

# N-test szimuláció

Krucsay András

2025. 05. 04.

## A programról általánosságban

A program fő feladata egy N-test szimuláció során megvizsgálni, hogy a Jupiter megléte mennyire befolyásolja egy, a Naprendszeren kívülről érkezett objektum Földbe csapódását.

## A program fizikai háttere

Az alap elképzelés szerint a Naprendszer egy egyszerűsített változatával számolunk, ahol minden bolygó kezdetben körpályán mozog a Nap körül. Természetesen ezt befolyásolja a többi égitest gravitációs ereje, de ez csak hosszútávon okozna problémát, éppen ezért rövidebb, pár éves szimulációkat futtatunk csak. Ehhez viszont fontos, hogy a külső objektum elég gyorsan haladjon, hogy a vizsgált időtartamban elérhesse a bolygót, ezért a külső objektumok a lehető legnagyobb sebességgel haladnak, amit ezen honlap alapján határoztam meg.

Ezen kívül egyszerűbb, két dimenziós síknak van tekintve a Naprendszer, de a programban van lehetőség a harmadik térdimenzió beállítására is (három dimenziós vektorok vannak, de jelenleg a harmadik komponensek mind 0-ák).

A külső objektum fix hellyel rendelkezik, míg a változó a sebességvektor iránya. Ez az irány a szimulációk során egyenletesen változik 0-tól  $2\pi$ -ig, lefedve mindent. Minél nagyobbra állítjuk a szimulációk számát, annál több részre fogja megvizsgálni, így csupán nagyobb teljesítményű számítógépre és több időre lenne szükség a jobb eredmények érdekében. A bolygók kerekített adatai a Wikipédia különböző cikkeiből, a külső objektum tömege meg a ChatGPT javaslata alapján került meghatározásra.

Az erők, sebességek és pozíciók a Newton-féle gravitációs törvény alapján és 4-edrendű Runge-Kutta-módszerrel kerültek számításra,  $G = 6.67430 \cdot 10^{-11}$  értékkel.

## A programról részletesebben

A program fő adatstruktúrája a `Body` struktúra, amely egy égitest tömegét, pozícióját és sebességét tartalmazza. A vektoros műveletekhez egy `Vector3` segédstruktúra lett definiálva.

Az égitestek adatai `unordered_map`okban vannak tárolva. A bolygók kezdőpozíciója véletlenszerű szög alapján generált, fix nagyságú körpálya mentén történik (de fix seeddel a reprodukálhatóság érdekében). Ha nincs megadva egy égitest adata valamelyik mapben, akkor a program generál neki egy fix  $10^6$  m sugarat és  $10^{20}$  kg tömeget. Pályasugárra és keringési sebességre nincs, a relatív stabil körpályához több számítás kéne ennél, ami több időt igényelne.

Az ütközést a testek közti távolság alapján érzékeli a program. Ha a távolság kisebb, mint a testek sugarának összege szorozva egy biztonsági tényezővel (jelen esetben 1.5), akkor ütközés történik. Ha a program ütközést érzékel, akkor azt hozzáadja egy változóhoz és átugorja a szimulációt, elkezdve a következőt (ha van).

A szimuláció során használt paraméterek:

- Időlépés: 3600 másodperc / 1 óra
- Szimuláció hossza: 5 év
- Sebességvektorok/szimulációk száma: 1000
- Kezdeti távolság a külső objektumnál:  $6 \cdot 10^{12}$  m

A szimuláció kétszer fut le: Jupiterrel és nélküle. A szimuláció végén a program kiírja a Földdel való ütközések számát a két esetben, külön-külön.

Ezen kívül hozzá van adva egy `debugMode` változó a program elejéhez, ellenőrzési célzattal, hogy a program megfelelően működik-e. True esetén debug mód van érvényben és csak a legelső szimuláció legelső pillanatát írja ki a program, míg false esetén normál módban rendesen végigfut a szimuláció `velocityIterations`  $\times 2$  alkalommal.

## Eredmények

A program eredményeket nem tudott produkálni, mindkét esetben 0 jött ki a Földdel való ütközések számának. Mivel különösebb hibát nem sikerült észrevenni a programban, valószínűleg nagyobb számítási kapacitás és idő birtokában lennének eredmények is (vagy csak többször is szerencsétlen Naprendszer-generálás történt). Mindenesetre a program könnyen változtatható, akár Pythonos vizualizálás vagy jobb számítás érdekében (pl. a több és pontosabb futtatáshoz átírva a paramétereiket).