

Simulering med Python Henfald fysik B.

Du kan f.eks. få Python med Anaconda

Med Anaconda:

Download Anaconda på anaconda.com

[Anaconda | The World's Most Popular Data Science Platform](https://anaconda.com)

Når du har installeret anaconda (det kan tage lidt tid) opdaterer du til version 2.2.0 eller nyere. Dernæst vælger du Spyder, her er python programmet.

Du åbner Spyder.

Vælg File, New file

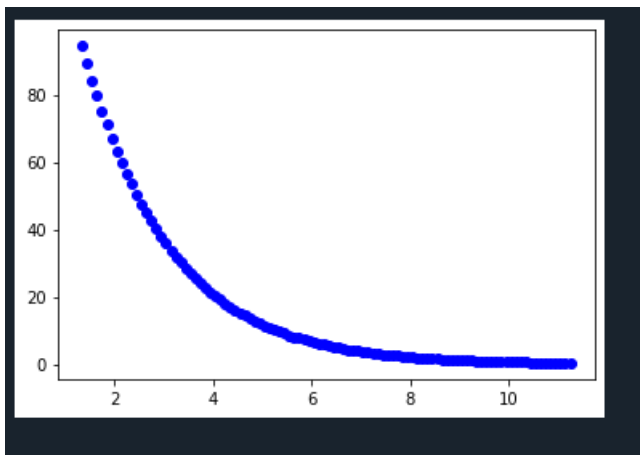
Skriv programmet ind.

Radioaktivt henfald

Vi ser på henfald af Te-131

Læs programmet. Hvor meget forstår du?

```
1  from matplotlib import pyplot as plt
2
3  n=100
4
5  t=0
6  dt=0.1
7
8  k=0.554518 #henfaldskonstanter er 0.554518 pr døgn
9
10
11 for i in range (100):
12     a=k*n
13     dn=a*dt
14     n=n-dn
15     t=t+dt
16     plt.plot(t,n, 'bo')
17
18
19 plt.show()
20
21
```



Dette program gør det samme, blot kun i 5 døgn:

```
1  from matplotlib import pyplot as plt
2
3  n=100
4
5  t=0
6  dt=0.1
7
8  k=0.554518 #henfaldskonstanter er 0.554518 pr døgn
9
10
11 while t<5:
12     a=k*n
13     dn=a*dt
14     n=n-dn
15     t=t+dt
16     plt.plot(t,n,'bo')
17
18
19 plt.show()
20
```

ADVARSEL, Hvis I benytter while, skal I være sikre på at betingelsen bliver opfyldt, ellers får I en uendelig løkke, der belaster computeren. Det sker ikke med en "For i en range (100)", da den kun gennemløber løkken 100 gange.

Skulle uheldet være ude, kan I stoppe programmet, ved at klikke på den røde firkant i nederste vindue for

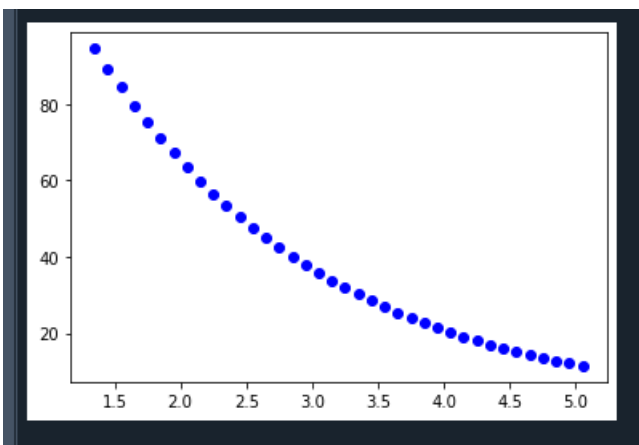


oven til højre.

Dette kan også gøres på en lidt nørdet måde, idet $\ln(2)$ skrives som $\text{math.log}(2,e)$.

Her skal vi først have importeret math biblioteket, og kaldt e fort math.e .

```
1  from matplotlib import pyplot as plt
2  import math
3
4  e=math.e
5
6  n=100
7
8  t=0
9  dt=0.1
10 t=1.25 #halveringstid for Te-131 er 1.25 døgn
11 k=math.log(2,e)/t
12
13
14 while t<5:
15     a=k*n
16     dn=a*dt
17     n=n-dn
18     t=t+dt
19     plt.plot(t,n,'bo')
20
21
22 plt.show()
23
```



Opgave 2: sæt navne på akserne, kør programmet i 10 døgn

Henfaldskæder

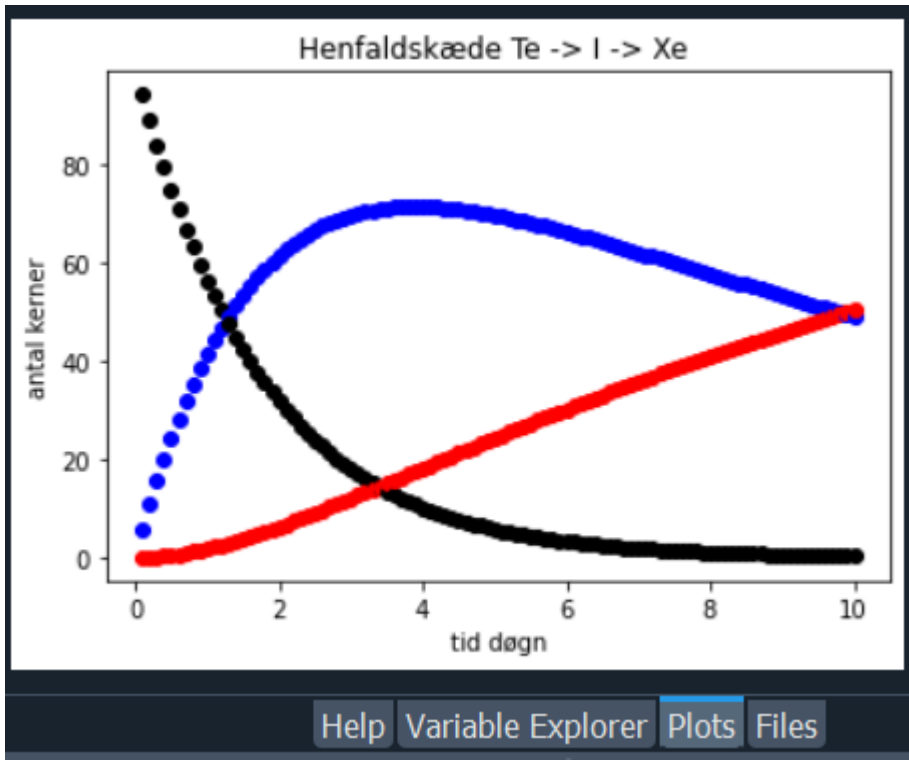
Vi vil se på henfaldskæden Tellur-131 -> Iod -131-> Xenon-131

```

1  from matplotlib import pyplot as plt
2  import sys
3
4  n1=100 #antal Te-131
5  n2=0   #antal I-131
6  n3=0   #antal Xe-131
7  t=0
8  dt=0.1
9  k1=0.554518 # henfaldskonstant for Te-131
10 k2=0.08664 #henfaldskonstant for I-131
11
12 for i in range (100):
13     a1=k1*n1
14     a2=k2*n2
15     dn1=a1*dt
16     dn2=a2*dt
17     n1=n1-dn1
18     n2=n2+dn1-dn2
19     n3=n3+dn2
20     t=t+dt
21     plt.plot(t,n2, 'bo')
22     plt.plot(t,n1, marker = 'o', color='k')
23     plt.plot(t,n3, marker = 'o', color='r')
24     plt.title("Henfaldskæde Te -> I -> Xe ")
25     plt.xlabel("tid døgn")
26     plt.ylabel("antal kerner")
27     if t>10:
28         sys.exit()
29 plt.show()
30

```

Her er resultatet af denne kørsel:



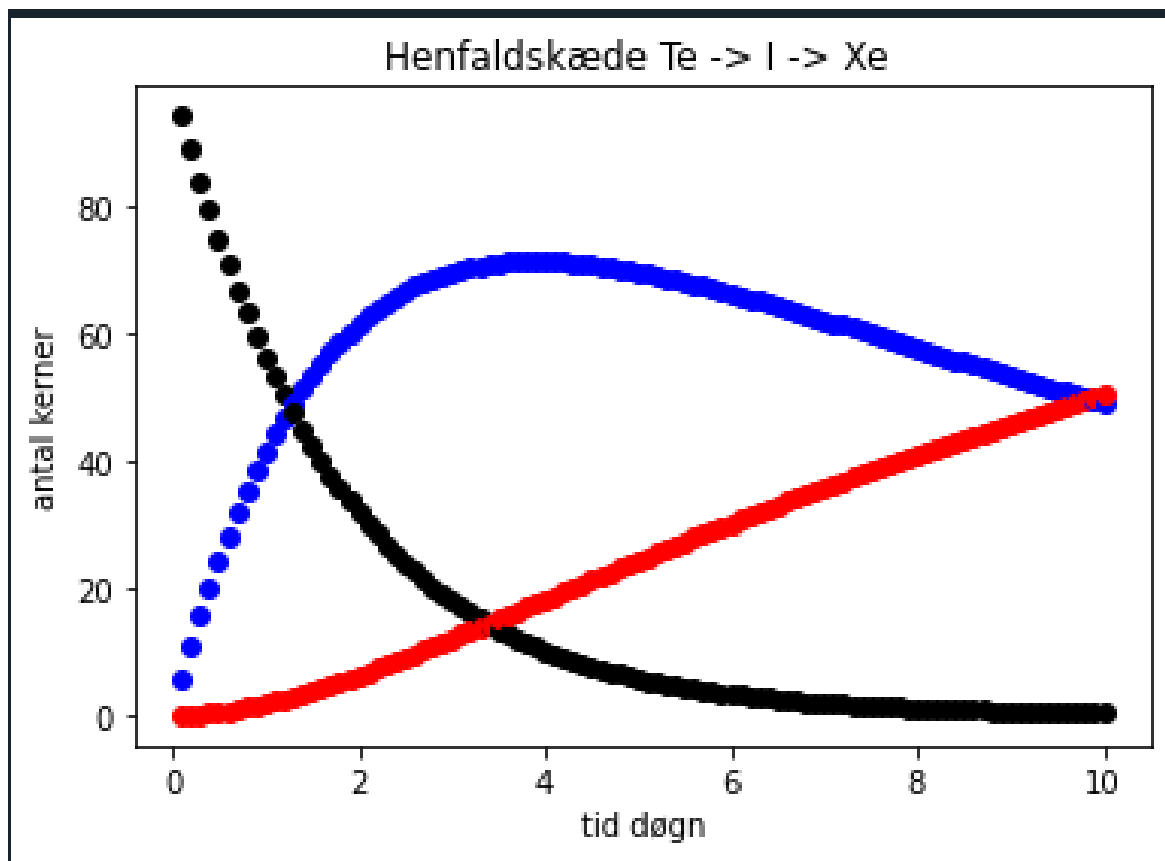
Eller mere nørdet:

```

1  from matplotlib import pyplot as plt
2  import math
3  import sys
4  e=math.e
5
6  n1=100 #antal Te-131
7  n2=0   #antal I-131
8  n3=0   #antal Xe-131
9  t=0
10 dt=0.1
11 t1=1.25 #halveringstid for Te-131 er 1.25 døgn
12 k1=math.log(2,e)/t1 #k1=ln(2)/t1
13 t2=8.0  #halveringstid for I-131 er 8 døgn
14 k2=math.log(2,e)/t2
15
16 for i in range (100):
17     a1=k1*n1
18     a2=k2*n2
19     dn1=a1*dt
20     dn2=a2*dt
21     n1+=-dn1
22     n2+=dn1-dn2
23     n3+=dn2
24     t+=dt
25     plt.plot(t,n2, 'bo')
26     plt.plot(t,n1, marker = 'o', color='k')
27     plt.plot(t,n3, marker = 'o', color='r')
28     plt.title("Henfaldskæde Te -> I -> Xe ")
29     plt.xlabel("tid døgn")
30     plt.ylabel("antal kerner")
31     if t>10:
32         sys.exit()
33 plt.show()

```

Her er et resultat af denne kørsel:



Opgave : Skriv programmet og kør programmet.

Prøv med andre henfaldskæder.

nuclide	half-life (a=year)	product of decay
^{232}Th	1.405×10^{10} a	^{228}Ra
^{228}Ra	5.75 a	^{228}Ac
^{228}Ac	6.25 h	^{228}Th
^{228}Th	1.9116 a	^{224}Ra
^{224}Ra	3.6319 d	^{220}Rn
^{220}Rn	55.6 s	^{216}Po
^{216}Po	0.145 s	^{212}Pb
^{212}Pb	10.64 h	^{212}Bi
^{212}Bi	60.55 min	^{212}Po ^{208}Tl
^{212}Po	299 ns	^{208}Pb
^{208}Tl	3.053 min	^{208}Pb
^{208}Pb	.	.

Inspiration hentet fra

Ole Bakander
Ole Helweg-Larsen
Lasse Storr-Hansen

FPro3

[FPro3_en kort introduktion.pdf \(fys.dk\)](http://www.fys.dk/fpro3/en/kort%20introduktion.pdf)

www.fys.dk/fpro3/Tekster/FPro3_en%20kort%20introduktion.pdf

EH 19-01-2023