

SZÁMÍTÓGÉPES SZIMULÁCIÓK

Bolygómozgás jegyzőkönyv



Jegyzőkönyvet készítette:
Koroknai Botond (AT5M0G)

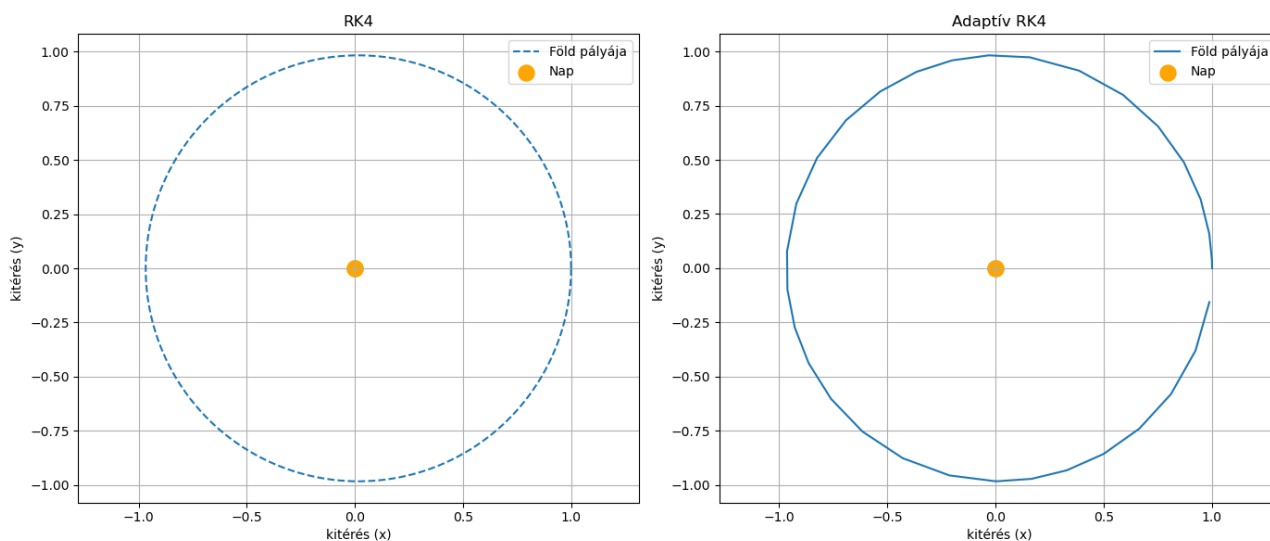
Jegyzőkönyv leadásának időpontja:
2024.03.28

1. Nagytengely vizsgálata

A feladat során a nagytengely hosszának változását és a szögének eltérését kellett vizsgálnunk. A szimulációhoz a Nap-Föld rendszert választottam, ami paraméterek szempontjából a következőt jelenti:

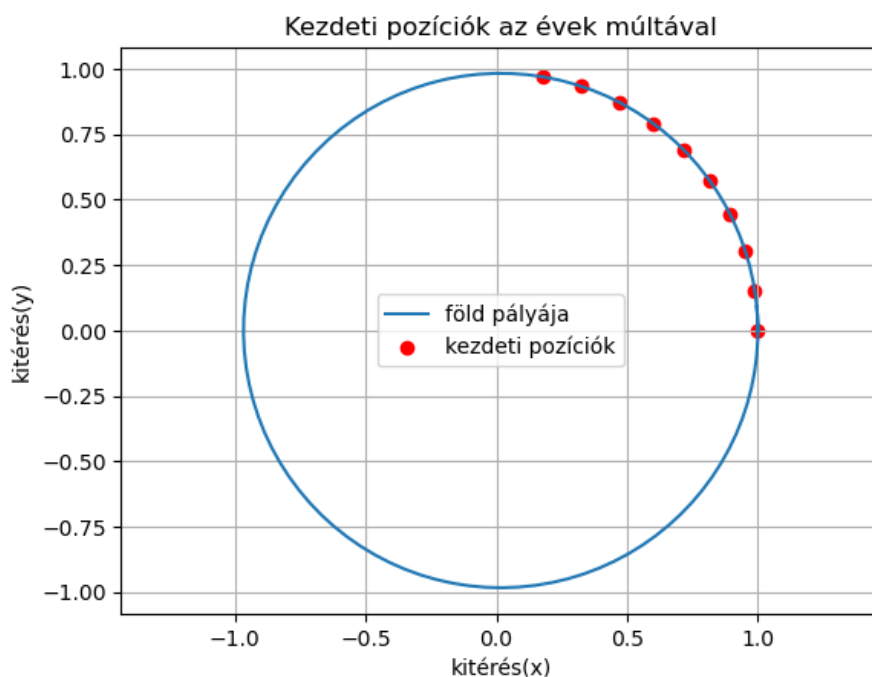
- aphélium = 1 CSE
- excentricitás = 0.016710219
- integrálási idő = 0.001
- adaptív RK4 pontossága = 10^{-5}

Annak érdekében, hogy ellenőrizsem, hogy a szimuláció helyesen futott-e le, mindkét módszer esetén ábrázoltam a föld pályáját 1 periódus erejéig.

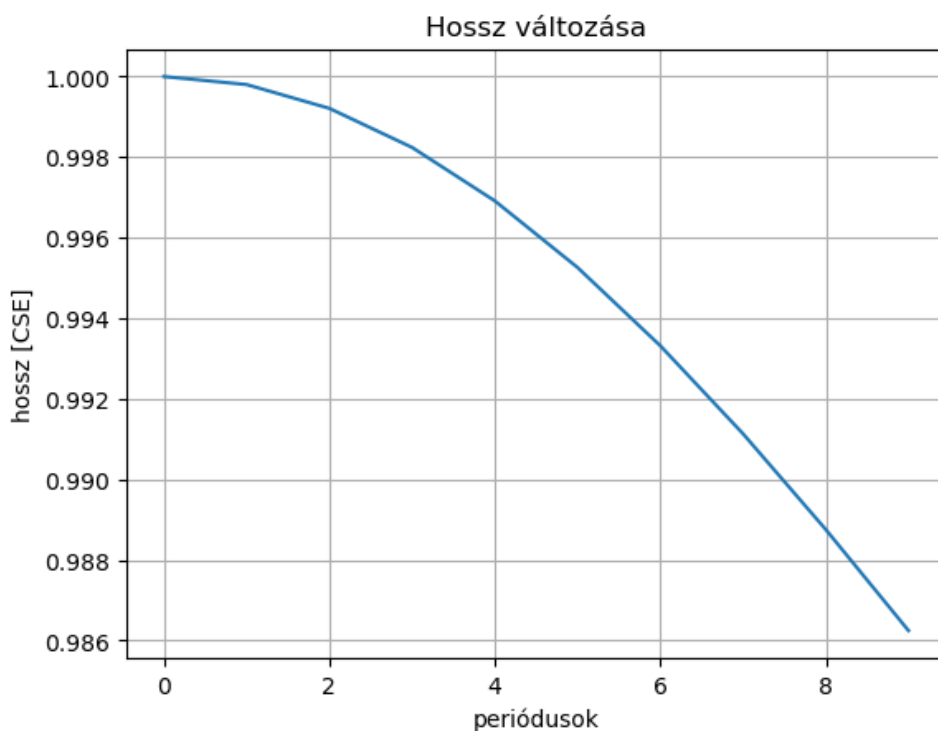


1.1. A nagytengely kezdeti szögének vizsgálata

A szimulációt mind a két esetben 10-10 periódusig futtattam, és hogy a szögeltérést vizsgáljam mindig az adott periódus kezdetéhez tartozó x,y koordinátákat kiválogattam.



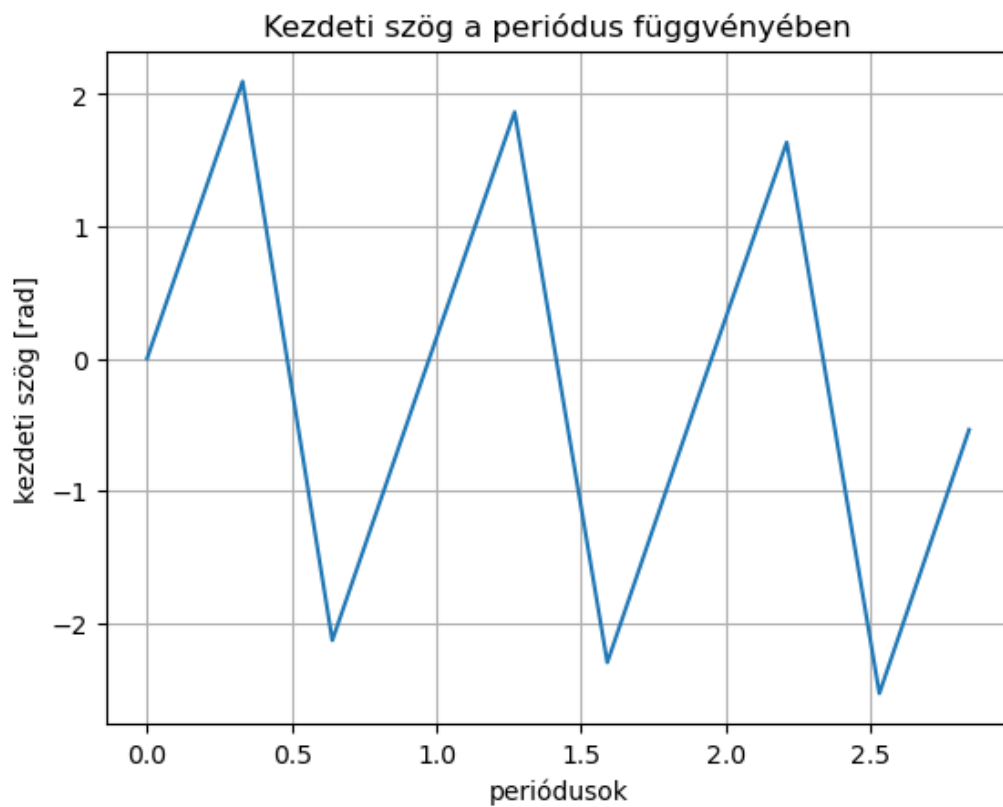
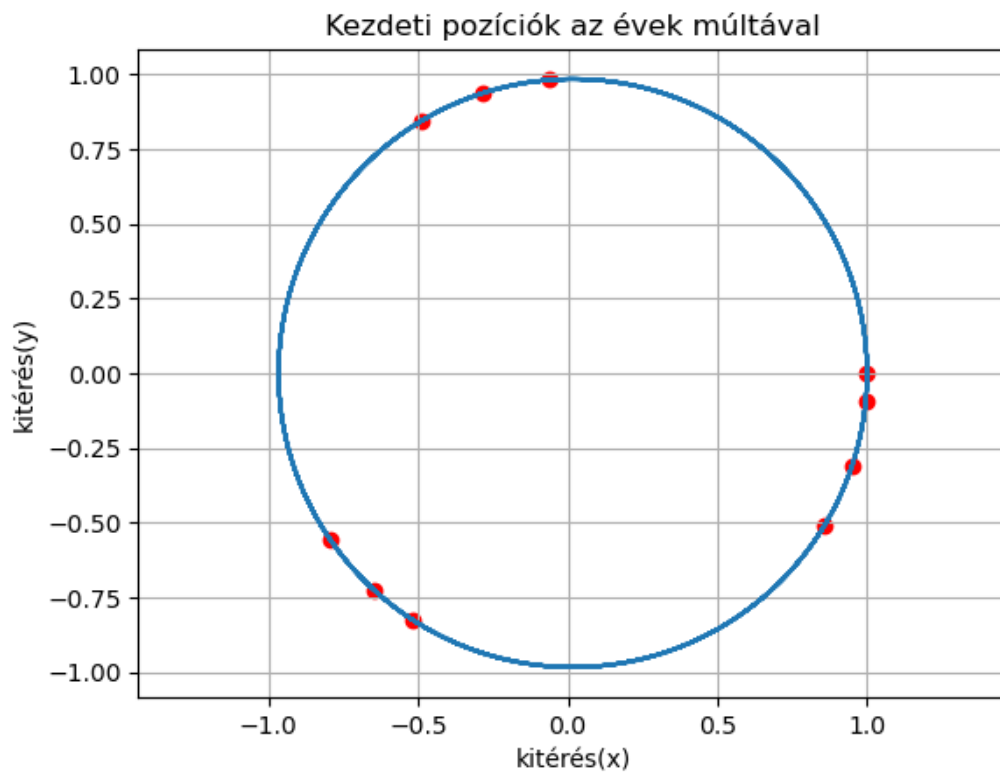
A hozzájuk tartozó szögeket úgy számoltam ki, hogy vettem az y és x koordináták arkusz tangensét. A kezdeti kitérésben lineáris növekedést tapasztaltam az idő előrehaladtával:



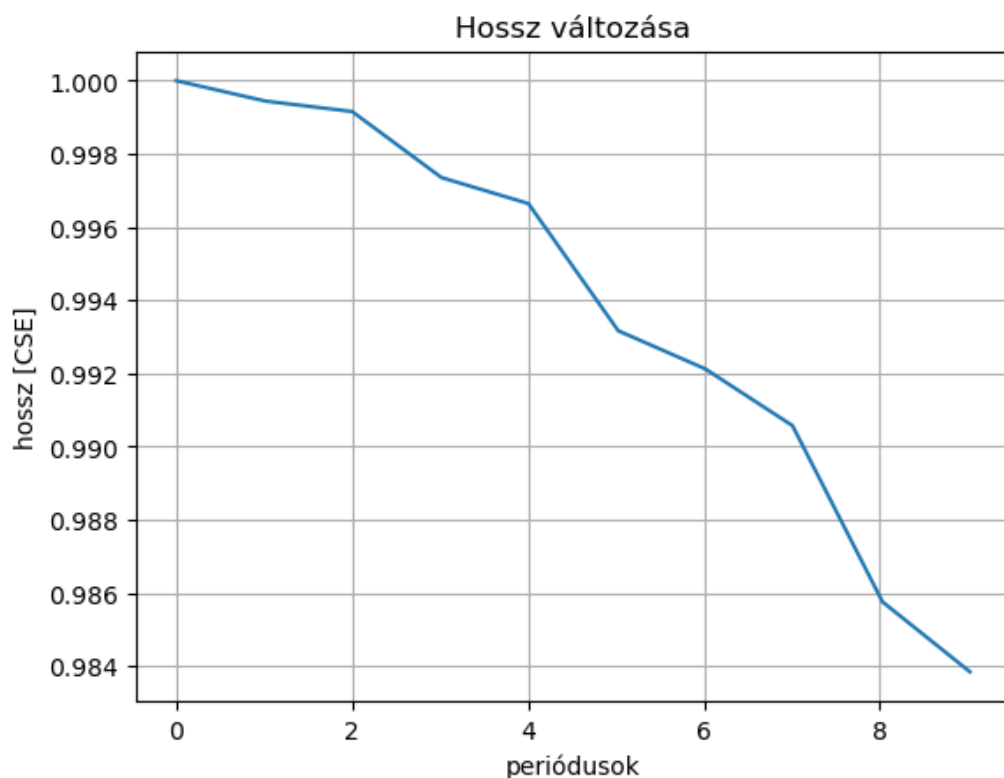
A tengely hosszakat a

$$\sqrt{x^2 + y^2}$$

képlettel számoltam, és csökkenést tapasztaltam. Ezt követően megvizsgáltam, hogy adaptív RK4 esetén milyen változásokat tapasztalhatunk. Az adaptív lépéshossznak köszönhetően nem tudtam pontosan 1 év, 2 év, stb. idő eltérés esetén vizsgálni a szögkitérést, így mindig az ezekhez legközelebbi időértékeket választottam ki.



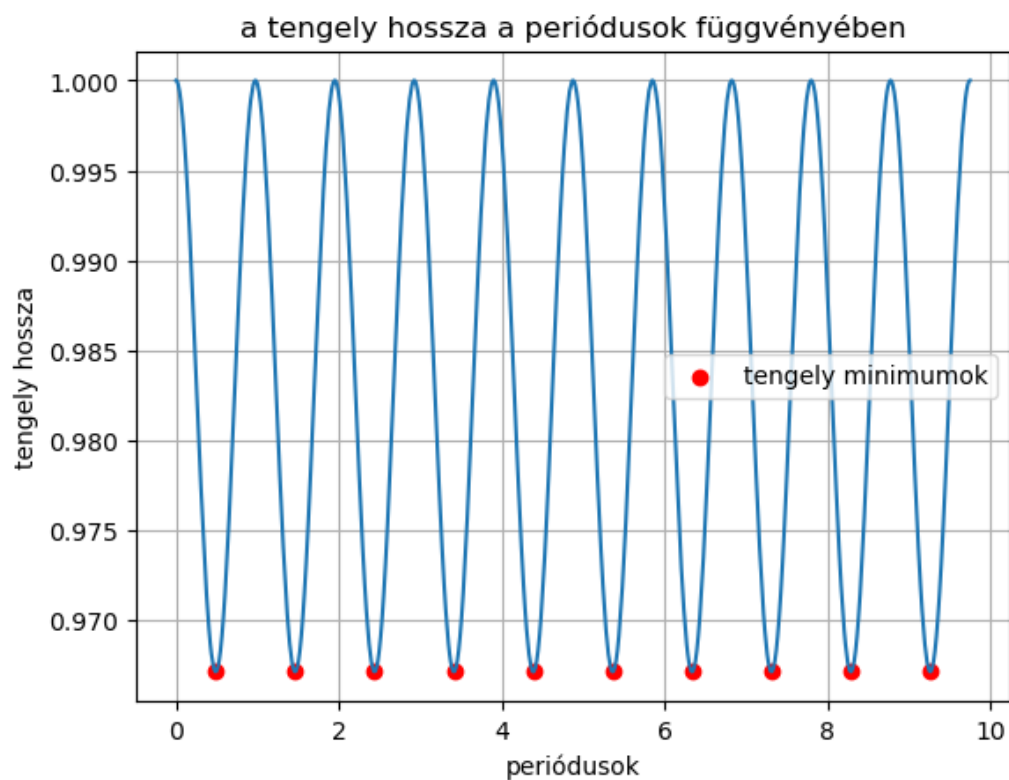
Az adaptív módszer mellett jóval nagyobb ingadozás tapasztalható a kezdeti kitérésben. Viszont itt is elmondható, hogy az abszolút eltérés a kezdeti értéktől nő az idő előrehaladtával.



A hosszváltozásban megintcsak csökkenést tapasztalok az adaptív esetben is, ugyanakkor a szögváltozásban megjelenő periodicitás is kirajzolódik.

1.2. Perihélium megkeresése

A perihélium megkereséséhez ábrázoltam a nagytengely hosszát az idő függvényében.



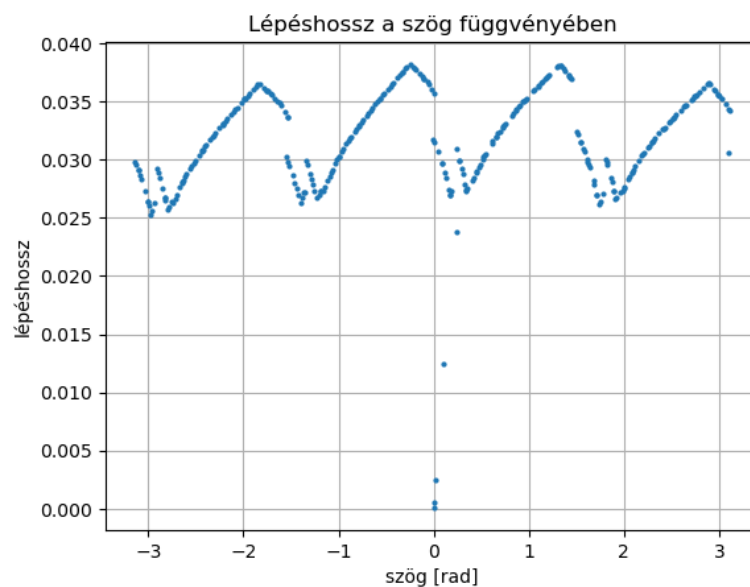
A minimumok értékét kiátlagolva megkaptam a perihélium távolságát, ami 0.967129 CSE-nek adódott, azaz 0.0328 CSE különbség van az aphélium, valamint a perihélium között. A Wikipédia szerint a perihélium

távolsága: 0.983289 CSE. A nagy eltérés annak köszönhető, hogy míg én 1 CSE-nek vettem a perihéliumot, a valódi, pontosabb értéke: 1.016710. Ha a Wikipédián található két értéket szintén kivonjuk egymásból, akkor egy 0.0334 CSE nagyságú értéket kapunk. A szimuláció ez alapján igencsak pontosnak mondható a két érték között mindössze 1.644 %-os eltérés van. Megvizsgálva az esetet az adaptív RK4-el hasonló eredményeket kaptam. A perihélium távolságára: 0.967163 CSE jött ki eredményül, ami így mindössze $3.39 \cdot 10^{-5}$ eltérést mutat az előző módszerhez képest.

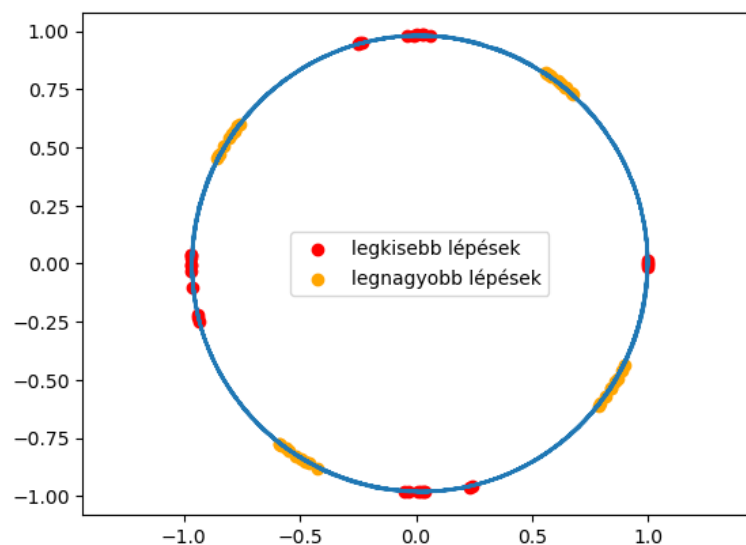
2. Lépéshossz vizsgálata a pálya mentén

Az adaptív RK4 lépéshosszának vizsgálatához két különböző szimulációt végeztem el, az egyik esetben az integrálás pontosságát $1 \cdot 10^{-5}$ -nek választottam, míg második esetben jóval nagyobb $1 \cdot 10^{-10}$ pontosságot követeltem meg. A kiértékelés során a lépéshosszt ábrázoltam a szög függvényében.

2.1. Alacsonyabb pontosság



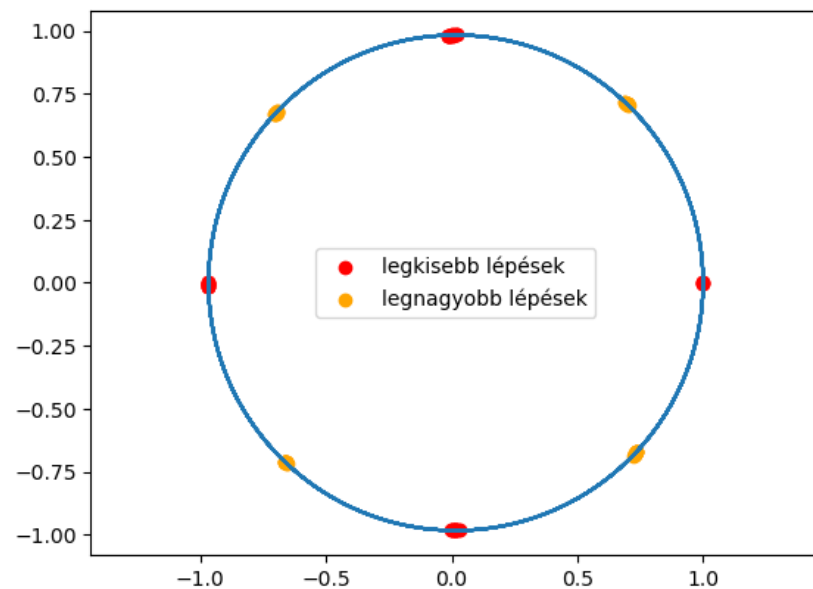
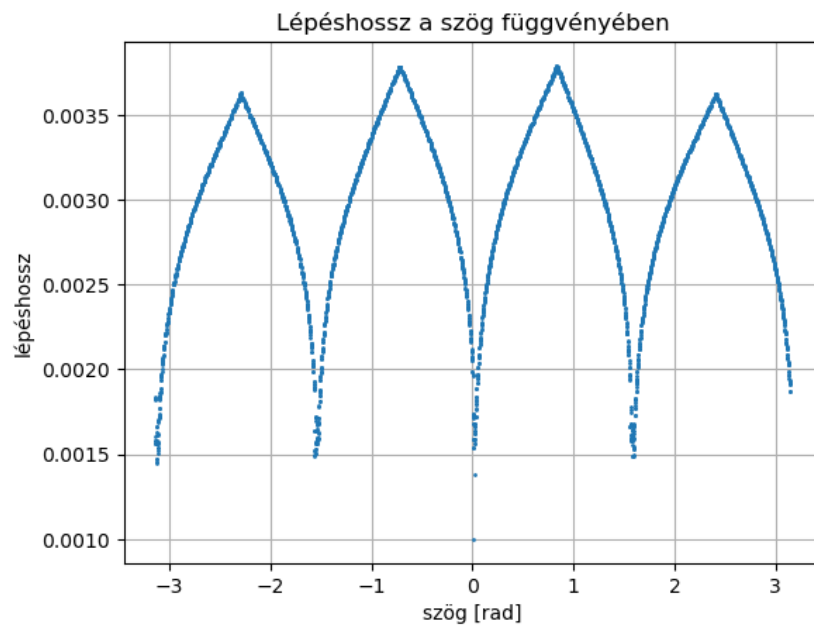
Hogy pontosabban lássuk mi történik megkerestem a leghosszabb lépésekhez tartozó x,y koordináták értékét (felső csúcsok), valamint a legkisebb lépésekhez tartozó koordináták értékét (alsó csúcsok) és ábrázoltam őket a pálya mentén.



Ha körbejárjuk a pályát az $x = 1, y = 0$ ponttól az óramutató irányának ellenkezően, akkor a piros pontoktól a

narancssárga pontokig lévő szakaszon nő a lépéstávolság, majd a narancssárga pontoktól a piros pontokig tartó szakaszon visszacsökken.

2.2. Magas pontosság



Itt is hasonló eredményeket kaptam, annyi különbséggel, hogy a lépéshossz ingadozás végpontjai pontosabban kirazolódtak.

2.3. Integráláshoz szükséges idő

A sima RK4 lépéshosszát 0.001-re állítottam. Az adaptív esetben, mind a az alacsonyabb mind a magasabb pontosság megkövetelése mellett nagyobb lépésközök jelennek meg, ami így kevesebb lépést és gyorsabb integrálást jelent.

3. Merkúr perihélium precessziója

3.1. Változtatások a kódon

Első lépésben a GmPLusM változó értékét cseréltem ki a

$$G(M + m) = \frac{4\pi^2 a^3}{t^2}$$

képletnek megfelelően. A félnagy tengely hosszának: 0.3871 CSE távolságot használtam, a keringési időnek: 0.24085 évet, így a GmplusM változó értékére: 39.47694-et kaptam. Ezt követően a differenciál egyenletek kiegészítetem a megkövetelt korrekcióval:

1. kód. A Nap-Merkúr rendszer differenciálegyenletei

```
Vector f(const Vector &x)
{
    double t = x[0], r_x = x[1], r_y = x[2], v_x = x[3], v_y = x[4];
    double rSquared = r_x * r_x + r_y * r_y;
    double rCubed = rSquared * sqrt(rSquared);
    Vector f(5);
    f[0] = 1;
    f[1] = v_x;
    f[2] = v_y;

    f[3] = -GmPlusM * r_x / rCubed * (1 + alpha / rSquared);
    f[4] = -GmPlusM * r_y / rCubed * (1 + alpha / rSquared);
    if (switch_t_with_y)
    {
        // use y as independent variable
        for (int i = 0; i < 5; i++)
            f[i] /= v_y;
    }
    return f;
}
```

3.2. Kiértékelés

A szimulációt követően megkerestem az első perihéliumhoz tartozó szöget, valamint a századik évhez tartozó perihéliumhoz tartozó szöget, és megvizsgáltam a különbségüket.

3.3. Sima RK4

A hagyományos RK4 esetén a kettő perihélium között 0.33 ° eltérést tapasztaltam, amit 3600-al megszorozva 1188 ívmásodperct eltérést kaptam.

3.4. Adaptív RK4

Az adaptív esetben 0.0704 ° eltérést kaptam, ami 253.7 ívmásodperccel felel meg.

3.5. Diskusszió

A perihélium precessziójának irodalmi értéke 43 ívmásodperc. Mint látjuk mindkét szimuláció helytelen eredményeket adott, viszont azt megállapíthatjuk, hogy az adaptív RK4 arányaiban nézve sokkal kisebb hibát vétett a hagyományos RK4-hez képest.

4. Háromtest-probléma

A háromtest probléma megoldásához a következő módosításokat végeztem a kódon a feladathoz tartozó diasoron bemutatott példának megfelelően:

2. kód. A háromtest-probléma differenciálegyenletei

```
Vector f(const Vector &x)
{
    double r1x = x[0], r1y = x[1]; // Position of body 1
    double v1x = x[2], v1y = x[3]; // Velocity of body 1

    double r2x = x[4], r2y = x[5]; // Position of body 2
    double v2x = x[6], v2y = x[7]; // Velocity of body 2

    double r3x = x[8], r3y = x[9]; // Position of body 3
    double v3x = x[10], v3y = x[11]; // Velocity of body 3

    double r12_squared = (r1x - r2x) * (r1x - r2x) + (r1y - r2y) * (r1y - r2y);
    double r13_squared = (r1x - r3x) * (r1x - r3x) + (r1y - r3y) * (r1y - r3y);
    double r23_squared = (r2x - r3x) * (r2x - r3x) + (r2y - r3y) * (r2y - r3y);

    double r12_cubed = r12_squared * sqrt(r12_squared);
    double r13_cubed = r13_squared * sqrt(r13_squared);
    double r23_cubed = r23_squared * sqrt(r23_squared);

    Vector f(12);

    // Derivatives for body 1
    f[0] = v1x;
    f[1] = v1y;
    f[2] = -G * m2 * (r1x - r2x) /
    fabs(r12_cubed) - G * m3 * (r1x - r3x) / fabs(r13_cubed);
    f[3] = -G * m2 * (r1y - r2y) /
    fabs(r12_cubed) - G * m3 * (r1y - r3y) / fabs(r13_cubed);

    // Derivatives for body 2
    f[4] = v2x;
    f[5] = v2y;
    f[6] = -G * m3 * (r2x - r3x) /
    fabs(r23_cubed) - G * m1 * (r2x - r1x) / fabs(r12_cubed);
    f[7] = -G * m3 * (r2y - r3y) /
    fabs(r23_cubed) - G * m1 * (r2y - r1y) / fabs(r12_cubed);

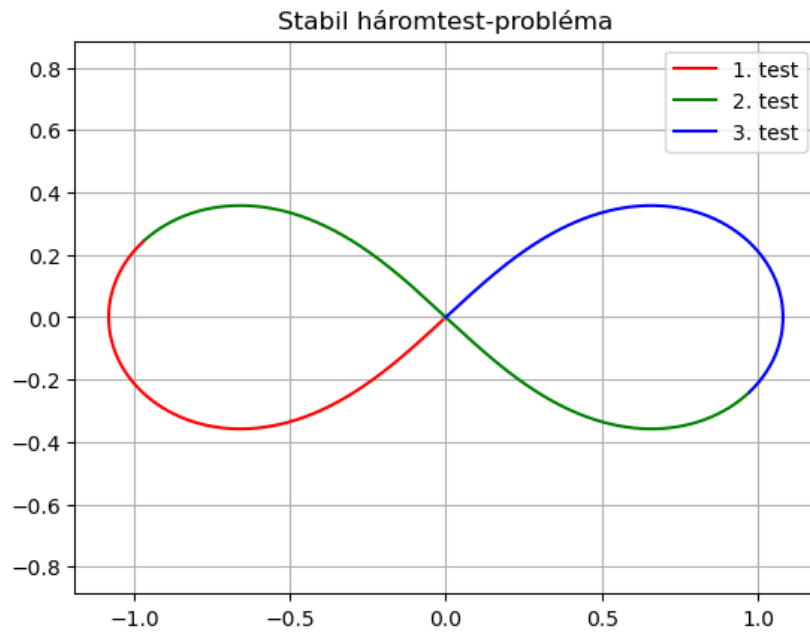
    // Derivatives for body 3
    f[8] = v3x;
    f[9] = v3y;
    f[10] = -G * m1 * (r3x - r1x) /
    fabs(r13_cubed) - G * m2 * (r3x - r2x) / fabs(r23_cubed);
    f[11] = -G * m1 * (r3y - r1y) /
    fabs(r13_cubed) - G * m2 * (r3y - r2y) / fabs(r23_cubed);

    return f;
}
```

5. Stabil megoldás

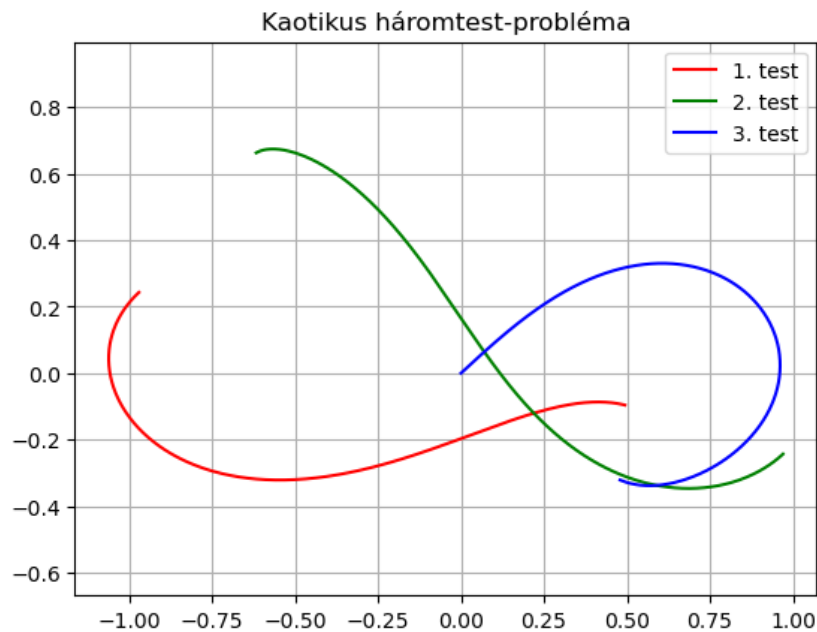
Az első szimulációhoz egy olyan rendszert választottam, amit 1993-ban a Santa Fe Intézet fizikusa, Cris Moore numerikus úton fedezett fel. Három egyenlő tömegű testet ír le, amelyek egy nyolcas alakzat mentén mozognak. Egy ilyen rendszer megvalósításához, indítsuk el mindhárom bolygót úgy, hogy egy legyen a

középpontban. A két külső bolygó kezdeti sebessége egyenlő legyen és merőleges a három bolygó közötti egyenesre. Végül a központi bolygó sebessége legyen a két külső bolygó sebességének kétszerese, és ellenkező irányú.



5.1. Kaotikus megoldás

Annak érdekében, hogy demonstráljam, hogy mennyire fontos a kezdeti paraméterek megválasztása stabilitási szempontból, az m_2 , és m_3 értékét 1.1, és 1.2 növeltem.



A rendszer már ilyen kis változtatások esetén is kaotikus viselkedésbe kezd.

6. Lagrange-pontok

Az utolsó feladatrészt számos próbálkozást követően sajnos nem tudtam megvalósítani.