Lärarhandledning - labb 6

Den första delen av denna labb handlar om trigonometri eftersom det behövs för att förstå sinusfunktionen som anpassas till datan för att få perioden.

Huvudsyftet med funktionen `pyplot.scatter()` är att man kan låta en tredje datamängd bestämma olika storlek eller färg på markörerna och på så sätt visualisera en tredje parameter i grafen. Den egenskapen används inte i denna labb, istället används funktionen istället för `pyplot.plot(x, y,'*')`.

`pyplot.grid()` behöver inga argument i vårt fall. Att anropa den utan argument slår på rutnät om det är av eller stänger av om det redan är på.

Genom att lägga till argumentet `label='namn'` sist i alla plot- eller scatter-kommandon och anropa `pyplot.legend()` innan `pyploy.show()` skapar man en legend som förklarar vilken data som är vilken. Genom att använda argumentet `loc='best'` i `legend()` hamnar den automatiskt på bästa plats.

`scipy.optimize.curve_fit` gör minsta-kvadrat-anpassning för en godtycklig funktion, see SciPy v1.4.1 Reference Guide[1] för detaljer.

Ett mer praktiskt sätt att skriva ut siffror i print är att använda metoden `.format`. Metoden anropas på en sträng med en (eller flera) platshållare i form av klammerparenteser. Innehållet i klammerparenteserna bestämmer hur datan i `.format` formateras. T.ex betyder `:f` att datan ska hanteras som en float. Utskriften blir då
`print("Period is {:f} days".format(period))`.
För mer detaljer se https://pyformat.info/.

Om utskriften av en anpassa kurvan ser konstig ut, se bilder nedan för exempel, beror det antagligen på olämpliga startgissningar. Gissningarna som används i labben är både rimliga och ger rätt svar.

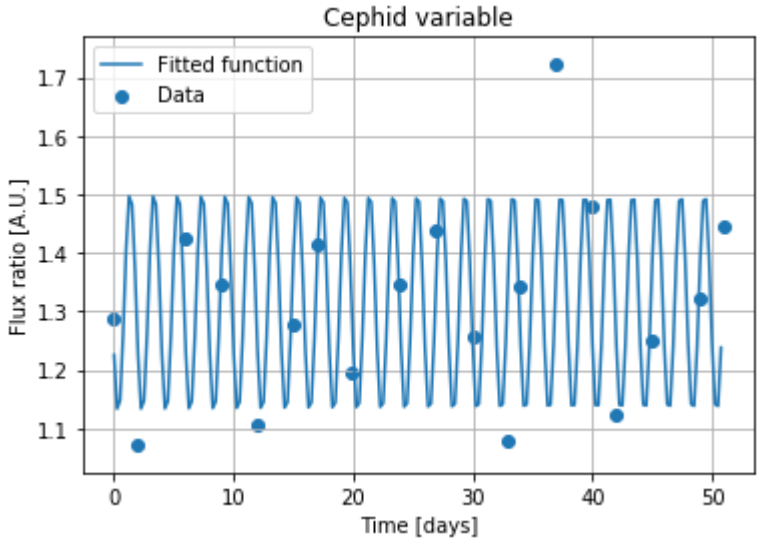
[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

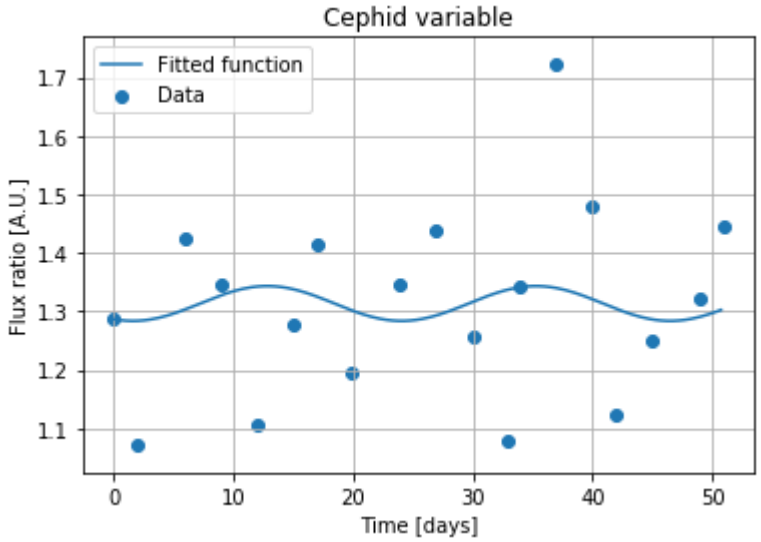
E-post: cansat@au.se
Telefon: 070-000 90 56

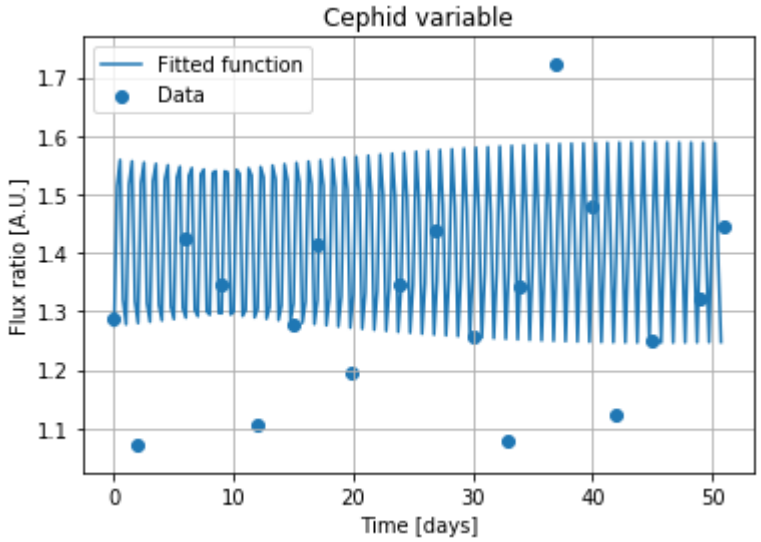
En funktion i Python skrivs med `def` först, sedan funktionens namn och en parentes med funktionens argument och avslutas med ett kolon. Koden i funktionen skrivs med ett indrag.

Funktionen i som ska anpassas skrivs som en normal funktion och ska svara med ekvationen vi vi anpassa. För att ge ett exempel, om vi skulle anpassa en rak linje med ekvationen y = k*x + m skulle vi skriva det som

```
def test_line(x, k, m):
    y = k * x + m
    return y
```


Cephid variable
Fitted function
Data
Flux ratio [A.U.]
Time [days]


Cephid variable
Fitted function
Data
Flux ratio [A.U.]
Time [days]



Ordlista

SciPy

Bibliotek till Python som innehåller funktioner för vetenskapligt arbete.

Funktioner som används

`pyplot.scatter(x-values, y-values)`

Skapar en spridningsgrad med den avgivna datan. Ytterligare kommandon kan anges för att ändra markörenas egenskaper.

`pyplot.grid()`

Slår på eller stänger av rutnät i bakgrunden av grafen.

`numpy.mean(array)`

Returnerar medelvärdet av *array*.

`max(array)`

Returnerar det största värdet i *array*.

`min(array)`

Returnerar det största värdet i *array*.

`userFunctions.sineFit(`*`x, y, guesses`*`)`

Funktion som anpassar en sinuskurva till datamängden som bestäms av `x` och `y`. `guesses` är en lista med 4 startgissningar i ordningen [Höjd, Amplitud, Svängningshastighet, Fasförskjutning]. Funktionen utgår från dessa när den söker efter en passande kurva. Funktionen svarar med 3 variabler, först en lista på 4 parametrar för den anpassade sinuskurvan, sedan en tätare lista på x-koordinater, och sist en lista med y-koordinater för den anpassade sinuskurvan som stämmer med x-koordinaterna som just gavs. Skriven av AU.

`scipy.optimize.curve_fit(`*`function, x-data, y-data,`* `p0=`*`list`*`)`

Returnerar parametrarna av funktionen *`function`* som bäst anpassar den till den angivna x- och y-datan. p0 är en lista på startvärden för alla parametrar. Används i `userFunctions.sineFit`.