PLANETENSTELSELS - WERKCOLLEGE 3 EN 4

Opdracht 3: Baanintegratie: Planeet in een dubbelstersysteem

In de vorige werkcolleges heb je je pythonkennis opgefrist. Je hebt een aantal fysische constanten ingelezen, een paar simpele berekeningen uitgevoerd en een plot gemaakt. In deze opgave gaan we deze kennis gebruiken om de baan van een exoplaneet in een dubbelstersysteem te **simuleren**. Dit is uitgebreide procedure. Lees deze instructies daarom goed.

Sterren en planeten bewegen zoals bekend onder invloed van de zwaartekracht. In een enkel-ster systeem bewegen de ster en de planeet in ellipsbanen om het zwaartepunt. Wanneer er een tweede ster aan het systeem wordt toegevoegd, zal de planeet over het algemeen niet in een ellipsbaan kunnen bewegen.

In dit scenario rekenen we de onderlingen krachten van 2 sterren en 1 planeet uit in een tweedimensionaal vlak. In werkelijkheid trekken alle drie deze objecten aan elkaar, en zijn er dus 6 krachten in het spel. We maken in deze opgave echter de benadering dat de sterren veel zwaarder zijn dan de planeet. Zodoende is de zwaartekracht die de planeet op de beide sterren uitoefent verwaarloosbaar, en hoeven we nog maar 4 krachten te berekenen: De zwaartekracht tussen beide sterren, en de zwaartekracht die de planeet van beide sterren ondervindt.

De banen van deze 3 objecten gaan we **iteratief** uitrekenen. Dat wil zeggen dat we de berekening stapsgewijs uitvoeren: We rekenen uit hoe groot de kracht is, gaan na hoeveel de positie van het object verandert gedurende een korte tijdstap dt en herhalen dit ad infinitum. Hier onder staat hoe dit stapsgewijs gaat. Je kan deze opgave benaderen in termen van **grootheden**, maar het is makkelijker is om in **vectoren** te denken:

- Definiëer de massa's van de objecten m_1, m_2 en m_3, waarbij $m_1, m_2 >> m_3$.
- Definiëer de beginposities van de sterren en de planeet op tijdstip $t = 0$: $\vec{r_i}(t = 0)$. $\vec{r_i}$ is een vector met daarin de x en y coördinaten van m_i: $[x, y]$. $i = 1, 2, 3$ slaat op de drie massa's zoals boven.
- Definiëer evenzo de beginsnelheden: $\vec{v_i}(t = 0)$.
- Reken de 4 relevante krachten tussen m_i en m_j uit: $\vec{F_{ij}} = \frac{-Gm_im_j}{r_{ij}^2}\frac{\vec{r_{ij}}}{\|\vec{r_{ij}}\|}$. Hierbij is r_{ij} de afstands**vector** tussen de objecten i en j.
- Tel voor alle objecten de relevante krachten bij elkaar op (de sterren voelen elk maar 1 kracht, maar de planeet voelt er 2: $F_3 = F_{13} + F_{23}$).

Date: 19 februari 2015.

- Nu weet je de versnellingen $\vec{a_1}$, $\vec{a_2}$ en $\vec{a_3}$ die de 3 objecten ondergaan op $t = 0$. Reken uit hoe de snelheid als gevolg van deze versnellingen verandert **na een kleine tijdstap d**t: $\vec{v_i}(\mathrm{d}t) = \vec{v_i}(0) + \mathrm{d}t * \vec{a_i}$.
- Doe hetzelfde voor de nieuwe posities: $\vec{r_i}(\mathrm{d}t) = \vec{r_i}(0) + \mathrm{d}t * \vec{v_i}(\mathrm{d}t$.
- Reken opnieuw de krachten tussen de objecten uit. Dit is de kracht die ze op $t = \mathrm{d}t$ voelen.
- Dit geeft je nieuwe versnellingen...
- ...en nieuwe posities...
- etc.
- Wanneer je dit N keer gedaan hebt, kun je x en y tegen elkaar plotten, en krijg je de baan te zien, zie figuur 1.

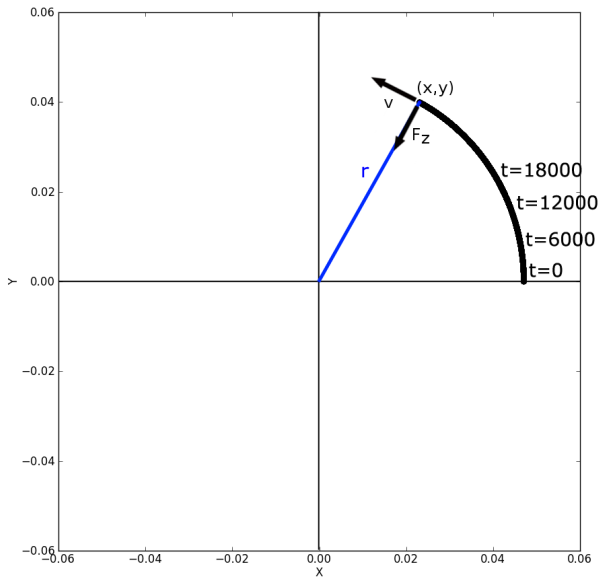

FIGURE 1. Baanintegratie

Om deze simulatie te doen moet je stap 4 t/m 7 dus N keer herhalen. Dit doen we door een *for*-loop te gebruiken (zie het voorbeeld hieronder). Voordat we beginnen zijn er echter twee dingen die je je moet realizeren: Deze simulatie maakt de benadering dat de tijdstap dt verwaarloosbaar klein is. Hoe kleiner je dt dus kiest, hoe nauwkeuriger de simulatie zal zijn. Echter, je simulatie bestaat uit N stappen. De totale tijdsduur die de simulatie bedraagt, is dus $\mathrm{d}t \times N$. Zo kun je 3600 stappen van 1 seconde nemen, of 60 stappen van 1 minuut. In beide gevallen zal de simulatie hetzelfde tijdvak beslaan, maar 3600 stappen kosten natuurlijk meer rekenkracht (lees: computertijd) dan 60 stappen.

Je wilt N en dt dus zodanig kiezen dat je simulatie redelijk nauwkeurig is, maar niet onredelijk lang duurt om uit te rekenen.

```
x[1]  = x[0]+v[0]*dt
v[1]  = v[0]+a[0]*dt
a[1]  = f(x[1])
t[1]  = t[0]+dt
...
x[i]  = x[i-1]+v[i-1]*dt
v[i]  = v[i-1]+a[i-1]*dt
a[i]  = f(x[i])
t[i]  = t[i-1]+dt
...
x[N]  = x[N-1]+v[N-1]*dt
v[N]  = v[N-1]+a[N-1]*dt
a[N]  = f(x[N])
t[N]  = t[N-1]+dt
```

Voor het gemak beginnen we deze opgave met een enkele planeet die om een enkele ster draait. Daarna voegen we de tweede ster toe. **Let ten alle tijde op welke eenheden je gebruikt!**

(1) Voordat de *for*-loop begint moeten er een aantal parameters vastgesteld worden. Definiëer de tijdstap d$t = 3600$ seconden, en het aantal tijdstappen $N = 5,000$. Definiëer ook arrays x en y, waarin voor elke tijdstap de $x(t)$ en $y(t)$ zal worden opgeslagen. $\vec{x}$ heeft dus de lengte N.

(2) Definiëer ook de fysische constanten die nodig zijn om de zwaartekracht uit te rekenen. Een aantal hiervan heb je al in een module gedefiniëerd. Stel verder $M_* = 1.0M_\odot$, $m_\mathrm{p} = M_\mathrm{Aarde}$.

(3) Definiëer tot slot de beginwaarden van de simulatie: De ster begint op $(x(0), y(0)) = (0, 0)$ met snelheid $(v_x(0), v_y(0)) = (0, 0)$ (en zal daar overigens blijven omdat we de kracht tussen de planeet en de ster verwaarlozen). De planeet begint op $(x(0), y(0)) = (1\mathrm{AU}, 0)$, met snelheid $(v_x(0), v_y(0)) = (0, 28\mathrm{km/s})$.

(4) In het voorbeeld hierboven zie je hoe de berekening start. Wat is de positie *x[1],y[1]* van de planeet na de eerste stap? *Tip: Posities, snelheden, versnellingen en krachten zijn vectoren. Het is niet verplicht om ze als vectoren uit te rekenen (je kunt de x en y componenten onafhankelijk van elkaar uitrekenen), maar het kan je code wel aanzienlijk minder verwarrend maken. De positie $\vec{r}$ kun je in python namelijk gewoon definiëren als vector $[x, y]$.*

(5) Hoeveel is de planeet gedurende de eerste stap dus opgeschoven in de y-richting? Kun je dit verklaren?

(6) Reken nu uit, met behulp van de formule voor de zwaartekracht, welke **versnelling** de planeet op tijdstip $t = 0$ ondervindt, en wat dus de snelheid op $t = 1$ is geworden.

(7) Nu moeten deze berekeningen N keer herhaald worden. Doe dit in een *for*-loop, en plot de resulterende baan door x tegen y uit te zetten.

(8) Wat gebeurt er als je de beginsnelheid van de planeet vermindert tot 3km/s?

(9) Wat gebeurt er als je nu de tijdstap veel groter maakt, bijvoorbeeld 1 dag?

(10) Nu voegen we de tweede ster toe. We maken de eerste ster $M_1 = 1.2M_\odot$ en de tweede $M_2 = 0.6M_\odot$. Zet de beginposities op $\vec{r}(0)_1 = (-0.2\text{AU}, 0)$ en $\vec{r}(0)_2 = (0.4\text{AU}, 0)$, en de beginsnelheden op $\vec{v}(0)_1 = (0, -12\text{km/s})$ en $\vec{v}(0)_2 = (0, 24\text{km/s})$.

(11) Voeg de tweede ster ook toe in de *for*-loop. Eerst rekende je alleen de positie, snelheid, versnelling en kracht op de planeet uit. Nu moet je dus ook de veranderende posities, snelheden, versnellingen en krachten op de sterren uitrekenen. Tel met name de krachten goed op! Ster 1 voelt een kracht van ster 2. Ster 2 voelt een kracht van ster 1 De planeet voelt een kracht van beide sterren: $F_p = F_1 + F_2$.

(12) Zet de planeet op dezelfde beginpositie als voorheen (dus 1 AU met een snelheid van 28km/s, maar met een kleinere dt). Plot, beschrijf en interpreteer de resulterende banen van de sterren en de planeet. Is de planeet baan stabiel over langere tijd? Wat gebeurt er uiteindelijk?

(13) Wat zou je kwalitatief met de beginpositie -en snelheid van de planeet moeten doen om de baan stabieler te maken? Waarom?

(14) Zet de sterren op $\vec{r}(0)_1 = (-1.0\text{AU}, 0)$ en $\vec{r}(0)_2 = (2.0\text{AU}, 0)$, en de beginsnelheden op $\vec{v}(0)_1 = (0, -8\text{km/s})$ en $\vec{v}(0)_2 = (0, 16\text{km/s})$, en vind een planeetbaan waarbij de planeet stabiel om een van beide sterren draait.

(15) Wat valt je op aan de verhouding tussen de massa's van de sterren, de verhouding tussen hun beginsnelheden en de verhouding tussen de beginposities? Wat gebeurt er als je deze verhoudingen verstoort door bijvoorbeelde de massa van een van de sterren te veranderen?

(16) Tot nu toe hebben we de sterren en de planeet allebei tegen de klok in laten draaien. Denk je dat dat uitmaakt? Wat denk je dat er in algemene zin gebeurt als je de sterren tegen de klok in laat draaien, maar de planeet met de klok mee (dat wil zeggen, een negatieve beginsnelheid)? Probeer dit uit.

(17) We verwaarlozen constant de kracht van de planeet op de sterren. Eigenlijk verwaarlozen we daarmee dus de massa van de planeet. Wat gebeurt (kwalitatief) er als je de kracht van de planeet op de sterren niet zou verwaarlozen? Stel je voor dat het een zware planeet zou zijn, met $m_p \sim 0.2M_\odot$?

Uitwerking

Maak een klein verslag over je antwoorden.
Let bij het maken van je uitwerking erop dat antwoorden met de juiste (en astronomisch zinnige!) eenheden gegeven worden en met een beredenering toegelicht worden. Ze hebben een zinnig aantal significante cijfers hebben (de afstand tussen Jupiter en de Zon is dus geen 778547200000m, maar 5.20AU).
Laat ook zien hoe je berekeningen hebt gedaan. Doe je dit niet, dan zal de vraag niet goed gerekend worden!
Antwoorden horen in het verslag te staan, de python code moet niet gebruikt hoeven worden om de antwoorden te krijgen.
Let erop dat alle grafieken een titel hebben en dat er eenheden langs de assen staan. Bedenk of de eenheden die je ziet op de assen, goed kunnen zijn. Leg uit wat er in een plot te zien is en of het resultaat is wat je verwacht.

De uitwerkingen moeten individueel digitaal ingeleverd worden (graag als PDF) met naam en studentnummer erop en de python code die je hebt geschreven moet mee gestuurd worden als losse file die meteen uit te voeren is (dus niet als tekst in het verslag). **Deadline voor het inleveren is 23 maart 2015 (begin hoorcollege).** Uitwerkingen kun je sturen naar herbonnet@strw.leidenuniv.nl (Linuxzaal 4de verdieping) of hoeijmakers@strw.leidenuniv.nl (Windowszaal 3de verdieping). Zodra het is nagekeken, ontvang je een email met commentaar en/of correcties op jouw ingeleverde werk.

Nuttige webadressen

- Website van het college:
  `http://home.strw.leidenuniv.nl/ linnartz/2015.html`
- Introductie in het gebruik van `python` is te vinden op
  `http://www.strw.leidenuniv.nl/local/computers/it_local/practicum/html/python.php?node=7` (deze link vind je ook op de website van het college).
- Informatie over `numpy` is te vinden op `http://numpy.scipy.org/` (deze link vind je ook op de website van het college).