

ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

FAKULTA STROJNÍ



BAKALÁŘSKÁ PRÁCE

2D simulátor manévrování  
kosmických těles

2018

KATEŘINA  
MÄRZOVÄ

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Märzová** Jméno: **Kateřina** Osobní číslo: **459973**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav letadlové techniky**  
Studijní program: **Teoretický základ strojního inženýrství**  
Studijní obor: **bez oboru**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**2D simulátor manévrování kosmických těles**

Název bakalářské práce anglicky:

**2D simulator of orbital maneuvers**

Pokyny pro vypracování:

Cíl práce:

- Software demonstrující základní manévrování v kosmickém prostoru, vhodný jako doplněk výuky orbitální mechaniky
- Pro vypracování proveďte:
  - Vytvořte software (soubor skriptů, webovou aplikaci apod.) demonstrující základní mechaniku manévrování v kosmickém prostoru ve 2D (s možností pozdějšího rozšíření do 3D)
  - simulátor bude obsahovat idealizovaná keplerovská řešení i simulaci newtonovského gravitačního působení ve zjednodušeném modelu Sluneční soustavy
  - bude možno demonstrovat použití různých typů kosmických pohonů
- Vytvořte dokumentaci k použití a možnému dalšímu rozvoji simulátoru

Seznam doporučené literatury:

Malcolm Macdonald, Viorel Badescu (ed.): The International Handbook of Space Technology, Springer, 2014, ISBN 978-3-642-41101-4

Howard D. Curtis: Orbital Mechanics for Engineering Students, Elsevier, 2005, ISBN 0-7506-6169-0

Dále dle doporučení vedoucího práce

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Mgr. Jaroslav Kousal, Ph.D., ústav letadlové techniky FS**


Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **30.04.2018**

Termín odevzdání bakalářské práce: **13.08.2018**

Platnost zadání bakalářské práce:

  
Mgr. Jaroslav Kousal, Ph.D.  
podpis vedoucí(ho) práce


  
Ing. Robert Theiner, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

  
prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

26.04.2018  
Datum převzetí zadání

  
Podpis studentky

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího bakalářské práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků bakalářské práce nebo její podstatné části, pokud budu uvedena jako její spoluautor.

Dne .....

Podpis .....

## Poděkování

Chtěla bych poděkovat Mgr. Jaroslavi Kousalovi, Ph.D. za odborné vedení, vstřícnost při konzultacích a za možnost zpracovávat v bakalářské práci toto zajímavé téma.

## **Abstrakt**

Bakalářská práce obsahuje popis a řešení základních problémů astrodynamiky a manévrování kosmických těles. Zejména se zabývá řešením polohy, parametrů oběžných drah a elementárních impulzních manévřů. Vytvořený simulátor bude sloužit jako doplněk výuky orbitální mechaniky a bakalářská práce jako manuál k použití a případnému rozšíření.

## **Abstract**

Bachelor thesis contains description and solution of basic astrodynamics issues and space maneuvering. Especially, it deals with solution of position, parameters of orbits and elementary impulse maneuvers. Created simulator will serve as supporting educational tool for teaching of orbital mechanics and bachelor thesis may be used as manual for its use or further extension.

## **Klíčová slova**

2D simulátor, orbitální mechanika, oběžná dráha, manévrování, výuková pomůcka

## **Keywords**

2D simulator, orbital mechanics, orbit, maneuvering, teaching aid

# Obsah

<b>1</b>	<b>Teoretická část</b>	<b>8</b>
1.1	Základní fyzikální vztahy a zákony . . . . .	8
1.1.1	Keplerovy zákony . . . . .	8
1.1.2	Newtonovy zákony . . . . .	9
1.1.3	Newtonův gravitační zákon . . . . .	9
1.1.4	Případ pohybu dvou těles . . . . .	11
1.2	Oběžné dráhy a jejich parametry . . . . .	12
1.2.1	Hybnost a základní vzorce pro oběžné dráhy . . . . .	13
1.2.2	Základní parametry oběžné dráhy . . . . .	15
1.3	Vis - viva rovnice a rychlost . . . . .	16
1.4	Keplerova rovnice . . . . .	17
1.5	Kosmické pohony . . . . .	18
1.5.1	Specifický impulz $I_{sp}$ . . . . .	18
1.5.2	Druhy kosmických pohonů . . . . .	19
<b>2</b>	<b>Praktická část</b>	<b>20</b>
2.1	Programovací prostředí MATLAB . . . . .	20
2.1.1	Proč si vybrat MATLAB . . . . .	20
2.1.2	MATLAB - přehled základních použitých příkazů a funkcí . . . . .	20
2.1.3	Současná nabídka kosmických simulátorů . . . . .	21
2.2	Výpočet základních parametrů oběžné dráhy . . . . .	22
2.2.1	Skript - funkce pro výpočet parametrů oběžné dráhy . . . . .	22
2.3	Výpočet Keplerovy rovnice . . . . .	23
2.3.1	Skript - funkce pro výpočet Keplerovy rovnice dle předpisu (2.6) . . . . .	24
2.3.2	Skript - funkce pro výpočet excentrické anomálie z excentricity a periody . . . . .	25
2.4	Výpočet parametrů oběžných drah z radiusvektoru . . . . .	26
2.4.1	Postup výpočtu . . . . .	27
2.4.2	Skript - funkce pro výpočet parametrů oběžné dráhy . . . . .	28

2.5	Srovnání Hohmannova a Bi-eliptického manévru . . . . .	31
2.5.1	Hohmannův manévr . . . . .	31
2.5.2	Bi-eliptický manévr . . . . .	32
2.5.3	Výpočet změn rychlostí $\Delta V$ . . . . .	33
2.5.4	Skript -funkce pro výpočet změn rychlostí pro Hohmannův manévr . . . . .	34
2.5.5	Skript -funkce pro výpočet změn rychlostí pro bi-eliptický manévr . . . . .	34
2.5.6	Výpočet přechodových elips . . . . .	35
2.5.7	Skript - trajektorie Hohmannova manévru . . . . .	37
2.5.8	Skript - trajektorie bi-eliptického manévru . . . . .	37
2.5.9	Skript - Ciolkovského rovnice a demonstrace typu pohonu	39
<b>3</b>	<b>Závěr</b>	<b>42</b>
<b>4</b>	<b>Seznam obrázků</b>	<b>43</b>
	<b>Reference</b>	<b>44</b>
<b>5</b>	<b>Příloha - Skripty</b>	<b>45</b>
5.1	Výpočet základních parametrů oběžné dráhy . . . . .	45
5.2	Výpočet parametrů oběžné dráhy z radiusvektoru . . . . .	48
5.3	Hohmannův versus bi-eliptický manévr . . . . .	53

# 1 Teoretická část

## 1.1 Základní fyzikální vztahy a zákony

První část této práce se zabývá přehledem základních fyzikálních vztahů a zákonů, na kterých je simulátor postavený.

### 1.1.1 Keplerovy zákony

Keplerovy zákony jsou fyzikální zákony, které popisují pohyb planet, nebo obecněji těles, kolem hmotnějšího tělesa (Slunce). Tyto zákony odvodil německý astronom a matematik Johannes Kepler v první polovině 17. století.

1. Planety obíhají kolem Slunce po elipsách málo odlišných od kružnic, v jejich společném ohnisku je Slunce. [1]
2. Plochy opsané průvodičem planety za jednotku času jsou konstantní. [1]  
Průvodič (radiusvektor)  $r$  je úsečka pomyslně spojující planetu a Slunce. Pokud plochu opsanou průvodičem za jednotku času  $\frac{dA}{dt}$  označíme jako plošnou rychlost, můžeme říci, že plošná rychlost planety na oběžné dráze je konstantní.
3. Poměr druhých mocnin period  $T_1$  a  $T_2$  je přímo úměrný poměru třetích mocnin hlavních poloos  $a_1$  a  $a_2$ . [2]

$$\frac{T_1^2}{T_2^2} = \frac{a_1^3}{a_2^3} \quad (1.1)$$

Z třetího zákona vyplývá, že perioda  $T$  oběžné dráhy planety kolem Slunce závisí pouze na velikosti hlavní poloosy, tudíž:

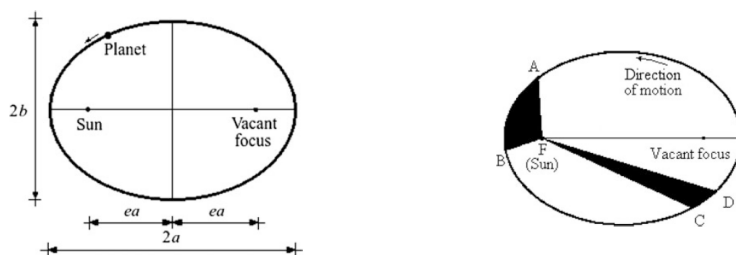
$$T = 2\pi \sqrt{\frac{a^3}{\mu}}, \quad (1.2)$$

kde  $\mu$  je gravitační parametr Slunce:

$$\mu = GM, \quad (1.3)$$



kde  $G$  je univerzální gravitační konstanta a  $M$  je hmotnost tělesa. Gravitační parametr Slunce  $\mu_S = 1,327 \cdot 10^{11} \text{ km}^3/\text{s}^2$ . Gravitační parametr Země je  $\mu_Z = 398600 \text{ km}^3/\text{s}^2$ .



Obrázek 1: 1. Keplerův zákon [3] Obrázek 2: 2. Keplerův zákon.[3]

### 1.1.2 Newtonovy zákony

Keplerovy zákony byly později ukázány jako speciální případy obecných zákonů pohybu, které formuloval Isaac Newton (1643–1727).

1. Těleso zůstává v klidu, nebo v pohybu rovnoměrném přímočarém, dokud není vnějšími silami donuceno tento svůj stav změnit.
2. Časová změna hybnosti je přímo úměrná síle a působí ve stejném směru.

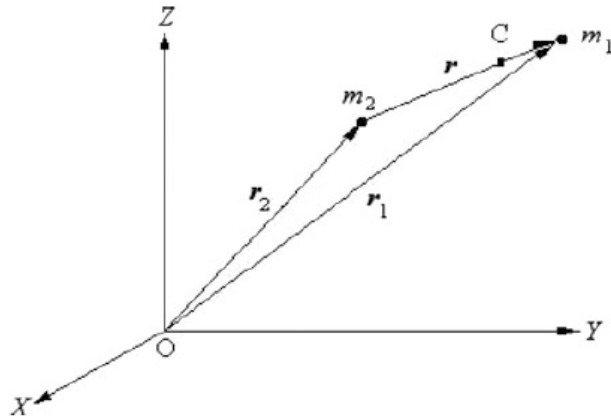
$$\vec{F} = \frac{d(m\vec{v})}{dt} = m\vec{a}, \quad (1.4)$$

kde  $\vec{a} = \frac{d\vec{v}}{dt}$  je zrychlení tělesa. Toto zjednodušení můžeme použít pouze tehdy, nedochází-li ke změně hmotnosti tělesa.

3. Silové působení hmotných těles je vždy vzájemné, síly jsou stejně velké a opačné orientované.

### 1.1.3 Newtonův gravitační zákon

Gravitační síla je příkladem nepřímé silové interakce, stejně jako například magnetická síla, to znamená, že nedochází k přímému kontaktu těles, ale



Obrázek 3: Působení hmotných bodů [3]

působí v určitých vzdálenostech. Pro zjednodušení uvažujeme tělesa jako hmotné body, tedy bezrozměrné, s veškerou hmotou koncentrovanou v těžišti tělesa.

Gravitační síla působící mezi dvěma hmotnými body  $m_1$  a  $m_2$ , které jsou od sebe vzdáleny  $r$  je potom:

$$F_g = G \frac{m_1 m_2}{r^2}, \quad (1.5)$$

kde  $G = 6,6742 \cdot 10^{11} \text{ m}^3/\text{kg} \cdot \text{s}^2$  je univerzální gravitační konstanta. Univerzální gravitační konstanta je obecně považována za nejobtížněji měřitelnou konstantu. S Newtonovým gravitačním zákonem jsou při dnešním chápání fyziky spojené některé problémy, jako například nekonzistentnost, nerelativistická rychlost interakce a podobné. Dnes ho nahrazuje mnoho nových přístupů, např. Einsteinova Teorie relativity nebo kvantová teorie pole. I přesto se ukazuje jako vynikající aproximace pro výpočet a je tím pádem naprosto postačující pro tento simulátor.

### 1.1.4 Příklad pohybu dvou těles

Aby se dalo ukázat, jak Keplerovy zákony sledují Newtonovy, zavedeme následující předpoklady:

- obě tělesa jsou reprezentovány hmotnými body majícími hmotu soustředěnou v těžištích,
- každé těleso je vystaveno pouze působení gravitační síly od druhého tělesa, která působí ve směru spojnice hmotných bodů.[3]

Následně můžeme popsat pohyb tělesa v inerciální (rovnoměrně se pohybující, nerotující) soustavě.

Mějme dvě izolovaná tělesa reprezentovaná hmotnostmi  $m_1$  a  $m_2$  a kartézský systém  $XYZ$  s počátkem  $O$  podle Obr. 3:

$$\vec{r} = \vec{r}_1 - \vec{r}_2 \quad (1.6)$$

Bod  $C$  je z definice těžiště hmoty  $m_2$  a proto můžeme vektor  $\vec{r}_C$  popsat:

$$\vec{r}_C = \frac{m_1\vec{r}_1 + m_2\vec{r}_2}{m_1 + m_2} \quad (1.7)$$

Po úpravách, dle [3], dostaneme:

$$\vec{r}_1 - \vec{r}_C = \frac{m_2}{m_1 + m_2} \vec{r} \quad (1.8)$$

$$\vec{r}_2 - \vec{r}_C = \frac{m_1}{m_1 + m_2} \vec{r} \quad (1.9)$$

Využitím Newtonova druhého zákona pohybu dostaneme síly  $\vec{F}_1$  a  $\vec{F}_2$  působící na hmotné body:

$$\vec{F}_1 = m_1\vec{a}_1 = m_1\vec{r}_1'' = \frac{m_1m_2}{m_1 + m_2} \vec{r}'' + m_1\vec{r}_C'' \quad (1.10)$$

$$\vec{F}_2 = m_2\vec{a}_2 = m_2\vec{r}_2'' = \frac{m_1m_2}{m_1 + m_2} \vec{r}'' + m_2\vec{r}_C'' \quad (1.11)$$

Z třetího Newtonova pohybového zákona vyplývá, že:  $F_1 + F_2 = 0$ , a tudíž:

$$m_1\vec{r}_C'' + m_2\vec{r}_C'' = 0 \quad (1.12)$$

Aby rovnice (1.12) platila, musí  $r_C'' = 0$ . Dosazením do rovnic (1.11) a (1.11):

$$\vec{F}_1 = m_1 \vec{a}_1 = m_1 \vec{r}_1'' = \frac{m_1 m_2}{m_1 + m_2} \vec{r}'' \quad (1.13)$$

$$\vec{F}_2 = m_2 \vec{a}_2 = m_2 \vec{r}_2'' = \frac{m_1 m_2}{m_1 + m_2} \vec{r}'' \quad (1.14)$$

Uvažujeme-li těleso  $m_1$  za hmotnější, tedy  $F_1$  jako gravitační sílu můžeme popsat vztahem:

$$\vec{F}_1 = -G \frac{m_1 m_2}{r^2} \frac{\vec{r}}{r}, \quad (1.15)$$

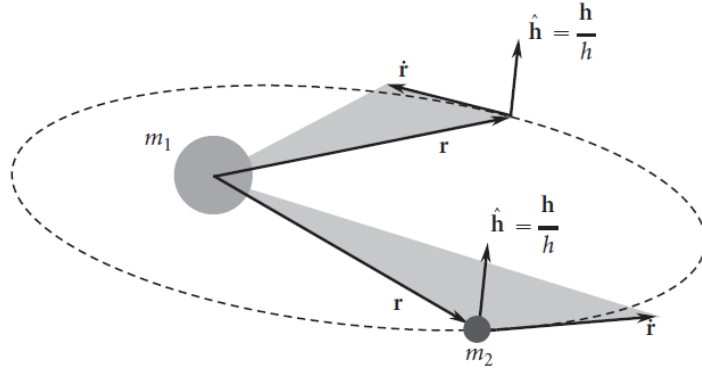
kde znaménko mínus značí přitažlivý charakter síly  $F_1$ , která působí proti směru vektoru  $\vec{r}$ . Zavedením gravitačního parametru  $\mu = G(m_1 + m_2)$  s využitím rovnic (1.12), (1.13) a (1.14) dostaneme výslednou základní rovnici pohybu:

$$\vec{r}'' + \frac{\mu}{r^3} \vec{r} = 0 \quad [3] \quad (1.16)$$

Pohybová rovnice (1.16) je výchozí rovnicí pro většinu výpočtů v astrodynamice. [4]

## 1.2 Oběžné dráhy a jejich parametry

Oběžné dráhy jsou trajektorie, po kterých se pohybuje těleso vesmírem, obvykle kolem těžiště hmotnějšího tělesa, nebo kolem těžiště soustavy [5]. Těžiště soustavy se v některé odborné literatuře označuje jako tzv. gravitační centrum. Oběžné dráhy jsou geometrické časově nezávislé křivky a mohou být otevřené nebo uzavřené. Skutečná trajektorie vesmírného tělesa se bude odlišovat od své nominální oběžné dráhy (vypočítané ideální) kvůli pertubracím a odchýlkám neuvažovaným při výpočtu nominální.



Obrázek 4: Trajektorie  $m_2$  kolem  $m_1$  v rovině kolmé na na vektor  $\vec{h}$  [6]

### 1.2.1 Hybnost a základní vzorce pro oběžné dráhy

Základní fyzikální veličinou, kterou můžeme popsat pohyb tělesa na oběžné dráze, je hybnost  $\vec{H}$ . Hybnost  $\vec{H}_{2,1}$  tělesa  $m_2$  vzhledem k tělesu  $m_1$ :

$$\vec{H}_{2,1} = \vec{r} \times m_2 \vec{r}', \quad (1.17)$$

kde  $\vec{r}' = \vec{v}$  je rychlost  $m_2$  vůči  $m_1$ . Pro výpočty v astrodynamice se zavádí jednotková hybnost  $h = \frac{H_{2,1}}{m_2}$  s jednotkou  $km^2s^{-1}$ , a platí:

$$\vec{h} = \vec{r} \times \vec{r}' \quad [3] \quad (1.18)$$

Pokud jednotkovou hybnost zderivujeme podle času:

$$\begin{aligned} (\vec{r} \times \vec{r}' = 0) \\ \frac{d\vec{h}}{dt} &= \vec{r}' \times \vec{r}' + \vec{r} \times \vec{r}'' = 0 + \vec{r} \times \vec{r}'' \end{aligned} \quad (1.19)$$

$$\vec{r}'' = -\frac{\mu}{r^3} \vec{r} \quad (1.20)$$

$$\frac{d\vec{h}}{dt} = \vec{r} \times \vec{r}'' = \vec{r} \times \left(-\frac{\mu}{r^3} \vec{r}\right) = 0 \quad (1.21)$$

Tudíž v jakémkoli čase máme daný vektor  $\vec{r}$  a vektor rychlosti  $\vec{r}'$  ležící ve stejné rovině je dle rovnice (1.21) vektor hybnosti  $\vec{h}$  konstantní. Zavedeme-li

$\hat{u}_r$  jako jednotkový vektor v radiálním směru a  $\hat{u}_n$  jako jednotkový normálový vektor, dále radiální rychlost  $\vec{v}_r = v_r \hat{u}_r$  a normálovou složku rychlosti  $\vec{v}_n = v_n \hat{u}_n$ , dle obrázku 5, můžeme hybnost vyjádřit jako [4]:

$$\vec{h} = r \hat{u}_r \times (v_r \hat{u}_r + v_n \hat{u}_n) = r v_n \hat{h} \quad (1.22)$$

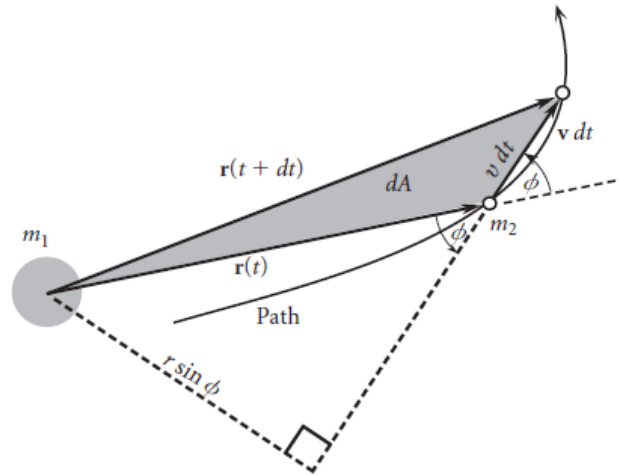
$$h = r v_n \quad [4] \quad (1.23)$$

Z rovnice 1.23 je zřejmé, že úhlová hybnost závisí pouze na normálové složce relativní rychlosti.

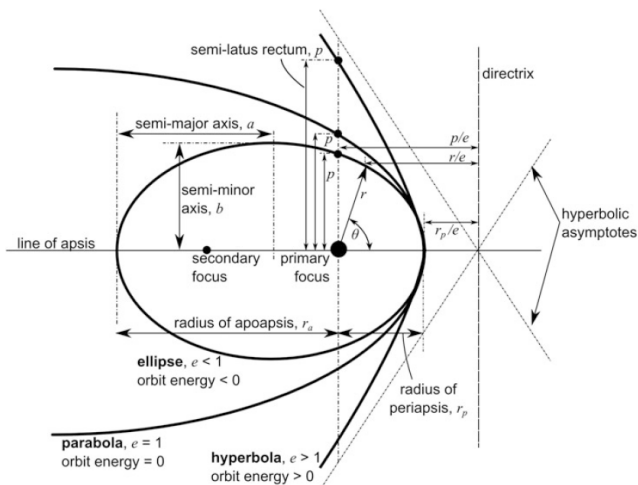
Za čas  $dt$  vektor polohy  $\vec{r}$  "vytýčí" plochu  $dA$ . Z Obr. 5 je zřejmé, že lze plochu  $dA$  popsat jako  $dA = \frac{1}{2} r v_n dt$  a dále dosazením do rovnice rovnice (1.23):

$$\frac{dA}{dt} = \frac{h}{2}, \quad [3] \quad (1.24)$$

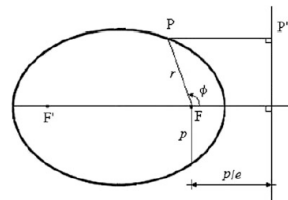
jak bylo již uvedeno dříve,  $\frac{dA}{dt}$  se označuje jako plošná rychlost a vede nás opět k druhému Keplerovu zákonu, který hovoří o tom, že tato plošná rychlost je v čase konstantní.



Obrázek 5: Diferenciální plocha  $dA$  dána relativní polohou vektoru polohy  $\vec{r}$  za časový úsek  $dt$  [6]



Obrázek 6: Kuželosečky [4]



Obrázek 7: Polární souřadnice [3]

### 1.2.2 Základní parametry oběžné dráhy

Všechny oběžné dráhy jsou kuželosečky. Kuželosečky mají dvě ohniska, v astrodynamice je gravitační centrum soustředěno právě v jednom z ohnisek, označuje se jako primární. Speciálním případem je kružnice a parabola. U kružnice jsou obě ohniska shodná, u paraboly je jedno ohnisko vzdáleno v nekonečnu. Velikost kuželosečky je dána hlavní  $a$  a vedlejší  $b$  poloosou a excentricitou (výstředností)  $e$ .

Další pojmy používané v souvislosti s popisem oběžných drah je apocentrum a pericentrum, které označují nejvzdálenější, respektive nejbližší, místo oběžné dráhy vzhledem k primárnímu ohnisku, viz Obr. 6. Tyto pojmy se používají modifikované pro různá vesmírná tělesa, např. apo- a perihelium pro Slunce a apo- a perigeum pro Zemi.

Poloha tělesa oběžné dráze je dána skutečnou anomálií  $\theta$ , která je měřena jako úhlová odchylka polohového vektoru od pericentra, viz Obr. 7. Vztahy pro základní parametry oběžných drah tedy jsou:

- radius (poloměr) pericentra:

$$r_p = a(1 - e) \quad [6] \quad (1.25)$$

- radius (poloměr) apocentra:

$$r_a = a(1 + e) \quad [6] \quad (1.26)$$

- excentricita (vždy kladná):

$$e = \frac{\sqrt{a^2 - b^2}}{a} = \frac{r_a - r_p}{r_a + r_p} \quad [6] \quad (1.27)$$

Dalším důležitým parametrem kuželoseček je semi-latus rectum, nebo-li parametr kuželosečky [4].

$$p = \frac{h^2}{\mu} = a(1 - e^2) \quad (1.28)$$

Převedením předchozích rovnic pohybu do polárních souřadnic se dostaneme k tzv. rovnici oběžné dráhy (v anglické literatuře *Orbital equation*):

$$r = \frac{p}{1 + e \cos(\theta)}, \quad [4] \quad (1.29)$$

která je pro výpočty výhodnější.

### 1.3 Vis - viva rovnice a rychlost

Rychlost  $\vec{v}$  tělesa na oběžné dráze je tečná k trajektorii. Rychlost můžeme obecně vyjádřit jako:

$$\vec{v}^2 = \dot{r}^2 + r^2\dot{\theta}^2 \quad [3] \quad (1.30)$$

S využitím této rovnice a základních vztahů pro kuželosečky dostaneme *vis - viva* rovnici:

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \quad (1.31)$$

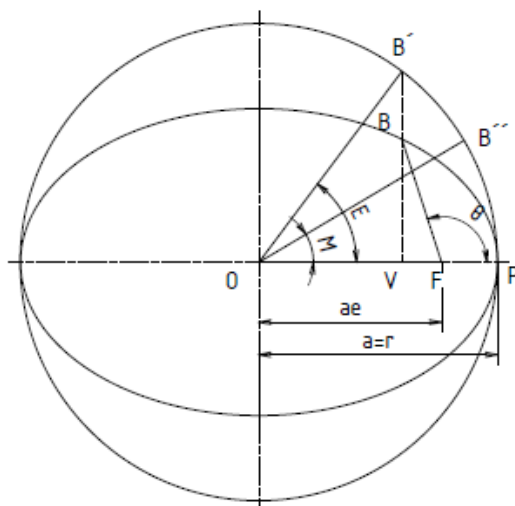
Tato rovnice se v některé literatuře označuje také jako energetická, jelikož  $\frac{1}{2}v^2$  a  $-\mu/r$  jsou kinetická, respektive potenciální energie vztažená k jed-



notce hmotnosti daného tělesa obíhajícího kolem hmotného středu (tedy například Země obíhající kolem Slunce). Potenciální energie je záporná, jelikož gravitační síla má přitažlivý charakter a jednotková potenciální energie je definována jako nulová pro planetu v nekonečné vzdálenosti od Slunce. [3]

## 1.4 Keplerova rovnice

Keplerova rovnice vyjadřuje vztah mezi střední anomálií  $M$  a excentrickou anomálií  $E$ . Keplerova rovnice má různé podoby v závislosti na různých druzích oběžných drah.



Obrázek 8: Schéma eliptické oběžné dráhy a jí opsané kruhové dráhy

Keplerova rovnice popisuje závislost času a polohy na tvaru eliptické oběžné dráhy a umožňuje zjistit polohu planety na oběžné dráze v daném čase.

Nejlépe lze vysvětlit Keplerovu rovnici z obr. 8, kde je kromě eliptické oběžné dráhy po které obíhá těleso (bod  $B$ ), znázorněna pomocná kruhová dráha o poloměru shodném s velikostí hlavní poloosy elipsy  $a$ , po které "obíhá" tzv. pomocné těleso (bod  $B''$ ), které má stejnou periodu oběhu. Když spustíme kolmici z bodu  $B$  k hlavní poloose, protne kružnici v bodě  $B'$ . Střed elipsy

i kružnice opsané označíme  $O$ , primární ohnisko elipsy  $F$ , pericentrum  $P$  a průsečík přímky  $B'B$  a  $OP$  označíme  $V$ , potom můžeme definovat potřebné úhly:

- úhel  $\sphericalangle B'OP$  - excentrická anomálie  $E$ ,
- úhel  $\sphericalangle BFP$  - pravá anomálie  $\theta$ ,
- úhel  $\sphericalangle B''OP$  - střední anomálie. [3] [4]

Výsledná Keplerova rovnice má tvar:

$$M = E - e \sin E \quad [4] \quad (1.32)$$

$$M = \frac{2\pi t}{T}, \quad [6] \quad (1.33)$$

kde  $t$  je čas a  $T$  je perioda oběhu.

## 1.5 Kosmické pohony

K pohonu kosmických lodí se používají raketové motory. Zjednodušeně lze říci, že raketové motory pracují na podobném principu jako klasické proudové, tzn. odcházející "spaliny" vytváří tah a tím pohání raketu dopředu. Na rozdíl od klasických reaktivních motorů však není (a ze své podstaty ani nemůže být) závislý na přísunu atmosférického kyslíku.

### 1.5.1 Specifický impulz $I_{sp}$

Veličina, která se používá k popisu tahu, dalo by se říci palivové účinnosti, pohonu je specifický impulz  $I_{sp}$ . Je to impulz, který dá hmotností jednotka paliva, nebo množství paliva spálené za jednotku času, jednotkou je  $Ns/kg$ . Specifický impulz je dán rovnicí:

$$I_{sp} = \frac{I_T}{W_T} = \frac{F}{\dot{W}}, \quad [7] \quad (1.34)$$

kde  $W_T$  je celková hmotnost spáleného paliva,  $\dot{W}$  je hmotnostní tok paliva. Specifický impulz se používá k výpočtu ideálního zisku rychlosti  $\Delta v_I$ . Ideální

$\Delta v_I$  je dán tzv. Ciolkovského rovnicí:

$$\Delta v_I = I_{sp} \ln \left( \frac{m_0}{m} \right), \quad [6] \quad (1.35)$$

$m_0$  je hmotnost paliva na počátku a  $m$  je hmotnost zbylého paliva po provedení manévru.

### 1.5.2 Druhy kosmických pohonů

Pohony se dělí na dvě hlavní skupiny: chemické a fyzikální. Chemické raketové motory využívají k pohonu energii vznikající chemickými procesy (hoření, slučování, rozklad, apod.). Dělí se na tuhé pohonné látky (TPL), kapalně pohonné látky (KPL) a hybridní. Dále je můžeme rozdělit dle počtu reagujících složek na jednosložkové (monergoly), dvousložkové (dierygoly) a vícesložkové (polyergoly).

Druh paliva	$I_{sp}[Ns/kg]$	Použití na motory
homogenní TPL	2 300	vojenská technika
heterogenní TPL	2 500	urychlovací startovní
kapalný kyslík a vodík	4 000	univerzální
kapalný kyslík a raketový petrolej	3 200	univerzální
hybridní pohonné látky	2 500	suborbitální nosiče, vývoj
elektrotermální	až 8 000	
motory na stlačený plyn	až 2 000	orientační a stabilizační
iontové	20 000 až 40 000	orientační a stabilizační, experimentálně pohon sond
jaderné	8 000	vývoj

Tabulka 1: Druhy pohonných hmot [8] [7]

Zdrojem primární energie pro fyzikální pohony není energie z paliva, ale energie z vnějších zdrojů, dosahuje vysokých  $I_{sp}$ . Jedná se o pohony iontové, jaderné, pohony využívající stlačený vzduch a podobně. [8] [7] [3]

## 2 Praktická část

### 2.1 Programovací prostředí MATLAB

MATLAB (Matrix Laboratory) je interaktivní programovací prostředí vyvíjené společností MathWorks. MATLAB je v první řadě používán k výpočtům, simulacím, implementaci algoritmů a analýze dat. Proměnné v MATLABu nemají pevný datový typ a mění se v průběhu výpočtu, nevyžadují deklaraci, vzniknou přiřazením hodnoty. Základním druhem všech proměnných je matice.

#### 2.1.1 Proč si vybrat MATLAB

MATLAB jsem zvolila jako programovací prostředí z několika důvodů:

1. Je jednoduchý, přehledný a po krátkém "zaškolení" se ho naučí používat téměř každý. Zejména studenti Strojní fakulty ocení známé prostředí.
2. Ze zadání vyplývá, že účelem této práce je vytvořit 2D simulátor s možností rozšíření do 3D a možností úpravy výpočtů pro konkrétní výpočty. Jelikož MATLAB pracuje s proměnnými jako s maticemi  $m \times n$ , je práce s 2D, popřípadě 3D proměnnými naprosto identická.
3. Skript psaný v MATLABu, ačkoli není plně kompatibilní, je snadno upravitelný pro spuštění ve volně šiřitelných programech typu Octave, Mathematica, Maple, SciLab, Python, apod.
4. Některé novější verze MATLABu taktéž umožňují převést program, napsaný ve formě funkce (m-file), do samonosné aplikace, která ke svému spuštění MATLAB nepotřebuje. Studentská licence ČVUT tuto funkci bohužel nepodporuje.

#### 2.1.2 MATLAB - přehled základních použitých příkazů a funkcí

[0:0.1:10] - vytvoří vektor o prvcích [0, 0.1, 0.2, ...]

[x, y]=pol2cart(fi, r) - převod polárních souřadnic  $fi$  a  $r$  do kartézských  $x$  a  $y$

$A(:, 2)$  - vybere z matice  $A$  2. řádek  
 abs - absolutní hodnota  
 clear - vymaže proměnnou, příkaz, okno ...  
 comet(x, y) - vykreslí jednotlivé body vstupních matic  $x$  a  $y$ , animace  
 cross(v, w) - vytvoří vektorový součin, tedy vektor, který je kolmý k vektoru  $\vec{v}$  a  $\vec{w}$   
 disp(n) - ukáže proměnnou  $n$   
 dot(A, B) - skalární součin matic  $A$  a  $B$   
 fprintf('Proměnná a: %g', a) - vypíše text a hodnotu proměnné  $a$   
 function  $[y_1, y_2, \dots, y_N] = \text{mojefunkce}(x_1, x_2, \dots, x_N)$  - definuje funkci *mojefunkce*, která bere vstupy  $y_1, \dots, y_N$  a vrací výstupy  $x_1, \dots, x_N$   
 input('Napiš číslo') - načte z klávesnice hodnotu, ukáže text  
 norm(v) - norma (velikost) vektoru  $v$   
 plot(x, y) - graf závislosti  $x$ ,  $y$   
 sin(x) - počítá hodnotu  $\sin(x)$ , analogicky  $\cos(x)$ ,  $\tan(x)$ ,  $\sinh(x)$ ,  $\cosh(x)$ , ...  
 sqrt(B) - vrací odmocninu z proměnné  $B$  [9]

### 2.1.3 Současná nabídka kosmických simulátorů

V současnosti existuje velké množství různých kosmických simulátorů, zaměřují se na různé problémy a cílí na různé skupiny uživatelů. Jako příklad bych ráda uvedla:

- Space Engine - dostupný na: <http://www.spaceengine.org/>
- Universe Sandbox - dostupný na: <http://www.universesandbox.com/>
- Orbiter Space Flight Simulator - dostupný na:  
<http://www.orbit.medphys.ucl.ac.uk/>
- Kerbal Space Program - dostupný na:  
<https://www.kerbalspaceprogram.com/en/>

Mým cílem nebylo vytvořit simulátor, který by se těmito graficky propracovaným aplikacím mohl rovnat, ale vytvořit program demonstrující způsob

a postup výpočtu základních astrodynamických problémů, jako je poloha, oběžná dráha, přelety apod. Výhodou oproti těmto programům je absolutní transparentnost a snadná upravitelnost programu (výpočtu) pro konkrétní využití. Díky tomu se dá v budoucnu rozšířit a vylepšit.

## 2.2 Výpočet základních parametrů oběžné dráhy

Jako první příklad jsem vybrala základní výpočet parametrů eliptické oběžné dráhy. Skript se dá jednoduše upravit pro libovolné veličiny. Ze vstupních parametrů: excentricity  $e$  a vzdálenost pericentra  $z_p$  od povrchu planety se dopočítá hybnost na oběžné dráze  $h$ , rychlosti v apocentru  $v_a$  a pericentru (o poloměru  $polo$ )  $v_p$  a perioda oběhu. Uživatel si nejprve vybere, kolem kterého tělesa Sluneční soustavy se chce pohybovat (Merkur, Země, Slunce, atd.), dále zadá  $z_p$ , tedy nejmenší vzdálenost mezi povrchem tělesa a oběžnou drahou, nakonec výstřednost (excentricitu) elipsy. Vzorce a vztahy pro výpočet jsou popsány v odstavci 1.2.

Simulátor obsahuje funkci `[h, vp, va, T, a, b]=zpegiven(zp, e, polo)` a skript, který s použitím této funkce načítá vstupní hodnoty a vytiskne výsledky a grafický výstup viz Obr. 9. Kompletní skript je v Příloze - Skripty, odst. 5.1.

### 2.2.1 Skript - funkce pro výpočet parametrů oběžné dráhy

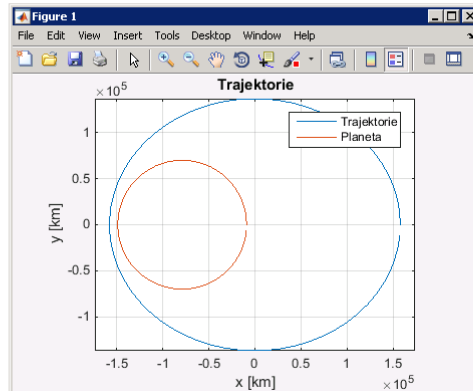
```
function [h, vp, va, T, a, b]=zpegiven(zp, e, polo)
rp=polo+zp; % radius v pericentru
h=(rp*mu*(1+e))^0.5; % hybnost
vp=h/rp; % rychlost v pericentru
ra=h^2/(mu*(1-e)); % radius v apocentru
a=(rp+ra)/2; % velikost hlavní poloosy
va=h/ra; % rychlost v apocentru
T=(2*pi/mu^2)*(h/sqrt(1-e^2))^3; % perioda oběhu
b=sqrt(a^2-(e*a)^2); % vedlejší poloosa
end
```

```

Tento skript pocita parametry obezne elipticke drahy ze zadane velikosti periapsidy a excentricity
Zvolete kolem ktereho telesa se pohybujeme.
1 - Merkur, 2 - Venuse, 3 - Zeme, 4 - Mars, 5 - Jupiter, 6 - Saturn, 7 - Uran, 8 - Neptun, 9 - Pluto, 10 - Slunce
Zvolte cislo:
5
Zadejte delku periapsidy zp: 9000
Zadejte vystrednost elipticke drahy e: 0.5

Hybnost:          h = 3.87239e+06
Rychlost v periapside: vp = 49.0729
Rychlost v apoapside: va = 16.3576
Perioda obehu:    T = 34999.8>>

```



Obrázek 9: Výpočet parametrů oběžné dráhy pro  $z_p = 9000 \text{ km}$  a  $e = 0,5$  kolem Jupitera

## 2.3 Výpočet Keplerovy rovnice

Výpočet Keplerovy rovnice, popsané výše v odstavci 1.4, je základním stavebním kamenem pro animace pohybu po oběžných drahách. Protože je k vykreslování animací použita funkce `comet`, která vykresluje trajektorii po jednotlivých bodech, hledala jsem způsob, jak zohlednit nekonstantní průběh rychlosti. Jelikož je úhlová rychlost konstantní, rozdělíme-li periodu oběhu na stejné díly a dopočítáme-li pro každý díl periody příslušnou polohu, výsledná animace bude teoreticky tvořit iluzi zrychlování, popřípadě zpomalování dle aktuální polohy. Rychlost vykreslování lze upravit přidáním příkazu `pause()` do `for` cyklů funkce `comet`.

Při znalosti  $E$  lze souřadný systém umístit do středu elipsy  $O$  dle Obr. 8, a trajektorii (elipsu) popsat analyticky jako:

$$x = a \cos(E) \quad (2.1)$$

$$y = b \sin(E) \quad (2.2)$$

Řešení Keplerovy rovnice provedeme pomocí Newtonovy metody, dle obec-

ného předpisu:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad [10] \quad (2.3)$$

Dle předpisu označíme:

$$f(E) = E - e \sin E - M \quad (2.4)$$

$$f'(E) = 1 - e \cos E \quad (2.5)$$

Dosazením do rovnice (2.3) poté dostaneme předpis pro výpočet:

$$E_{i+1} = E_i - \frac{E_i - e \sin E_i - M}{1 - e \cos E_i} = E_i - P \quad (2.6)$$

Nejprve určíme počáteční aproximaci pro  $E_0$ , například jako  $E_0 = M$ . Poté opakujeme výpočet dle předpisu (2.6), dokud absolutní hodnota zlomku  $P$  nebude menší než požadovaná chyba výpočtu. Dále můžeme s ohledem na rychlost výpočtu stanovit maximální počet iterací, které může program provést, dokud se výpočet nepřerušší.

### 2.3.1 Skript - funkce pro výpočet Keplerovy rovnice dle předpisu (2.6)

```
function E=keplerE(E, M) % funkce pro řešení Keplerovy
                        % rovnice E-e*sin(E)=M

error=1.e-8;           % volba přesnosti výpočtu
if M<pi                % pokud bude M menší než π
E=M+e/2;               % volba počáteční hodnoty E
else
E=M-e/2;               % pokud bude M ≥ π
                        % volíme počáteční hodnotu E

end

nMAX=1000;             % maximální počet iterací
n=0;                   % počáteční hodnota n
P = 1;                 % zvolíme počáteční hodnotu zlomku P
```



```

while abs(P)>error & n < = nMAX
                                % výpočet probíhá dokud abs. hodnota poměru
                                % P je větší než zvolená chyba, nebo se
                                % nedosáhne zvoleného počtu iterací
n=n+1;                            % počet iterací
P=(E-sin(E)-M)/(1-e*cos(E)); % počítá zlomek dle předpisu (2.6)
E=E-P;                            % výpočet nové hodnoty E
end

```

### 2.3.2 Skript - funkce pro výpočet excentrické anomálie z excentricity a periody

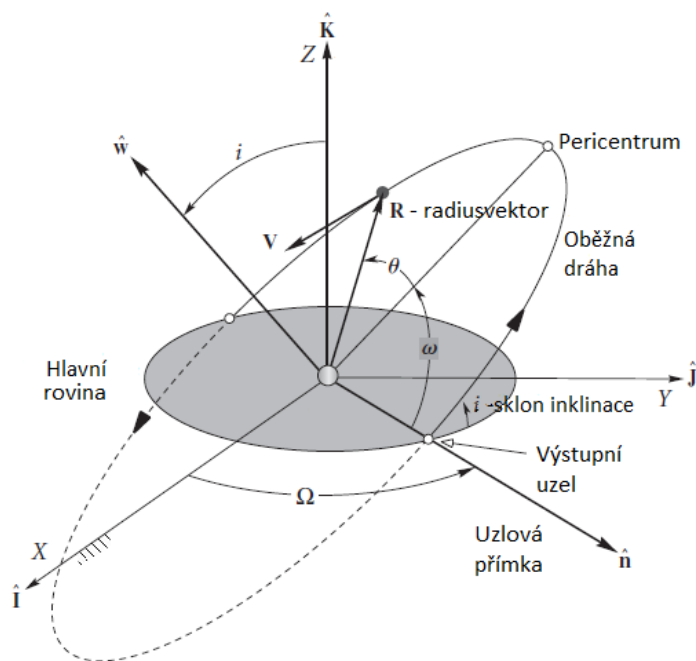
Analogicky můžeme počítat excentrickou anomálii  $E$ , pokud známe excentricitu  $e$  a periodu oběhu  $T$ , potom můžeme určit aktuální polohu ( $E$ ) na oběžné dráze v zadaném čase  $t$ .

```

function [E]=kepleretT(e, t, T) % funkce počítá excentrickou
                                % anomálii ze zadaných parametrů e, T
error=10e-8;                    % zvolíme chybu výpočtu
pomer=1000;                      % vstupní hodnota
M=2*pi*t/T;                     % vypočítáme střední anomálii M
E=M;                             % první iteraci položíme rovnou M

while abs(pomer)>error            % výpočet probíhá dokud pomer bude
                                % větší než zadaná chyba
pomer=(E-e*sin(E)-M)/(1-e*cos(E));% počítá zlomek dle předpisu (2.6)
E=E-pomer;                       % počítá novou hodnotu E
end
end

```



Obrázek 10: Parametry oběžné dráhy [6]

## 2.4 Výpočet parametrů oběžných drah z radiusvektoru

Další skript umožňuje výpočet elementů oběžných drah ze zadaného vektoru rychlosti  $\vec{v}$  a polohy  $\vec{r}$ . K výpočtu budou použity následující veličiny:

- uzlová přímka  $N$  - přímka, která vznikne průnikem roviny ekliptiky (rovin rovníku) a hlavní roviny oběžné dráhy,
- jednotková úhlová hybnost  $h$ ,
- úhel inklinace  $i$  - úhel sklonu roviny oběžné dráhy,
- délka vzestupného uzlu  $\Omega$  - úhel mezi uzlovou přímkou  $N$  a kladnou částí osy  $X$ ,
- excentricita  $e$ ,
- argument (šířky) pericentra  $\omega$  - úhel mezi uzlovou přímkou  $N$  a spojnici apsid,
- skutečná anomálie  $\theta$ .

### 2.4.1 Postup výpočtu

Nejprve vypočítáme velikost  $|\vec{r}|$  a  $|\vec{v}|$ , dále radiální rychlost  $v_r$  a jednotkovou hybnost. Úhel inklinace  $i$  určíme jako:

$$i = \arccos\left(\frac{h_z}{h}\right) \quad [6] \quad (2.7)$$

Úhel inklinace  $i$  musí ležet mezi  $0^\circ$  až  $180^\circ$ , aby byla oběžná dráha definována jednoznačně [6].

V dalším kroku určíme uzlovou přímku:

$$\vec{N} = \hat{K} \times \vec{h} = [0 \ 0 \ 1] \times \vec{h} \quad [6] \quad (2.8)$$

Pro výpočet délky vzestupného uzlu  $\Omega$  je třeba ještě spočítat  $|\vec{N}|$  a dále:

$$\Omega = \begin{cases} \arccos\left(\frac{N_x}{N}\right), & N_y \geq 0 \\ 360^\circ - \arccos\left(\frac{N_x}{N}\right), & N_y < 0 \end{cases} \quad [6] \quad (2.9)$$

Vektor excentricity:

$$\vec{e} = \frac{1}{\mu} \left[ \left( v^2 - \frac{\mu}{r} \right) \vec{r} - r v_r \vec{v} \right] \quad [6] \quad (2.10)$$

Velikost excentricity vypočítáme jako  $e = |\vec{e}|$ .

Skutečná anomálie  $\theta$ :

$$\theta = \begin{cases} \arccos\left(\frac{\vec{e} \cdot \vec{r}}{er}\right), & v_r \geq 0 \\ 360^\circ - \arccos\left(\frac{\vec{e} \cdot \vec{r}}{er}\right), & v_r < 0 \end{cases} \quad [6] \quad (2.11)$$

Skutečná anomálie  $\theta$  vyjde  $0^\circ \leq \theta < 180^\circ$  letí-li těleso směrem od pericentra, nebo  $180^\circ \leq \theta < 360^\circ$  letí-li těleso k pericentru [6]. Výpočet hned provedeme dle 2.11, abychom dostali jasně definovanou hodnotu, ale ve skriptu bude příkaz (`fprintf('Teleso leti smerem od pericentra')`), díky kterému o tuto informaci nepřijdeme.

Program obsahuje funkci `[r, v, Om, theta, i, a, b, p, e, h]=paramzvektoru(R, V)` a skript, který po načtení vstupních parametrů s použitím této funkce vytiskne výstup viz Obr. 11 a Obr. 12. Kompletní skript je v Příloze - Skripty

v odst. 5.2.

V tomto programu bylo nejobtížnější postihnout všechny možné varianty, které mohou pro různé oběžné dráhy nastat. Pro všechny druhy oběžných drah program spočítá veličiny uvedené v rovnicích (2.7) až (2.11). Dále pro eliptickou dráhu je za pomoci Keplerovy rovnice a upraveného skriptu pro funkci `comet` vytvořena animace pohybu po oběžné dráze. Pro ostatní kuželosečky je pomocí rovnice (1.29) vytvořen univerzální graf. Pro matici úhlů od 0 do  $2\pi$  se dopočítá matice poloměrů ze zmíněné rovnice (1.29), následně se pomocí příkazu `pol2cart` převedou souřadnice do kartézských a vykreslí se graf, který dá představu o tvaru oběžné dráhy, viz Obr. 12.

#### 2.4.2 Skript - funkce pro výpočet parametrů oběžné dráhy

```
function [r, v, Om, theta, i, a, b, p, e, h]=paramzvektoru(R, V)
global mu;           % gravitační parametr jako globální proměnná
r=norm(R);          % velikost vektoru R
v=norm(V);          % velikost vektoru V
vr=dot(R,V)/r;      % skalární součin V a R dá radiální rychlost
H=cross(R,V)        % vektor úhlové hybnosti - kolmý k R a V
h=norm(H);          % velikost vektoru úhlové hybnosti
i=acos(H(3)/h);     % vypočte úhel inklinace
N=cross([0 0 1],H); % vypočte vektor uzlové přímky - ten musí
                    % být kolmý k H a [0 0 1]
n=norm(N);          % velikost vektoru uzlové přímky
if N(2)>=0           % provede výpočet Om dle rovnice 2.11
Om=acos(N(1)/n);    % vypočítám úhel omega takto
if N(2)<0            % pokud je y-ová složka vektoru uzlové
                    % přímky záporná
omega=2*pi-omega;   % přepočítá omegu tak, aby byla mezi 0 a 180°
end
else                 % pokud je n = 0
omega=0;            % omega je také nula
end
```

```

EX=1/mu*((v^2-mu/r)*R-r*vr*V); % výpočet vektoru excentricity
e = norm(EX); % excentricita

if e~=0 % pokud je excentricita různá od nuly
    if vr>=0 % pokud je radiální rychlost větší
        % nebo rovna nule
        theta=acos(dot(EX,R)/e/r);
        % výpočet skutečné anomálie
        fprintf('Těleso letí směrem od pericentra');
    else
        theta=2*pi-acos(dot(EX,R)/e/r);
        fprintf('Těleso letí směrem k pericentru');
    end
end

else
    pom = cross(N,R); % pro případ nulové e zavedeme pomocnou
        % veličinu pro výpočet, jako vektor kolmý
        % k uzlové přímce a vektoru polohy
    if pom(3)>=0 % pokud je z-ová složka nezáporná
        theta=acos(dot(N,R)/n/r);
        % vypočítá skutečnou anomálii jako arccos
        % skalárního součinu N a R
    end
end

end

if e>1 % pokud je oběžná dráha hyperbola
    a=h^2/(mu*(e^2-1)); % výpočet a
    b=a*sqrt(e^2-1); % výpočet b
    fprintf('Oběžná dráha je hyperbola s a = %g km...
    ...a b = %g km \n\n', a, b);
    % zobrazí
elseif e<1 % pokud je oběžná dráha elipsa
    a=h^2/(mu*(1-e^2)); % výpočet a

```

```

b=a*sqrt(1-e^2); % výpočet b
fprintf('Oběžná dráha je elipsa s a = %g km...
...a b = %g km \n\n', a, b);
elseif e==0      % pokud je oběžná dráha kružnice
fprintf('Oběžná dráha je kružnice \n\n');
                % zobrazí
a=0;            % formálně načteme nulu
b=0;            % formálně načteme nulu
elseif e==1      % pokud je oběžná dráha parabola
fprintf('Oběžná dráha je parabola \n\n');
                % zobrazí
a=0;            % formálně načteme nulu
b=0;            % formálně načteme nulu
end
p=h^2/mu;       % výpočet parametru kuželosečky
end

```

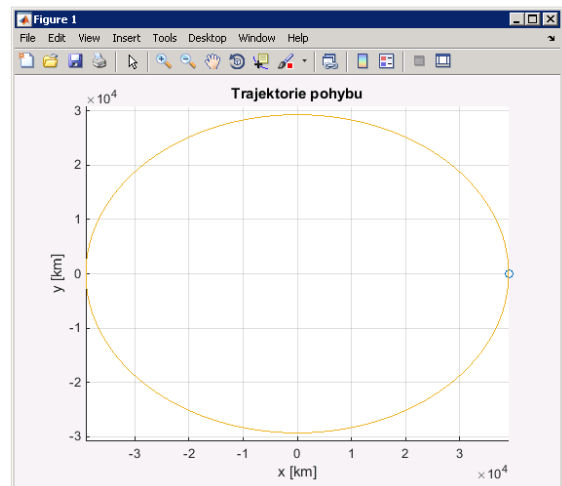
```

Zadejte vektor polohy:
rx: -7000
ry: -9500
rz: 6000
Zadejte vektor rychlosti:
vx: -4
vy: 5
vz: 3

Těleso letí směrem k pericentru
Oběžná dráha je elipsa s a = 39003.3 km a b = 29277.9 km

Velikost vektoru polohy:    13238.2 km
Velikost vektoru rychlosti: 7.07107 km/s
Inklinace:                  141.256°
Delka vzestupneho uzlu:    272.936°
Excentricita:               0.660697
Skutečna odchylka:         357.692°
>>

```



Obrázek 11: Výpočet parametrů oběžné dráhy z radiusvektoru - elipsa

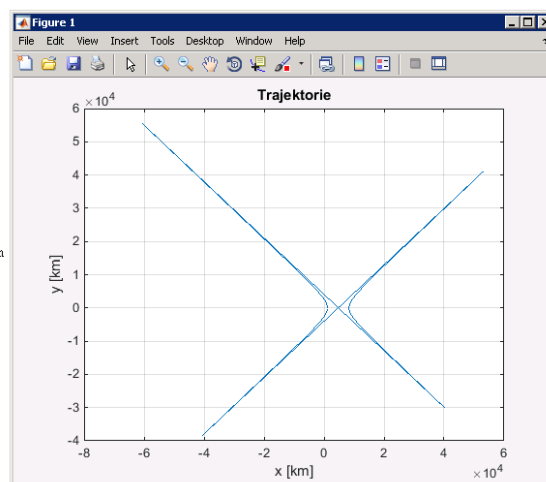
```

Zadejte vektor polohy:
rx: -9600
ry: 5000
rz: -9600
Zadejte vektor rychlosti:
vx: 9
vy: -6
vz: 7

Těleso letí směrem k pericentru
Oběžná dráha je hyperbola s a = 3594.27 km a b = 3059.63 km

Velikost vektoru polohy:      14467.9 km
Velikost vektoru rychlosti:   12.8841 km/s
Inklinace:                    66.9798°
Delka vzestupného uzlu:      310.35°
Excentricita:                 1.31325
Skutečná odchylka:           231.363°
>>

```



Obrázek 12: Výpočet parametrů oběžné dráhy z radiusvektoru - hyperbola

## 2.5 Srovnání Hohmannova a Bi-eliptického manévru

Dalším řešeným problémem jsou impulzní manévry, jmenovitě jednoduchý Hohmannův manévr, viz Obr. 13, a bi-eliptický manévr, viz Obr. 14. Skript má za úkol spočítat parametry přechodových elips, požadované změny rychlosti  $\Delta v$  pro provedení manévru a v neposlední řadě umožnit vizualizaci průběhu manévru, která bude jasně demonstrovat přeletovou dráhu.

### 2.5.1 Hohmannův manévr

Hohmannův transfer je nejjednodušší, nejběžnější pro většinu praktických situací a nejvíce energeticky úsporný transfer mezi dvěma souosými netangenciálními (nedotýkajícími se) kruhovými oběžnými drahami ležícími v jedné rovině. Hohmannův manévr typicky umožňuje minimální požadovanou změnu v rychlosti  $\Delta v$ . Přechodná elipsa je tečná k počáteční i konečné oběžné dráze. Hohmannův manévr je uskutečněn dvěma impulzními změnami rychlosti, první při přechodu z počáteční na přechodnou oběžnou dráhu a druhou při přechodu z přechodové dráhy na koncovou, viz Obr. 13. [4]

Celková změna rychlosti  $\Delta V$  potřebná k provedení Hohmannova manévru se

určí s pomocí *Vis – Viva* rovnice č. 1.31:

$$\Delta V = |V_A| + |V_B| \quad [4] \quad (2.12)$$

$$V_A = V_{p,t} - V_{p,i} = \sqrt{\mu \left( \frac{2}{r_{p,t}} - \frac{1}{a_t} \right)} - \sqrt{\mu \left( \frac{2}{r_{p,i}} - \frac{1}{a_i} \right)} \quad (2.13)$$

$$V_B = V_{a,f} - V_{a,t} = \sqrt{\mu \left( \frac{2}{r_{a,f}} - \frac{1}{a_f} \right)} - \sqrt{\mu \left( \frac{2}{r_{a,t}} - \frac{1}{a_t} \right)} \quad (2.14)$$

Index  $i$  je použit pro primární oběžnou dráhu,  $f$  pro konečnou,  $p$  pro přechodnou. Index  $a$  pro apocentrum a  $p$  pro pericentrum.

Trvání Hohmannova manévru je půlperioda přechodné oběžné dráhy, tedy dle [6]:

$$\tau = \pi \sqrt{\frac{a_p^3}{\mu}} \quad (2.15)$$

### 2.5.2 Bi-eliptický manévř

Pokud je vyžadována velká změna poloměru oběžné dráhy, není Hohmannův manévř optimální volbou. Při bi-eliptickém manévřu je přechodná elipsa zahájena ve stejném místě jako při Hohmannově, ale změna rychlosti je větší a tedy apocentrum přechodné oběžné dráhy je větší než koncové. Druhý manévř začíná v apocentru přechodné dráhy, kde změnou rychlosti vytvoříme druhou přechodnou elipsu s pericentrem shodným s koncovou oběžnou dráhou, v tomto bodě se provede poslední manévř a dosáhneme tím požadované oběžné dráhy, je to tedy trojimpulzní manévř.

Trvání bi-eliptického manévřu je součtem půlperiod přechodných elips  $p_1$  a  $p_2$  [3] [6]:

$$\tau = \pi \left( \sqrt{\frac{a_{p_1}^3}{\mu}} + \sqrt{\frac{a_{p_2}^3}{\mu}} \right) \quad (2.16)$$

Hohmannův transfer je vhodný a na  $\Delta v$  nejúspornější pro kruhové oběžné dráhy, kde poměr poloměrů počáteční a koncové oběžné dráhy je menší



než 11,94. Potom je k provedení manévru nutná minimální změna rychlosti a tím i minimální spotřeba paliva.

Bi-eliptický transfer oproti tomu je z hlediska  $\Delta v$  vhodnější pro použití pro poměr poloměrů větší než 15,58.[4]

### 2.5.3 Výpočet změn rychlostí $\Delta V$

Postup výpočtu jednotlivých a celkových  $\Delta V$  lze zjednodušeně popsat následovně:

1. Pro výpočet poloměru  $r_A$  se vyjde z rovnic (1.28) a (1.29):

$$r_A = \frac{h_1^2}{\mu} \frac{1}{1 + e_1 \cos \theta} \quad [4] \quad (2.17)$$

Pokud  $\theta$  položíme rovno nule, dostaneme vztah pro momentovou hybnost  $h_1$  počáteční oběžné dráhy.

$$h_1 = \sqrt{\mu(1 + e_1 \cos(0))}, \quad [6] \quad (2.18)$$

2. Se znalostí hybnosti potom lze dopočítat rychlost v bodě  $A$  na počáteční oběžné dráze:

$$v_{A,1} = \frac{h_1}{r_A} \quad (2.19)$$

3. Analogicky pro přechodovou elipsu 2 v místě  $A$ :

$$h_2 = \sqrt{\mu(1 + e_2 \cos(0))} \quad (2.20)$$

$$v_{A,2} = \frac{h_2}{r_A} \quad (2.21)$$

4. Změna rychlosti v místě  $A$  potřebná k provedení Hohmannova manévru  $\Delta V_A$ :

$$\Delta V_A = v_{A,2} - v_{A,1} \quad (2.22)$$

Dále analogicky pro místo  $B$  v případě Hohmannova manévru a pro místo  $B$  a  $C$  v případě bi-eliptického manévru.[6]

#### 2.5.4 Skript -funkce pro výpočet změn rychlostí pro Hohmannův manévr

```
function [dvAh, dvBh]=hohmann(rA, rB)
                                % funkce pro výpočet  $\Delta v$  pro Hohmannův manévr
rC=rB/2;
mu = 398600;                    % gravitační parametr
e2 = (rB-rA)/(rB+rA);          % výstřednost přechodové elipsy
h2 = (2*mu)^0.5*((rA*rC)/(rA+rC))^0.5;
                                % hybnost na přechodové elipse
vA2 = h2/rA;                    % rychlost na přechodové elipse v bodě A
dvAh = vA2-vA1;                % změna rychlosti v bodě A
vB2 = h2/rC;                    % rychlost na přechodové elipse v bodě B
vB3 =sqrt(mu/rC);              % rychlost na koncové kruhové oběžné dráze
dvBh = vB3-vB2;                % změna rychlosti v bodě B
end
```

#### 2.5.5 Skript -funkce pro výpočet změn rychlostí pro bi-eliptický manévr

```
function [dvA, dvB, dvC]=bielliptic(rA, rB)
rC=rB/2;
mu = 398600;                    % gravitační parametr
vA1=sqrt(mu/rA);                % rychlost na kruhové oběžné dráze 1 v bodě A

h2 = (2*mu)^0.5*((rA*rB)/(rA+rB))^0.5;
                                % hybnost na přechodové elipse 1
vA2 = h2/rA;                    % rychlost na přechodové elipse 1 v bodě A
vB2 = h2/rB;                    % rychlost na přechodové elipse 1 v bodě B
```

```

h3 = (2*mu)^0.5*((rC*rB)/(rC+rB))^0.5;
                                % hybnost na přechodové elipse 2
vB3 = h3/rB;                    % rychlost na přechodové elipse 2 v bodě A
vC3 = h3/rC;                    % rychlost na přechodové elipse 2 v bodě C

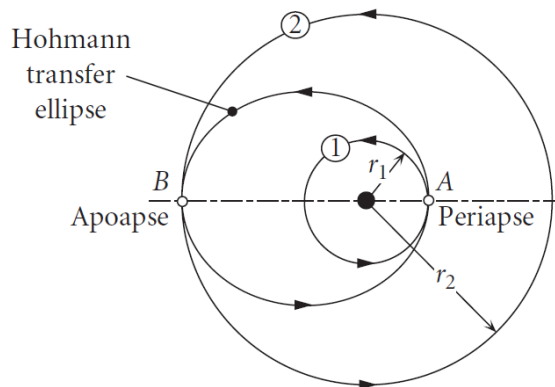
vC4=(mu/rC)^0.5;                % rychlost na kruhové oběžné dráze 2
                                % v bodě C
vD4=vC4;                        % rychlost na kruhové oběžné dráze 2
                                % v bodě D

dvA=vA2-vA1;                    % celková změna rychlosti v bodě A
dvB=vB3-vB2;                    % celková změna rychlosti v bodě B
dvC=vC4-vC3;                    % celková změna rychlosti v bodě C
end

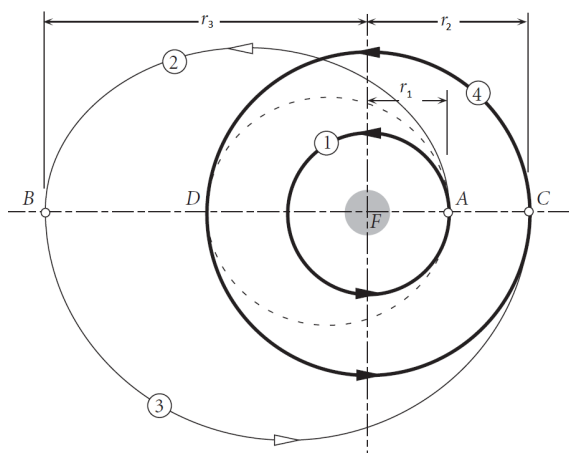
```

### 2.5.6 Výpočet přechodových elips

Výpočet přechodových elips, tedy rovnice (2.23) až (2.28), vychází z obrázků Obr. 13 a Obr. 14.



Obrázek 13: Hohmannův manévr [6]



Obrázek 14: Bi-eliptický manévr. [6]

Výpočet se provádí ze zadaných parametrů, tedy poloměrů počáteční a koncové oběžné dráhy (v tomto zjednodušeném případě jsou to kružnice). Pro Hohmannův manévr tedy platí:

$$a_p = \frac{r_1 + r_2}{2} \quad (2.23)$$

$$e_p = \frac{r_2 - r_1}{r_2 + r_1}, \quad (2.24)$$

kde  $r_1$  je radius počáteční oběžné dráhy,  $r_2$  je radius konečné oběžné dráhy,  $a_p$  je hlavní poloosa přechodné elipsy a  $e_p$  je excentricita přechodové elipsy. Pro výpočet bi-eliptického manévru ještě zavedeme pomocný poloměr  $r_T$ . Velikost tohoto poloměru se volí dle počítaného případu tak, aby byly přechodové elipsy v co nejširším měřítku volitelných parametrů (v našem případě poměru poloměrů vstupní a výstupní oběžné dráhy) tečné k oběžným drahám. Po několika simulacích jsem navrhla velikost  $r_T = 3r_A$  pro  $r_A > r_B$  a  $r_T = 3r_B$  pro  $r_B > r_A$ .

$$a_{p,1} = \frac{r_A + r_T}{2} \quad (2.25)$$

$$e_{p,1} = a_{p,1} - r_A \quad (2.26)$$

$$a_{p,2} = \frac{r_B + r_T}{2} \quad (2.27)$$

$$e_{p,2} = a_{p,2} - r_B, \quad (2.28)$$

kde  $a_{p,1}$  je hlavní poloosa první přechodné elipsy,  $e_{p,1}$  je excentricita první

přechodné elipsy,  $a_{p,2}$  a  $e_{p,2}$  je hlavní poloosa a excentricita druhé přechodné elipsy.

### 2.5.7 Skript - trajektorie Hohmannova manévru

```
function [x1, y1]=hohmanntjectory(rA, rB)
    % funkce, která vytvoří data pro animaci
    fi11=(-2*pi):0.1:0; % vytvoří matici úhlů po kroku 0,1 rad
    x11=rA*cos(fi11); % data x pro graf a animaci kružnice
    y11=rA*sin(fi11); % data y pro graf a animaci kružnice

    fi21=0:0.1:pi; % vytvoří matici úhlů po kroku 0,1 rad
    a21=(rB+rA)/2; % hlavní poloosa přechodové elipsy 1
    e21 = (rB-rA)/(rB+rA); % excentricita přechodové elipsy 1
    b21=sqrt(a21^2-e21^2); % vedlejší poloosa přechodové elipsy 1
    x21=a21*cos(fi21)-(a21-rA);
    % data x pro graf a animaci přechodové elipsy
    y21=b21*sin(fi21); % data y pro graf a animaci
    % přechodové elipsy
    fi31=(pi):0.1:(3*pi); % vytvoří matici úhlů po kroku 0,1 rad
    x31=rB*cos(fi31); % data x pro graf a animaci kružnice
    y31=rB*sin(fi31); % data y pro graf a animaci kružnice

    x1=[x11 x21 x31]; % spoji data x do jediné matice
    y1=[y11 y21 y31]; % spoji data y do jediné matice
end
```

### 2.5.8 Skript - trajektorie bi-eliptického manévru

```
function [x2, y2]=bielliptictrajectoryupr(rA, rB)
    % stanovíme velikost pomocného poloměru rT
    if rA>rB
        rT = rA*3; % zvolíme
```

```

else
    rT = rB*3;          % zvolíme
end
fi12 = -2*pi:0.1:0;    % vytvoří matici úhlů s krokem 0,1 rad
x12=rA*cos(fi12);     % data x pro graf a animaci kružnice
y12=rA*sin(fi12);     % data y pro graf a animaci kružnice

fi22= 0:0.1:pi;       % vytvoří matici úhlů s krokem 0,1 rad
atr1 = (rA+rT)/2;     % hlavní poloosa přechodové elipsy 1
etr1 = atr1-rA;       % excentricita přechodové elipsy 1
btr1 = sqrt(atr1^2-etr1^2);
                        % vedlejší poloosa přechodové elipsy 1
x22=((atr1))*cos(fi22)-atr1+(rA);
                        % data x pro graf a animaci
y22=btr1*sin(fi22);   % data y pro graf a animaci

fi32=pi:0.1:2*pi;     % vytvoří matici úhlů s krokem 0,1 rad
atr2=(rB+rT)/2;       % hlavní poloosa přechodové elipsy 2
etr2=atr2-rB;         % excentricita přechodové elipsy 2
btr2=sqrt(atr2^2-(etr1)^2);
                        % vedlejší poloosa přechodové elipsy 2
x32=atr2*cos(fi32)-atr2+rB;
                        % data x pro graf a animaci
y32=btr2*sin(fi32);   % data y pro graf a animaci

fi42=2*pi:0.1:4*pi;   % vytvoří matici úhlů s krokem 0,1 rad
x42=rB*cos(fi42);     % data x pro graf a animaci
y42=rB*sin(fi42);     % data y pro graf a animaci

x2=[x12 x22 x32 x42]; % spoji všechny data x do jedné matice
y2=[y12 y22 y32 y42]; % spoji všechny data y do jedné matice
end

```

### 2.5.9 Skript - Ciolkovského rovnice a demonstrace typu pohonu

Jako demonstraci pro použití různých typů pohonu jsem zvolila jednoduchý výpočet na založený na Ciolkovského rovnici (1.35). Skript jsem spojila s předchozím výpočtem  $\Delta v$  ze skriptu pro výpočet impulzních manévřů. Na základě volby  $I_{sp}$  a výpočtu  $\Delta v$  pro Hohmannův a bi-eliptický manévr se spočítá konstrukční číslo  $C = \frac{m_0}{m}$ .

```
                                % typ pohonu
disp('Zvolte možnost: '); % ukáže text
TP = input('1 - Zvolte typ pohonu rakety, 2 - Zadejte Isp \n');
                                % ukáže text a načte hodnotu TP
if TP == 1                       % pokud zvolím možnost 1
    pohon = input('11 - TPL, 22 - KPL, 33 - jaderný \n');
                                % nabídne možnosti pohonů a načte hodnotu
                                % proměnné pohon
    if pohon == 11               % pokud zvolím TPL
        Isp = 2500;             % načte příslušnou hodnotu Isp
    elseif pohon == 22
        Isp = 4000;
    elseif pohon == 33
        Isp = 8000;
    else
        disp('Zadali jste neplatnou hodnotu');
    end
elseif TP == 2                   % pokud zvolíme možnost 2
    Isp = input('Zadejte hodnotu Isp: ');
                                % načte hodnotu Isp
else
    disp('Zadali jste neplatnou hodnotu');
end

cH=exp(deltavH/Isp);           % výpočet koef. C z Tsiolkovského rovnice
```

```

cB=exp(deltavB/Isp);      % výpočet koef. C z Tsiolkovského rovnice
                           % výstupy
fprintf('Poměr m0/m pro Hohmannův manévr: %g \n', cH);
fprintf('Poměr m0/m pro bi-eliptický manévr: %g \n', cB);

```

```

Hohmannův vs Bi-eliptický transfér
Uvažujme dvě kruhové soustředné oběžné dráhy ležící ve stejné rovině
Zadejte poloměr počáteční oběžné dráhy [km]: 500
Zadejte poloměr koncové oběžné dráhy [km]: 1600

```

```

Hohmannův manévr
delta-vA = 3.08889 km/s
delta-vB = 2.74425 km/s
celkové delta-v = 5.83314 km/s

```

```

Bi-Elliptic manévr
delta-vA = 6.61898 km/s
delta-vB = 1.99555 km/s
delta-vC = -3.45315 km/s
celkové delta-v = 12.0677 km/s

```

```

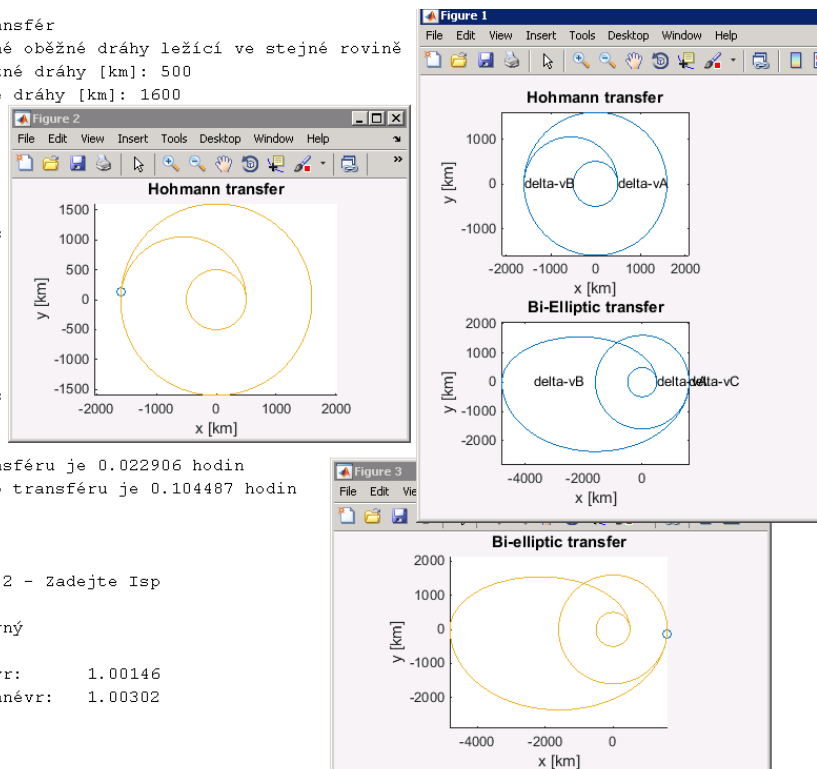
Trvání přeletu Hohmannova transféru je 0.022906 hodin
Trvání přeletu Bi-Elliptického transféru je 0.104487 hodin

```

```

Zvolte možnost:
1 - Zvolte typ pohonu rakety, 2 - Zadejte Isp
1
11 - TPL, 22 - KPL, 33 - jaderný
22
Poměr m0/m pro Hohmannův manévr:      1.00146
Poměr m0/m pro bi-eliptický manévr:    1.00302
>>

```



Obrázek 15: Výstupy - Hohmannův vs bi-eliptický manévr

Skript pro výpočet pohonu lze samozřejmě jednoduše upravit v závislosti na tom, jaké hodnoty chceme jako vstupní a které hodnoty jako výstupní. Konstrukční číslo  $C$  jsem jako výstup zvolila, protože umožňuje při použití stejného pohonu demonstrovat spotřebu paliva pro jednotlivé manévry a kromě demonstrace pohonu tedy jasně ukazuje na manévr, který je pro konkrétní příklad vhodnější.

Původně jsem zamýšlela vykreslovat elipsy a trajektorie pomocí Keplerovy rovnice, jako v předchozím případě, ale nakonec jsem se rozhodla pro jednoduché vykreslování založené pouze na funkci `comet`, které je pro tento případ



daleko přehlednější. Hlavním problémem je ale to, že při výpočtu dat pro kruhovou trajektorii probíhá výpočet daleko rychleji než pro eliptickou a tím pádem se každá část (počáteční oběžná dráha, přechodná elipsa, ...) vykresluje jinou rychlostí a tudíž neslouží účelu - vykreslovat v měřítku skutečné rychlosti.

Celý skript je součástí části Příloha - Skripty, odst. 5.3. Výstup z programu je znázorněn na Obr. 15.

### 3 Závěr

Bakalářská práce měla za úkol vytvořit 2D simulátor manévrování kosmických těles. Podařilo se mi vytvořit soubor skriptů v MATLABu, které simulují základní problémy orbitální mechaniky a manévrování. Simulátor je určen jako doplněk výuky orbitální mechaniky, proto jsem programy koncipovala tak, aby bylo možné demonstrovat způsob výpočtu doplněný základními grafickými výstupy.

Simulátor obsahuje skript pro výpočet základních parametrů oběžné dráhy ve 2D na základě znalosti pericentra a excentricity počítá vlastnosti elipsy a rychlosti v apo- a pericentru, dále obsahuje skript pro výpočet Keplerovy rovnice a její praktické využití pro animaci pohybu na oběžné dráze. Skript pro výpočet parametrů oběžné dráhy z radiusvektoru, který se zabývá polohou a vlastnostmi oběžné dráhy v třírozměrném prostoru. Jako příklad manévrování jsem vytvořila skript zabývající se Hohmannovým a bi-eliptickým manévrem, který demonstruje rozdíl mezi těmito manévry, dále počítá potřebné změny rychlostí, trajektorie pohybu a v neposlední řadě umožňuje výběr typu kosmického pohonu a na základě Ciolkovského rovnice demonstruje spotřebu paliva pro jednotlivé manévry a vybrané pohony.

Hlavní výhody vytvořeného simulátoru spatřuji v jednoduchosti, všechny stěžejní výpočty jsou napsány ve formě funkce (m-file), které lze libovolně kombinovat a vytvořit řešení pro zcela nové příklady.

Jako hlavní nedostatek a zároveň prostor pro zlepšení do budoucna vidím v tom, že simulátor není online, bylo by tedy vhodné simulátor přepsat do podoby webové nebo mobilní aplikace, což je spojeno s lepší grafickou stránkou, která by simulátoru přidala na atraktivitě.

## 4 Seznam obrázků

Obr. 1 - 1. Keplerův zákon, převzato z [3]

Obr. 2 - 2. Keplerův zákon, převzato z [3]

Obr. 3 - Působení hmotných bodů, převzato z [3]

Obr. 4 - Trajektorie  $m_2$  kolem  $m_1$  v rovině kolmé na na vektor  $\vec{h}$ , převzato z [6]

Obr. 5 - Diferenciální plocha  $dA$  dána relativní polohou vektoru polohy  $\vec{r}$  za časový úsek  $dt$ , převzato z [6]

Obr. 6 - Kuželosečky, převzato z [4]

Obr. 7 - Polární souřadnice, převzato z [3]

Obr. 9 - Výpočet parametrů oběžné dráhy pro  $zp = 9000 \text{ km}$  a  $e = 0,5$  kolem Jupitera

Obr. 8 - Schéma eliptické oběžné dráhy a jí opsané kruhové dráhy, převzato z [3]

Obr. 10 - Parametry oběžné dráhy, převzato z [6]

Obr. 11 - Výpočet parametrů oběžné dráhy z radiusvektoru - elipsa

Obr. 12 - Výpočet parametrů Oběžné dráhy z radiusvektoru - hyperbola

Obr. 13 - Hohmannův manévr, převzato z [6]

Obr. 14 - Bi-eliptický manévr, převzato z [6]

Obr. 15 Výstupy - Hohmannův vs bi-eliptický manévr

## Reference

- [1] Johannes Kepler. *Astronomia Nova*. 1609.
- [2] Johannes Kepler. *Harmonices Mundi*. 1619.
- [3] Alessandro de Iaco Veris. *Practical Astrodynamics*. Springer, 2016. 978-3-319-62220-0.
- [4] Viorel Badescu Malcolm Macdonald. *The International Handbook of Space Technology*. Springer, 2014. 978-3-642-41101-4.
- [5] Dušan Majer. Malý průvodce po oběžných drahách. <http://www.kosmonautix.cz/2013/02/maly-pruvodce-po-obeznych-drahach/>, 2018 (zveřejněno 11.02.2013).
- [6] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. Elsevier, 2005. 0-7506-6169-0.
- [7] Lála P Vítek, A. Malá encyklopedie kosmonautiky - pohonné systémy. <http://mek.kosmo.cz/zaklady/rakety/motory.htm>, 2018 (zveřejněno 24.07.2007).
- [8] Michal Václavík. Kosmonautika xvi - raketové motory. <http://www.hvezdarna-vsetin.cz/rservice.php?akce=tisk&cislocclanku=2004070001>, 2018 (zveřejněno 12. 07. 2004).
- [9] Inc. The MathWorks. Product documentation - matlab. <https://uk.mathworks.com/help/matlab/index.html>, 2018.
- [10] Nikos Drakos. Newtonova metoda. [https://math.fce.vutbr.cz/vyuka/matematika/numericke\\_metody/node10.html](https://math.fce.vutbr.cz/vyuka/matematika/numericke_metody/node10.html), 2018.

## 5 Příloha - Skripty

### 5.1 Výpočet základních parametrů oběžné dráhy

Funkce pro základních parametrů oběžné dráhy

```
function [h, vp, va, T, a, b]=zpegiven(zp, e, polo)
rp=polo+zp; % radius v pericentru
h=(rp*mu*(1+e))^0.5; % hybnost
vp=h/rp; % rychlost v pericentru
ra=h^2/(mu*(1-e)); % radius v apocentru
a=(rp+ra)/2; % velikost hlavní poloosy
va=h/ra; % rychlost v apocentru
T=(2*pi/mu^2)*(h/sqrt(1-e^2))^3; % perioda oběhu
b=sqrt(a^2-(e*a)^2); % vedlejší poloosa
end
```

Skript - výpočet základních parametrů oběžné dráhy

```
clc
clear all
close all
global mu % globální proměnná-gravitační parametr

fprintf('Tento skript pocita parametry obezne elipticke drahy ze
        zadane vzdálenosti pericentra a excentricity \n');

fprintf('Zvolete kolem ktereho telesa se pohybujeme.\n')
% načte vstupní parametry
fprintf('1 - Merkur, 2 - Venuse, 3 - Zeme, 4 - Mars, 5 - Jupiter,
        6 - Saturn, 7 - Uran, 8 - Neptun, 9 - Pluto, 10 - Slunce \n');
LALA = input('Zvolte cislo: \n'); % načte číslo do proměnné LALA
```

```

% přiřadí hodnotu mu

if LALA ==1
    mu=22032;
    polo = 2440;
elseif LALA == 2
    mu=324859;
    polo=6052;
elseif LALA == 3
    mu=389600;
    polo=6378;
elseif LALA == 4
    mu=42828;
    polo=3396;
elseif LALA == 5
    mu=126686534;
    polo=69911;
elseif LALA == 6
    mu=37931187;
    polo=58232;
elseif LALA == 7
    mu=5793947;
    polo=2536;
elseif LALA == 8
    mu=6836529;
    polo=24622;
elseif LALA == 9
    mu=1001;
    polo=2370;
elseif LALA == 10
    mu=132712440000;
    polo=1392020;
else

```

```

        fprintf('Zadali jste spatnou hodnotu, stisknete CTRL+C'
        a zacnete znovu');
end

zp=input('Zadejte delku percentra zp: ');
        % načte zp
e=input('Zadejte vystrednost elipticke drahy e: ');
        % načte e
if e<1 & e>0        % pokud je oběžná dráha elipsa
[h,vp,va,T,a,b]=zpegiven(zp,e, polo);
        % volá funkci a počítá parametry
alfa=0:0.1:2*pi;        % vytvoří matici úhlů [0, 0,1, 0,2....2*pi]
plot(a*cos(alfa), b*sin(alfa));% graf elipsy
hold on
grid on        % zapne mřížku
axis equal        % osy 1:1
title('Trajektorie');        % název grafu
xlabel('x [km]');ylabel('y [km]');        % popis os
plot(polo*cos(alfa)-a*e, polo*sin(alfa));
        % vykreslí planetu
legend('Trajektorie', 'Planeta');        % legenda
hold off

% výstupy
fprintf('\n Hybnost: h = %g', h);
fprintf('\n Rychlost v pericentru: vp = %g', vp);
fprintf('\n Rychlost v apocentru: va = %g', va);
fprintf('\n Perioda obehu: T = %g', T);
else        % pokud oběžná dráha není elipsa
printf('Zadana obezna draha neni elipticka');        % píše
end

```

## 5.2 Výpočet parametrů oběžné dráhy z radiusvektoru

### Funkce pro výpočet parametrů oběžné dráhy z radiusvektoru

```
function [r, v, Om, theta, i, a, b, p, e, h]=paramzvektoru(R, V)
global mu; % gravitační parametr jako globální proměnná
r=norm(R); % velikost vektoru R
v=norm(V); % velikost vektoru V
vr=dot(R,V)/r; % skalární součin V a R dá radiální rychlost
H=cross(R,V) % vektor úhlové hybnosti - kolmý k R a V
h=norm(H); % velikost vektoru úhlové hybnosti
i=acos(H(3)/h); % vypočte úhel inklinace
N=cross([0 0 1],H); % vypočte vektor uzlové přímky - ten musí
% být kolmý k H a [0 0 1]
n=norm(N); % velikost vektoru uzlové přímky
if N(2)>=0 % provede výpočet Om dle rovnice 2.11
Om=acos(N(1)/n); % vypočítám úhel omega takto
if N(2)<0 % pokud je y-ová složka vektoru uzlové
% přímky záporná
omega=2*pi-omega; % přepočítá omegu tak, aby byla mezi 0 a 180°
end
else % pokud je n = 0
omega=0; % omega je také nula
end

EX=1/mu*((v^2-mu/r)*R-r*vr*V); % výpočet vektoru excentricity
e = norm(EX); % excentricita

if e~=0 % pokud je excentricita různá od nuly
if vr>=0 % pokud je rad. rychlost větší nebo rovna nule
theta=acos(dot(EX,R)/e/r);
% výpočet skutečné anomálie
fprintf('Těleso letí směrem od pericentra');
```



```

else
    theta=2*pi-acos(dot(EX,R)/e/r);
    fprintf('Těleso letí směrem k pericentru');
end
else
pom = cross(N,R);% pro případ nulové e zavedeme pomocnou
                % veličinu pro výpočet, jako vektor kolmý
                % k uzlové přímce a vektoru polohy
if pom(3)>=0 % pokud je z-ová složka nezáporná
theta=acos(dot(N,R)/n/r);
                % vypočítá skutečnou anomálii jako arccos
                % skalárního součinu N a R

end
end

if e>1          % pokud je oběžná dráha hyperbola
a=h^2/(mu*(e^2-1));% výpočet a
b=a*sqrt(e^2-1); % výpočet b
fprintf('Oběžná dráha je hyperbola s a = %g km...
...a b = %g km \n\n', a, b);
                % zobrazí
elseif e<1     % pokud je oběžná dráha elipsa
a=h^2/(mu*(1-e^2));% výpočet a
b=a*sqrt(1-e^2); % výpočet b
fprintf('Oběžná dráha je elipsa s a = %g km...
...a b = %g km \n\n', a, b);
elseif e==0    % pokud je oběžná dráha kružnice
fprintf('Oběžná dráha je kružnice \n\n');
                % zobrazí
a=0;           % formálně načteme nulu
b=0;           % formálně načteme nulu
elseif e==1    % pokud je oběžná dráha parabola
fprintf('Oběžná dráha je parabola \n\n');

```

```

                                % zobrazí
    a=0;                          % formálně načteme nulu
    b=0;                          % formálně načteme nulu
end
p=h^2/mu;                        % výpočet parametru kuželosečky
end

```

### Funkce pro výpočet Keplerovy rovnice

```

function [E]=kepleretT(e, t, T)
                                % funkce pro výpočet Keplerovy rovnice
error=10e-8;                    % zvolíme chybu výpočtu
pomer=1000;                      % první hodnota proměnné pomer, která
                                % reprezentuje poměr  $f(E_n)/f'(E_n)$ 
M=2*pi*t/T;                      % vypočítáme střední anomálii M
E=M/2;                          % první iteraci položíme rovnou střední
                                % anomálii M
while abs(pomer)>error % výpočet E pomocí Newtonovy metody
    pomer = (E-e*sin(E)-M)/(1-e*cos(E));
                                % počítáme poměr dle předpisu
    E=E-pomer;                  % od předchozí proměnné odečítáme pomer,
                                % dokud absolutní hodnota pomer nebude menší
                                % než zvolená přesnost
end
end

```

### Skript - výpočet parametrů oběžné dráhy z radiusvektoru

```

clc
clear all
close all

```

```

global mu;
mu=398600;
fprintf('Zadejte vektor polohy: \n');
R(1)=input('rx: ');    % načte složku x vektoru polohy
R(2)=input('ry: ');    % načte složku y vektoru polohy
R(3)=input('rz: ');    % načte složku z vektoru polohy
fprintf('Zadejte vektor rychlosti: \n');
V(1)=input('vx: ');    % načte složku x vektoru rychlosti
V(2)=input('vy: ');    % načte složku y vektoru rychlosti
V(3)=input('vz: ');    % načte složku z vektoru rychlosti

[r, v, Om, theta, i, a, b, p, e, h]=paramzvektoru(R, V);
                                % volá funkci pro výpočet parametrů oběžné dráhy

if e>0 & e<1                    % pokud se jedná o elipsu
    X=0;                          % počáteční hodnota
    T=(2*pi*a^1.5)/mu^0.5;        % výpočet periody
    for tau=0:T/1000:T;          % pro tau = [0, T/1000, ..., T]
        E=kepleretT(e,tau,T);    % spočítá E v zadaných hodnotách dilu periody
        X=[X;E];                 % vytvoří matici úhlů pro jednotlivé díly periody
    end

    x=a*cos(X);y=b*sin(X) % výpočet dat pro vykreslení elipsy
    figure();                  % otevře nové okno
    comett(x,y);              % vykreslení elipsy pomocí upravené funkce comet
    axis equal;                % osy 1:1
    xlabel('x [km]');ylabel('y [km]'); % popis os grafu
    title('Trajektorie pohybu'); % název grafu
else
    fi=0:0.1:2*pi;            % vytvoří matici úhlů
    er=h^2./(mu*(1.+e.*cos(fi))); % rovnice oběžné dráhy (1.29)
    [x,y]=pol2cart(fi, er);    % převod polárních
                                % souřadnic do kartézských řadnic

```

```

plot(x,y);          % graf
grid on;           % zapne mřížku
xlabel('x [km]'); ylabel('y [km]') % popis os
title('Trajektorie'); % název grafu
end

                % výstupy
fprintf('Velikost vektoru polohy: %g km\n', r);
fprintf('Velikost vektoru rychlosti: %g km/s\n', v);
fprintf('Inklinace: %g° \n', i*180/pi);
fprintf('Delka vyestupneho uzlu: %g° \n', omega*180/pi);
fprintf('Excentricita: %g \n', e);
fprintf('Skutecna odchylka: %g° \n', TA*180/pi);

```

### 5.3 Hohmannův versus bi-eliptický manévr

#### Funkce pro výpočet změn rychlostí pro Hohmannův manévr

```
function [dvAh, dvBh]=hohmann(rA, rB)
                                % funkce pro výpočet  $\Delta v$  pro Hohmannův manévr
rC=rB/2;
mu = 398600;                    % gravitační parametr
e2 = (rB-rA)/(rB+rA);          % výstřednost přechodové elipsy
h2 = (2*mu)^0.5*((rA*rC)/(rA+rC))^0.5;
                                % hybnost na přechodové elipse
vA2 = h2/rA;                    % rychlost na přechodové elipse v bodě A
dvAh = vA2-vA1;                % změna rychlosti v bodě A
vB2 = h2/rC;                    % rychlost na přechodové elipse v bodě B
vB3 =sqrt(mu/rC);              % rychlost na koncové kruhové oběžné dráze
dvBh = vB3-vB2;                % změna rychlosti v bodě B
end
```

#### Funkce pro výpočet změn rychlostí pro bi-eliptický manévr

```
function [dvA, dvB, dvC]=bielliptic(rA, rB)
rC=rB/2;
mu = 398600;                    % gravitační parametr
vA1=sqrt(mu/rA);                % rychlost na kruhové oběžné dráze 1 v bodě A

h2 = (2*mu)^0.5*((rA*rB)/(rA+rB))^0.5;
                                % hybnost na přechodové elipse 1
vA2 = h2/rA;                    % rychlost na přechodové elipse 1 v bodě A
vB2 = h2/rB;                    % rychlost na přechodové elipse 1 v bodě B

h3 = (2*mu)^0.5*((rC*rB)/(rC+rB))^0.5;
                                % hybnost na přechodové elipse 2
vB3 = h3/rB;                    % rychlost na přechodové elipse 2 v bodě A
```

```

vC3 = h3/rC;           % rychlost na přechodové elipse 2 v bodě C

vC4=(mu/rC)^0.5;      % rychlost na kruhové oběžné dráze 2
                      % v bodě C
vD4=vC4;              % rychlost na kruhové oběžné dráze 2
                      % v bodě D

dvA=vA2-vA1;          % celková změna rychlosti v bodě A
dvB=vB3-vB2;          % celková změna rychlosti v bodě B
dvC=vC4-vC3;          % celková změna rychlosti v bodě C
end

```

### Funkce pro výpočet trajektorie pro Hohmannův manévr

```

function [x1, y1]=hohmanntjectory(rA, rB)
                      % funkce, která vytvoří data pro animaci
fi11=(-2*pi):0.1:0;  % vytvoří matici úhlů po kroku 0,1 rad
x11=rA*cos(fi11);    % data x pro graf a animaci kružnice
y11=rA*sin(fi11);    % data y pro graf a animaci kružnice

fi21=0:0.1:pi;       % vytvoří matici úhlů po kroku 0,1 rad
a21=(rB+rA)/2;        % hlavní poloosa přechodové elipsy 1
e21 = (rB-rA)/(rB+rA); % excentricita přechodové elipsy 1
b21=sqrt(a21^2-e21^2); % vedlejší poloosa přechodové elipsy 1
x21=a21*cos(fi21)-(a21-rA);
                      % data x pro graf a animaci přechodové elipsy
y21=b21*sin(fi21);    % data y pro graf a animaci
                      % přechodové elipsy
fi31=(pi):0.1:(3*pi); % vytvoří matici úhlů po kroku 0,1 rad
x31=rB*cos(fi31);    % data x pro graf a animaci kružnice
y31=rB*sin(fi31);    % data y pro graf a animaci kružnice

x1=[x11 x21 x31];    % spoji data x do jediné matice

```

```

y1=[y11 y21 y31];      % spoji data y do jediné matice
end

```

### Funkce pro výpočet trajektorie pro bi-eliptický manévr

```

function [x2, y2]=bielliptictrajectoryupr(rA, rB)
                                % stanovíme velikost pomocného poloměru rT
if rA>rB
    rT = rA*3;                  % zvolíme
else
    rT = rB*3;                  % zvolíme
end
fi12 = -2*pi:0.1:0;           % vytvoří matici úhlů s krokem 0,1 rad
x12=rA*cos(fi12);              % data x pro graf a animaci kružnice
y12=rA*sin(fi12);              % data y pro graf a animaci kružnice

fi22= 0:0.1:pi;                % vytvoří matici úhlů s krokem 0,1 rad
atr1 = (rA+rT)/2;              % hlavní poloosa přechodové elipsy 1
etr1 = atr1-rA;                % excentricita přechodové elipsy 1
btr1 = sqrt(atr1^2-etr1^2);
                                % vedlejší poloosa přechodové elipsy 1
x22=((atr1))*cos(fi22)-atr1+(rA);
                                % data x pro graf a animaci
y22=btr1*sin(fi22);           % data y pro graf a animaci

fi32=pi:0.1:2*pi;              % vytvoří matici úhlů s krokem 0,1 rad
atr2=(rB+rT)/2;                % hlavní poloosa přechodové elipsy 2
etr2=atr2-rB;                  % excentricita přechodové elipsy 2
btr2=sqrt(atr2^2-(etr1)^2);
                                % vedlejší poloosa přechodové elipsy 2
x32=atr2*cos(fi32)-atr2+rB;
                                % data x pro graf a animaci
y32=btr2*sin(fi32);           % data y pro graf a animaci

```

```

fi42=2*pi:0.1:4*pi;      % vytvoří matici úhlů s krokem 0,1 rad
x42=rB*cos(fi42);      % data x pro graf a animaci
y42=rB*sin(fi42);      % data y pro graf a animaci

x2=[x12 x22 x32 x42];  % spoji všechny data x do jedné matice
y2=[y12 y22 y32 y42];  % spoji všechny data y do jedné matice
end

```

### Skript - Hohmannův versus bi-eliptický manévr

```

clear all
close all
clc

disp('Hohmann vs Bi-elliptic transfér');
fprintf('Uvažujme dvě kruhové soustředné oběžné dráhy
ležící ve stejné rovině se \n');
rA=input('Zadejte poloměr počáteční oběžné dráhy: '); % načte rA
rB=input('Zadejte poloměr koncové oběžné dráhy: '); % načte rB
rC=rB/2;          % výpočet rC dle obrázku

[x1, y1]=hohmantrajectory(rA, rB);
                    % volá funkci pro výpočet Hohmannovy trajektorie
[x2, y2]=bielliptictrajectoryupr(rA, rB);
                    % volá funkci pro výpočet bieliptické trajektorie

figure(1);          % otevře nové okno
subplot(2,2,1);     % pozice
plot(x1, y1);       % vytvoří schéma Hohmannova manévru
axis equal          % osy 1:1
title('Hohmannuv transfer'); % název grafu
xlabel('x [km]'); ylabel('y [km]'); % popis os

```



```

hold on
text(rA, 0, 'delta-vA'); % popisek
text(-rB, 0, 'delta-vB'); % popisek
hold off
subplot(2,2,3); % umístění 2. schématu
plot(x2, y2); %vytvoří schéma bieliptického manévru
axis equal % osy 1:1
title('Bi-Ellipticky transfer'); % název grafu
xlabel('x [km]'); ylabel('y [km]'); % popis os
hold on
text(rA, 0, 'delta-vA'); % komentář v grafu
text(rB, 0, 'delta-vC'); % komentář v grafu
xvBB=-2*(2*rB+rA+rB)/2+rB; yvBB=0;str5='delta-vB';
text(xvBB, yvBB, str5); % komentář v grafu
hold off
figure(2); % nové okno
comet(x1, y1); % animace kružnice Hohmannova manévru
axis equal % osy 1:1
title('Hohmann transfer'); % název grafu
xlabel('x [km]'); ylabel('y [km]'); % popis os

figure(3); % nové okno
comet(x2, y2); % animace bieliptického manévru
axis equal % osy 1:1
title('Bi-Ellipticky transfer'); % název grafu
xlabel('x [km]'); ylabel('y [km]'); % popis os

[dvA, dvB, dvC]=bielliptic(rA, rB);
% volá funkci pro výpočet  $\Delta V$  pro
% bieliptický manévr
deltavB=abs(dvA)+abs(dvB)+abs(dvC);
% výpočet celkové změny rychlosti
[dvAh, dvBh]=hohmann(rA, rB); % volá funkci pro výpočet  $\Delta V$ 

```

```

                                % pro Hohmannův manévr
deltavH=dvAh+dvBh;              % výpočet celkové změny rychlosti

                                % výstupy
fprintf('\n\nHohmannův manévr \n');
fprintf('delta-vA = %g km/s \n', dvAh );
fprintf('delta-vB = %g km/s \n', dvBh );
fprintf('celkové delta-v = %g km/s \n\n\n', deltavH);

fprintf('Bi-Elliptic manévr \n');
fprintf('delta-vA = %g km/s \n', dvA);
fprintf('delta-vB = %g km/s \n', dvB);
fprintf('delta-vC = %g km/s \n', dvC);
fprintf('celkové delta-v = %g km/s \n\n\n', deltavB);

mu=398600;                       % gravitační parametr
tH=0.5*((2*pi*(0.5*(rA+rC))^(3/2))/mu^0.5)/(60*60);
                                % výpočet doby trvání Hohmannova manévru
fprintf('Trvání přeletu Hohmannova transféru je %g hodin\n', tH);
tB=0.5*((2*pi*(0.5*(rA+rB))^(3/2))/mu^0.5+((2*pi*(0.5*(rB+rC))^(1.5)...
.../mu^0.5))/(60*60);           % výpočet doby trvání bi-eliptického manévru
fprintf('Trvání přeletu Bi-Elliptického transféru je %g hodin\n', tB);

                                % typ pohonu
disp('Zvolte možnost: ');% ukáže text
TP = input('1 - Zvolte typ pohonu rakety, 2 - Zadejte Isp \n');
                                % ukáže text a načte hodnotu TP
if TP == 1                       % pokud zvolím možnost 1
    pohon = input('11 - TPL, 22 - KPL, 33 - jaderný \n');
                                % nabídne možnosti pohonů a načte hodnotu
                                % proměnné pohon
    if pohon == 11               % pokud zvolím TPL

```

```

        Isp = 2500;          % načte příslušnou hodnotu Isp
elseif pohon == 22
        Isp = 4000;
elseif pohon ==33
        Isp = 8000;
else
        disp('Zadali jste neplatnou hodnotu');
end
elseif TP == 2          % pokud zvolíme možnost 2
        Isp = input('Zadejte hodnotu Isp: ');
                        % načte hodnotu Isp
else
        disp('Zadali jste neplatnou hodnotu');
end

cH=exp(deltavH/Isp);    % výpočet koef. C z Tsiolkovského rovnice
cB=exp(deltavB/Isp);    % výpočet koef. C z Tsiolkovského rovnice
                        % výstupy
fprintf('Poměr m0/m pro Hohmannův manévr: %g \n', cH);
fprintf('Poměr m0/m pro bi-eliptický manévr: %g \n', cB);

```