

PLANETENSTELSELS - COMPUTER PRACTICUM 1 EN 2

INLEVEREN OPDRACHTEN EN EISEN VERSLAG

Integraal onderdeel van het college Planetenstelsels zijn de computer practica en werkcolleges. In deze korte handleiding hebben we het alleen over de computer practica.

In totaal zijn er drie opdrachten, waaraan jullie werken (onder begeleiding) tijdens zes computer practica. Jullie doen dat in groepen van twee of maximaal (en alleen als het niet anders kan) drie personen. Om te beoordelen of je de opdracht hebt begrepen, moet je een kort verslag inleveren. Je levert een verslag per groep in en dat doe je uiterlijk voor 12 uur op de inleverdag. De inlever data staan duidelijk vermeld op de homepage van het college.

Vertel in het verslag welke aannames je maakt, wat voor waardes je voor constanten aanneemt, en laat formules die je gebruikt ook expliciet zien. Werk systematisch en laat stap voor stap zien hoe je te werk bent gegaan en concludeer met een duidelijk eindresultaat dat je kort interpreteert en bespreekt. Werk netjes en leesbaar, gebruik zinvolle eenheden, let erop dat alle grafieken een titel hebben, dat er eenheden op de assen staan, leg uit wat je afbeeldt in de grafiek, of het resultaat aan je verwachtingen voldoet ... werk academisch. Voeg altijd de gebruikte `python` code toe.

Heel belangrijk: werk onafhankelijk. Het is de bedoeling dat iedere groep zijn eigen verslag schrijft en dat in dit verslag een eigen `python` script is bijgevoegd.

Inleveren per mail bij Vincent (kofman@strw.leidenuniv.nl) als je op de 3e verdieping werkt of Steven (stevenbos@strw.leidenuniv.nl) als je op de 4e werkt. Lever in voor 12:00 op de dag van de deadline. Verslagen zonder bijgevoegde `python` code worden niet nagekeken.

OPDRACHT 1: INTRODUCTIE PYTHON

Tijdens dit computer practicum en tijdens je verdere studie zul je vaak gebruik gaan maken van natuurkundige constanten en veel voorkomende natuurkundige formules. In deze eerste opdracht gaan we een `python` module maken die de waarden van deze constanten en een aantal standaard natuurkundige functies bevat. In de rest van het college zullen deze functies en constanten terugkomen. Om de opdrachten uit te voeren, moet je de `numpy` en `matplotlib` modules van `python` gebruiken. Deze modules zijn standaard beschikbaar op de computers van de Sterrewacht.

- (1) Maak een `python` module met de naam ‘astronomy’ die constantes bevat met de waarden (in SI-eenheden) van de volgende natuurkundige constanten: Newton’s Gravitatie constante (G), de constante van Planck (h) en lichtsnelheid (c). Je maakt een module door je broncode op te slaan als `modulenaam.py`; onze module wordt dus `astronomy`.
Tip: Zet als commentaar de eenheden erbij.
- (2) Verder is het vaak handig om SI eenheden te kunnen omrekenen in eenheden die in de sterrenkunde vaker gebruikt worden. Voeg daarom ook de volgende astronomische eenheden toe: AU, zonsmassa, straal van de zon, aardmassa, straal van de Aarde, massa van Jupiter en parsec. *Tip: Zet ook hier eenheden erbij.*
- (3) Het is ook handig om veel gebruikte natuurkundige formules in een module op te nemen. Schrijf een functie die de zwaartekracht F berekent voor twee gegeven massa’s m_1 en m_2 op een onderlinge afstand r .
- (4) Gebruik nu jouw zwaartekrachtfunctie om de versnelling te berekenen die wordt veroorzaakt door zwaartekracht die het internationale ruimtestation ISS voelt als gevolg van (a) de aarde, (b) de maan en (c) de zon. Je mag ervan uitgaan dat het ISS zich op een cirkelbaan op 407 km boven het aardoppervlak beweegt. Neem voor deze opgave aan dat het ISS zich op de lijn Aarde-Maan bevindt.
- (5) Op <http://nineplanets.org/data1.html> kun je enkele eigenschappen van objecten in het zonnestelsel vinden. Kopieer de gegevens van de *acht* planeten naar je module zodat je ze in het vervolg makkelijk kan gebruiken.
- (6) Plot de dichtheid van de planeten als functie van de straal. Label alle data punten met de naam van de planeet. Verklaar wat je ziet. Plot als horizontale lijnen de dichtheden van vloeibaar water, ijzer en zand.
- (7) Bereken de valversnelling op het oppervlak van elke planeet.
- (8) Zoals de andere rotsplaneten in ons zonnestelsel heeft de Aarde een kern, mantel en korst, zie figuur 1. Dit wordt in College 4 in detail behandeld. Hoewel de mantel niet direct te bestuderen is heeft men toch een goed idee waar deze uit bestaat: voornamelijk SiO_2 en MgO . De kern bestaat voornamelijk uit ijzer en nikkel. Uitgaande van een *transition zone* en *upper mantle* dichtheid van 3 g cm^{-3} , *lower mantle* dichtheid van 5 g cm^{-3} , en *outer core* dichtheid van 11.5 g cm^{-3} , bereken aan de hand van figuur 1 de dichtheid van de *inner core*. Vergelijk deze dichtheden met de dichtheden van alledaagse materialen als water, ijs, steen, en ijzer.

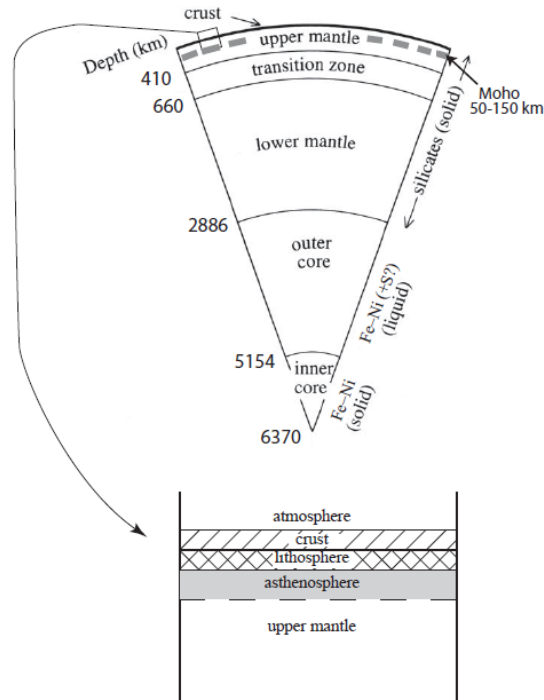


FIGURE 1. Schets van de binnenkant van de Aarde, *Putnis 1992. Introduction to Mineral Sciences.*

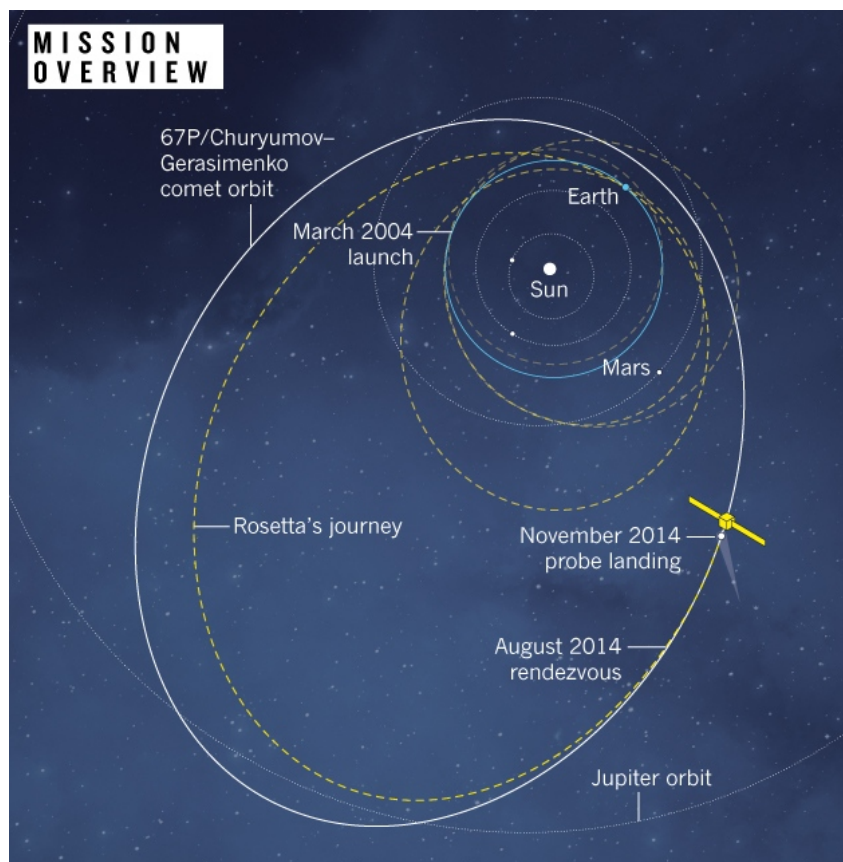


FIGURE 2. 67P trajet, *ESA*

OPDRACHT 2: SNELHEDEN

Bij interactie tussen meerdere objecten is de ontsnappingssnelheid van die objecten een belangrijke parameter voor de dynamica van het gehele systeem. Voor een object, dat beïnvloed wordt door een zwaartekrachtveld, is het de minimale snelheid die het nodig heeft om aan dat zwaartekrachtveld te ontsnappen.

- (1) Leid de formule voor de ontsnappingssnelheid af. Schrijf een `python` functie die de ontsnappingssnelheid voor een gegeven massa en een gegeven afstand tot die massa berekent.
- (2) In 2014 is het ESA gelukt om met een kleine sonde, Philae, op een komeet te landen. Philae zat aan boord van het ruimtevaartuig Rosetta dat in 2004 vanaf de Aarde werd gelanceerd en bereikte in 2014 de komeet 67P (zie figuur 2). Bereken wat de maximale snelheid is die deze komeet mogelijkwjs kan hebben, gegeven dat de komeet in een elliptische baan om de zon beweegt en dat de minimale afstand tot de zon 1.24 AU is. Bedenk hiervoor eerst op welk punt in de baan de snelheid van de komeet maximaal is.
- (3) Het aphelium van 67P ligt op 5.7 AU. Rosetta ontmoette de 67P op een afstand van 3.9 AU van de zon. Wat was dan de snelheid van de komeet ten tijde van de ontmoeting?
- (4) Na het bereiken van 67P liet Rosetta op een hoogte van 10 km boven het oppervlak de lander Philae zakken. Toen Philae het oppervlak bereikte, stuiterde deze terug omhoog.

We gaan de maximale snelheid berekenen waarbij Philae niet aan het zwaartekrachtsveld van de komeet ontsnapt bij het stuiteren. Ga er vanuit dat de snelheid waarmee Philae omhoog stuitert, de helft is van de snelheid waarmee Philae het oppervlak raakt. Bereken of de lander wegzweeft voor ontkoppelsnelheden tussen 0 km/s en 3 km/s. Neem aan dat de straal van de komeet 2 km is, dat deze 10^{13} kg weegt en dat Philae 100 kg weegt. Bereken en plot de stuitersnelheid als functie van de lanceersnelheid. Bereken ook de ontsnappingssnelheid vanaf het oppervlak van de komeet en voeg deze toe aan het plotje. Voor welke lanceersnelheden is Philae gravitationeel gebonden aan de komeet?

Tip: Gebruik energiebehoud en de omzetting van gravitationele energie in kinetische energie.

In College 6 zal er meer verteld worden over kometen.

- (5) Ruimtevaartorganisaties sturen regelmatig sondes naar interessante plekken in het zonnestelsel. De Voyager 1 en 2 zijn zelfs het zonnestelsel uitgestuurd. Kost het meer brandstof om een ruimtevaartuig het zonnestelsel uit te lanceren, dan om het in de Zon te sturen? Zijn interstellare missies dus makkelijker of moeilijker dan missies naar de binnenste delen van het zonnestelsel?

PYTHON TIPS

Modules.

In `python` laad je een module met:

```
import astronomy
```

Uiteindelijk is de bedoeling dat je deze constanten in `python` op de volgende manier kunt gebruiken:

```
import astronomy

# druk de constante G af
print astronomy.G
# zet snelheid op 0.2 maal de lichtsnelheid
velocity = 0.2 * astronomy.c
```

Tip: Je kunt de module naam ook afkorten met het `import .. as` commando. Als je een reeds geladen module wilt herladen (bijvoorbeeld omdat je de code hebt veranderd), gebruik dan het `reload` commando:

```
import astronomy as astr

# druk de constante G af
print astr.G
```

Lijsten.

Met `a=[2,4,6,8]` definieer je een `python` lijst. Op `python` lists kun je geen wiskundige operaties doen, zoals delen of machtsverheffen, op `numpy` arrays wel. In het voorbeeld hieronder geeft `python` een foutmelding als je een lijst door 2 probeert te delen. De `numpy` module voert het delen door 2 uit op elk element in het array. De `numpy` module is standaard beschikbaar op de computers van de Sterrewacht en op de website van het college staan links naar de documentatie erover. De module kan geladen worden, op dezelfde manier als boven beschreven is voor de `astronomy` module.

```
>>> a=[2,4,6,8]
>>> print a/2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for /: 'list' and 'int'
>>> import numpy
>>> a=numpy.array([2,4,6,8])
>>> print a/2
array([1, 2, 3, 4])
```

Selecties en criteria.

Het commando `(a > b)` laat zien bij welke items in array `a` de vergelijking waar is en waar deze niet waar is. Het commando `a[(a > b)]` geeft alle items van array `a` die groter zijn dan `b`. Als je een selectie wil maken die gelimiteerd is door 2 restricties, gebruik dan het `&` teken. Let op dat dit soort operaties alleen mogelijk zijn op `numpy` arrays.

```
>>> mijnarray=np.arange(1,5,1)
>>> print mijnarray>2
[False False  True  True]
>>> print mijnarray[mijnarray>2]
[3 4]
```

Plots.

Met `matplotlib` kun je data plotten met `python`. Grafieken (of plots) worden gebruikt om grote hoeveelheden data overzichtelijk weer te geven of om trends inzichtelijk te maken. Het goed kunnen plotten van data is dus onontbeerlijk in de rest van je studie. Hieronder staat een voorbeeld.

```
from matplotlib import pyplot as plt
import numpy

# Vul x array met nummers van 0 tot 100 in stappen van 10
x = numpy.arange(0, 100, 10)
# Vul y array met nummers van 0 tot 5 in stappen van -0.5
y = numpy.arange(5,0,-0.5)

# Maak de plot
plt.plot(x, y, 'ro', label='rode_punten')
plt.plot(x, y, label='blauwe_lijn')
# plot een lijn op y=2 tussen x=0 en x=1
plt.hlines(2,0,1)
plt.title("Title")
plt.xlabel("X-as")
plt.ylabel("Y-as")
plt.legend(loc='upper_left')
#opslaan naar png format
plt.savefig('figuur.png')
# Weergeven op scherm
plt.show()
# Het pop up scherm moet handmatig gesloten worden, dit zorgt
# er ook voor dat de figuur vergeten wordt.
# Wil je alleen opslaan, laat pyplot dan de figuur vergeten,
# anders is deze ook zichtbaar in de volgende plot
plt.clf()
```