \sqrt{2m_1/r} > v_3$, kde $r = \sqrt{x_3^2 + y_3^2}$. Rychlost v_3 nám zaručí, že se lehká sonda hned neotočí a nenarazí do těžkého tělesa m_1 . Zbývá nám určit rychlost kruhového pohybu tělesa m_2 kolem m_1 . Porovnáním gravitačního zákona a vztahu pro dostředivou sílu zjistíme počáteční rychlost tělesa m_2 , ve složkách $v_{x2} = \sqrt{m_1/x_2}$ a $v_{y2} = 0$. Poté zkusíme nalézt jednotlivé složky rychlosti třetího tělesa. Víme, že *x-ová počáteční složka rychlosti* je vázaná vztahem $v_{x3} = \sqrt{v_3^2 - v_{y3}^2}$, za *y-ovou počáteční složku rychlosti* v_{y3} postupně dosazujeme

hodnoty a zkusíme průběh simulace. Pro *gravitační prak* dobře sedí např. hodnota $v_{y3} = 95$ (pro *gravitační zpomalení* $v_{y3} = 135$).



Obr. 24: Simulace *gravitačního praku*. V pravé půlce je vidět graf závislosti rychlosti na čase.



Obr. 25: Simulace *gravitačního zpomalení*.

Závěr

Cílem mé bakalářské práce bylo vytvořit program simulující obecný problém tří těles. V teoretické části je zmíněn analyticky řešitelný problém dvou těles, dále se pak převážně věnuji popisu omezenému problému tří těles. Z literatury jsem vybíral některá zajímavá související témata a stručně jsem je v bakalářské práci popsal.

Určitou výzvou pak bylo v simulaci omezeného problému tří těles nalézt souřadnice nestabilních Lagrangeových bodů L_1, L_2, L_3 . Pro souřadnice těchto bodů není možné získat přesné analytické vyjádření, proto bylo potřeba použít vhodné numerické metody. Chtěl bych poděkovat prof. RNDr. Jiřímu Bajerovi, Csc., který mě navedl na metodu půlení intervalů. Tuto metodu jsem použil a s pomocí vedoucího práce jsme vytvořili cyklus, který vypočítá polohu nestabilních Lagrangeových bodů s dostatečnou přesností pro libovolné poměry hmotností těles.

Kapitolu 3 jsem věnoval programu Easy Java Simulation, ve kterém jsem naprogramoval všechny své simulace a vytvořil všechny obrázky. Na jednoduchém příkladu tělesa na pružině obeznámil čtenáře s prostředím tohoto programu. Ve zbytku práce jsem se vždy snažil vysvětlit každou důležitou část programového kódu. Záměrně jsem tomuto programu věnoval v práci velký prostor a to z důvodu, aby každý začínající uživatel mohl tuto práci využít jako „úvodní manuál“, neboť zmíněné prostředí u nás prozatím není příliš rozšířeno. Na internetu jsem nenašel jakýkoliv výukový materiál v českém jazyce, proto pro zájemce o tento program neovládající angličtinu může být text užitečný.

Kapitola 4 je obsahově nejrozsáhlejší. Kromě již výše zmíněné teorie omezeného problému tří těles jsou její součástí také screenshoty simulací týkajícího se omezeného problému. Za velice zajímavou považuji simulaci Arenstorfovy orbity.

Závěrečná kapitola prezentuje vytvořený program obecného problému tří těles a to jak ve dvou-souřadnicovém, tak i ve tří-souřadnicovém systému. Ve většině případů se potvrdilo, že numerické řešení studovaných úloh je velmi citlivé na přesné zadání počátečních podmínek i na použití vhodné numerické metody.

Všechny výstupy budou zveřejněny na webových stránkách a budou sloužit zájemcům o tuto problematiku.

Seznam použitých pramenů

- [1] *Algoritmus – Algoritmy.net*. Dostupné online: <http://www.algoritmy.net/>
- [2] Anderle P.: *Základy nebeské mechaniky*. Academia, Praha 1971.
- [3] Bajer J.: *Mechanika 1*. Univerzita Palackého, Olomouc 2004
- [4] Bajer J.: *Mechanika 2*. Univerzita Palackého, Olomouc 2004.
- [5] Brož M.: „Astronomický kurz (7) – Problém tří těles“. *Povětroň. Královéhradecký astronomický časopis* **16** 2008 (5), 4–18. Dostupné online: <http://sirrah.troja.mff.cuni.cz/~mira/ashk/povetron-2008-05.pdf>.
- [6] Celý J.: *Řešení fyzikálních úloh na mikropočítačích*. Masarykova universita, Brno 1990.
- [7] Esquembre F.: *Easy Java Simulations - The manual*. 2005. Dostupné online: http://www.phy.ntnu.edu.tw/oldjava/slovokia/Podpora/EjsManual_en_3.4_050914.pdf.
- [8] Montgomery R.: *On the N-body problem*. Dostupné online: <http://count.ucsc.edu/~rmont/Nbody.html>
- [9] Murray C. D., Dermott S. F.: *Solar System Dynamics*. Cambridge University Press, Cambridge 1999.
- [10] Pittich E.: *Astronomická ročenka 2010*. Slovenská ústredná hvězdáren, Hurbanovo 2009.
- [11] Wikipedia.: *Richard Arenstorf*. Online: http://en.wikipedia.org/wiki/Richard_Arenstorf