

DynRein-modellen: Ein dynamisk modell for radioaktivt cesium i reinsdyr

Referanse

Ytre-Eide MA, Skuterud L. DynRein-modellen: Ein dynamisk modell for radioaktivt cesium i reinsdyr. Teknisk dokument nr. 22. Østerås: Direktoratet for strålevern og atomtryggleik, 2021.

Publisert
Sider

22.03.2022
17

DSA,
Postboks 55,
No-1332 Østerås,
Norge.

Emneord

Cesium-137. Reinsdyr. Modellering. Python.

Telefon
Faks
Epost

67 16 25 00
67 14 74 07
dsa@dsa.no
dsa.no

Resymé

Dette dokumentet presenterer koden for DynRein-modellen med forklaringar.

ISSN 2387-5240

Reference

Ytre-Eide MA, Skuterud L. The DynRein model: A dynamic model for radioactive caesium in reindeer. Technical Document no. 22. Østerås: Norwegian Radiation and Nuclear Safety Authority, 2021. Language: Norwegian.

Key words

Caesium-137. Reindeer. Modelling. Python.

Abstract

This document presents the code for the DynRein model.

Prosjektleder: Lavrans Skuterud

Godkjent:



Astrid Liland, avdelingsdirektør, avdeling beredskap

DynRein-modellen: Ein dynamisk modell for radioaktivt cesium i reinsdyr

Martin Album Ytre-Eide

Lavrans Skuterud

fra Norsk institutt for luftforskning (NILU)
Kjeller,
Norway

fra Direktoratet for strålevern
og atomtryggleik (DSA)
Østerås,
Norway

Innhold

1	Innleiing	5
2	DynRein-modellen	6
3	Programkode og prosedyre for køying av modellen	7

1 Innleiing

Gjennom CERAD-samarbeidet¹ fekk DSA støtte til prosjektet Dynamic reindeer for å utvikle ein dynamisk modell for langtidsutvikling i konsentrasjonar av radioaktivt cesium i reinsdyr. Prosjektet fekk god hjelp av Vågå tamrein A/S (<https://vagarein.no/>) der mykje utviklingsarbeid har vore gjort gjennom alle åra etter Tsjernobyl-ulykka i 1986, og der dei allereie rundt 2010 tok i bruk GPS-klavar på dyr i flokken. Vi retter ei stor takk til Vågå tamreinlag for mykje hjelp og triveleg samarbeid gjennom mange år!

For å kunne utvikle ein funksjonell modell for langtidsutvikling av radioaktivt cesium i reinsdyr trong vi diverse bakgrunnsinformasjon frå beiteområdet som har blitt publisert i to artiklar:

1. Detaljert kartlegging av forureininga i beiteområdet er publisert i Thørring m.fl. (2019): «Airborne radiometric survey of a Chernobyl-contaminated mountain area in Norway – using ground-level measurements for validation»
2. Informasjon om korleis nivåa av radioaktivt cesium har endra seg i beiteplanter sidan 1986 er publisert i Skuterud og Thørring (2021): «Caesium-137 in mountain flora with emphasis on reindeer's diet – Spatial and temporal trends»

Det endelege produktet – DynRein-modellen – blir publisert i ein tredje fagartikkel (Skuterud m.fl. (under utarbeiding): «Modelling the dynamic behaviour of radiocaesium in reindeer grazing Chernobyl contaminated pastures»). I den artikkelen blir fagleg grunnlag og vurderingane bak modellen presentert. Dette tekniske dokumentet presenterer programkoden til modellen med diverse forklaring.

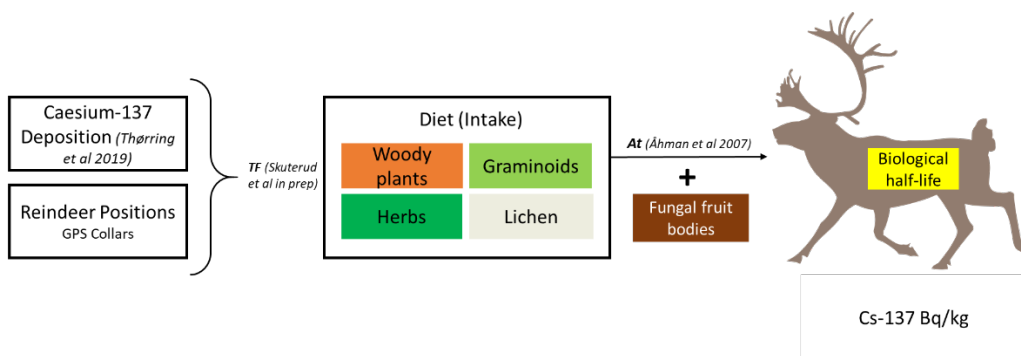
Arbeidet med modellutvikling og kart- og GPS-data frå Vågå tamreinlag sitt beiteområde gjennom mange år involverte fleire store datasett. Dette var noko av grunnen til at DynRein-modellen vart utvikla i programmeringsspråket Python. Koden er prega av den vart laga i samband med eit forskingsprosjekt, og vart ikkje utvikla for å vere ein operativ kode. For meir generell og framtidig bruk har det vore meininga at DynRein-modellen skulle overførast til ein annan plattform (for eksempel ECOLEGO, som også blir brukt i andre samanhengar ved DSA). Denne tekniske rapporten dokumenterer programkoden slik at overføring kan gjerast på eit seinare tidspunkt.

¹ Centre for Environmental Radioactivity (CERAD) er eit senter for framifrå forskning støtta av Forskningsrådet (prosjektnr. 223268/F50)

2 DynRein-modellen

Utgangspunktet i DynRein-modellen er at reinsdyra sin posisjon gir ein deposisjonsverdi (Bq/m^2). Denne deposisjonsverdien blir kombinert med data om overføring av cesium frå jord til dei ulike vegetasjonsgruppene som reinen beiter gjennom året, og kunnskap om reinen sin diett, slik at eit dagleg inntak (Bq/dag) kan reknast ut, sjå figur 1. Deretter blir dette inntaket mata inn i ein gjennomsnitt vaksen rein (70 kg), der ein viss del blir absorbert frå mage-tarm-kanalen. Reinsdyr skil så ut cesium igjen med ei biologisk halveringstid som er avhengig av sesong. Saman med stor sesongforskjell i diett gir sesongforskjellen i biologisk halveringstid at konsentrasjonane av cesium i reinsdyr generelt vil vere mange gonger høgare på vinteren enn på sommaren i fleire år etter eit nedfall.

For modellering av cesium-konsentrasjon i rein i fleire år etter eit nedfall er det viktig å inkludere at rein kan ete mykje sopp om hausten. For dei fyrste åra etter eit stort nytt nedfall, vil ikkje bidraget frå sopp vere viktig. I DynRein blir bidrag frå sopp lagt til konsentrasjonen som skuldast inntak av anna vegetasjon (dette blir ikkje estimert frå inntak, fordi vi t.d. ikkje veit kor mykje sopp reinsdyr et).



Figur 1 Skisse av DynRein-modellen (frå Skuterud m.fl., (under utarbeiding)).

Matematisk kan modellen for cesium-konsentrasjon (C_m ; Bq/kg) i reinsdyr skrivast:

$$C_m = \frac{Q_t}{0.47M_b} = \left(\frac{1}{0.47M_b} \sum_i D_i \times a_s \times TF_i \times e^{-\ln 2/T_{eco,i}} \times A_t + F_j \right) \times e^{-\ln 2/T_{phys}} + \frac{Q_{t-1}}{0.47M_b} \times e^{-\ln 2/T_{bio}}$$

der:

- Q_t er totalt innhald av ^{137}Cs i reinsdyr ved tid t (og Q_{t-1} ved tid $t-1$)
- M_b er kroppsvekta til rein, som er sett til 70 kg (vaksen rein). Muskelmasse (kjøtt) er 47% av kroppsvekta (frå Åhman (2007): Kroppsvekt utan mage-tarm-innhald er 83 % av livvekt, og muskel er 57 % av kroppsvekt utan mage-tarm-innhald, dvs. muskel er 47 % av livvekt)
- D_i er konsum (kg/dag) av vegetasjonsgruppe i (dietten er delt i grasaktige og treaktige planter, urter og lav)
- a_s er deposisjonsnivået av cesium i jorda (Bq/m^2)
- $TF_i(t)$ er overføringsfaktor ($10^{-3} \text{ m}^2/\text{kg}$) til vegetasjonsgruppe i ved år t
- $T_{eco,i}$ er økologisk halveringstid (år) i vegetasjonsgruppe i (ikkje inkludert fysisk halveringstid)
- A_t er absorpsjon av ^{137}Cs i mage-tarm-kanalen (satt til 61 %)
- F_j er bidraget frå sopp til auke i cesium-konsentrasjon i reinkjøtt om hausten (Bq/kg)
- T_{phys} er fysisk halveringstid for ^{137}Cs
- T_{bio} er biologisk halveringstid for cesium i reinsdyr (varierer frå 7 dagar i juni og juli til 30 dagar in desember-mars)

3 Programkode og prosedyre for køyring av modellen

Programmet er utvikla i Python og skrivne i Jupyter Notebook (filekstensjon ipynb). Koden er gitt på dei neste sidene. I starten er det vist kva for Python-bibliotek som trengs for å køyre modellen. Slik den er vist, inneheld koden ein del moglegheiter for sjekk av importerte data og mellomrekningar som kan ryddast bort (slettast).

For å køyre modellen for ein annan stad eller etter eit nytt nedfall må følgjande gjerast:

1. Det må lagast ein tabell med Cs-deposisjon for kvar dag gjennom året, dvs. antekne eller målte deposisjonsverdiar for dei ulike beiteområda reinsdyra brukar gjennom *heile* året. Formatet er vist i følgjande tabell (Cs-verdiar skal vere i kBq/m²). Tabellen må lagrast i csv-format (dvs. punktum må brukast for ev. desimaltal). I eksempelet heiter csv-fila Cs_dep.csv – dette må tilpassast det namnet som blir brukt.

dato	Cs
01.01.1986	58
02.01.1986	58
03.01.1986	59
04.01.1986	60
....	
....	
28.12.1986	55
29.12.1986	56
30.12.1986	56
31.12.1986	57

2. Året (årstalet) for ulykka/nedfallet må leggjast inn, og ein må bestemme kor mange år framover ein vil køyre modellen for. Det er fleire stader i koden at dette må gjerast (også i samband med soppbidrag).
3. For å få riktige modellresultat det året nedfallet skjer, må ein korrigere for at nedfallet neppe skjer den 1. januar. Det gjer ein ved å sette nokre omtrentlege verdiar frå 1. januar til nedfallet skjer, sjå eksempel i koden.
4. Dersom ein skal ta omsyn til sopp, må ein legge inn kategoriane for soppår (lite, middels, mykje). Dersom sopp ikkje blir vurdert som viktig, kan alle år settast til lite, eller talverdiane for lite, middels og mykje kan settast til 0.
5. På slutten av koden må ein også legge inn ein startverdi for Cs-konsentrasjonen i reinsdyra før nedfallet kjem. Den bør vere omtrent reell konsentrasjon. I eksempelet i denne rapporten er det brukt 700 Bq/kg.

Modellresultatet blir presentert i ein enkel graf, og blir også skrivne til csv-fil. Csv-fila har to kolonnar: Dato og Cs-konsentrasjon i Bq/kg. I eksempelet på dei neste sidene heiter resultatfila Cs_rein.csv – dette kan endrast.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import dates
from dateutil import rrule
from dateutil.relativedelta import relativedelta
import datetime
```

Start med å importere tabell med datoer (alle dager gjennom året) og Cs-deposisjon der dyra beiter gjennom eit år

Desse verdiane blir brukt til å rekne ut deposisjonen for heile perioden modellen skal køyrast

In [2]:

```
# Tabellen må vere csv-format (med punktum som desimalteikn), og Cs-verdiar skal vere i kBq/m2
df=pd.read_csv('Cs_dep.csv') # Filnamnet kan endrast
df['dato']=pd.to_datetime(df['dato']) # Gjer kolonna til datetime
df # Skriv tabellen for å sjekke at import har blitt riktig
```

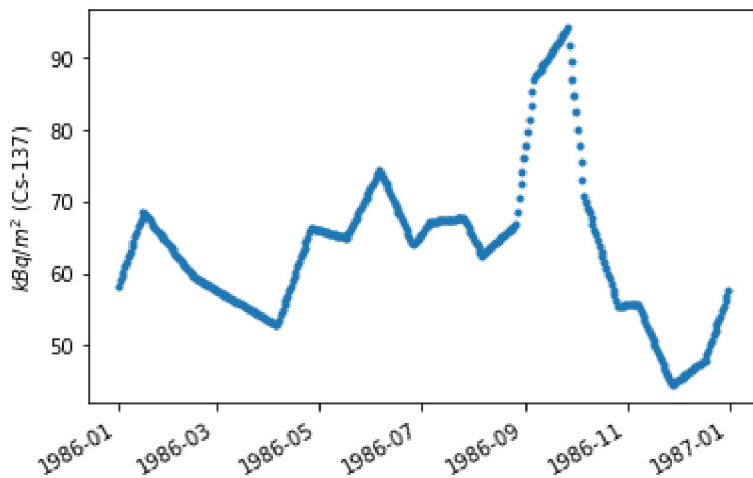
Out[2]:

	dato	Cs
0	1986-01-01	58.217528
1	1986-01-02	58.901883
2	1986-01-03	59.586238
3	1986-01-04	60.270592
4	1986-01-05	60.954947
...
360	1986-12-27	54.795753
361	1986-12-28	55.480108
362	1986-12-29	56.164463
363	1986-12-30	56.848818
364	1986-12-31	57.533173

365 rows × 2 columns

In [3]:

```
# Plott for sjekke at import av data har blitt riktig
plt.plot(df.dato,df.Cs, '.')
plt.ylabel("$kBq/m^2$ (Cs-137)")
plt.gcf().autofmt_xdate()
plt.show()
```



Legg inn start- og sluttdato for ønska modellperiode

In [4]:

```
# Må Lage dep-verdier for heile perioden som skal modellerast:
# NB: Nedfallsår må spesifiserast i for-løkke
# NB: Det må også leggest inn årstal i samband med sopp lenger ned
start=datetime.date(1986, 1, 1)
end=datetime.date(2018,12,31)
periode=[]
val=[]
periode= periode + [tid.date() for tid in rrule.rrule(rrule.DAILY, dtstart=start, until
=end)]
for i in range(0,len(periode)):
    aar=periode[i].year -1986 # Reknar ut år etter nedfall - må skrive inn nytt nedfall
sår
    if (periode[i].month==2) & (periode[i].day==29): # Tek omsyn til skotår: Sett Cs-de
posisjonsverdien for 29.feb. Lik 28.feb.
        cs_year=val[-1]
    else:
        cs_0=df[(df.dato.dt.month==periode[i].month) & (df.dato.dt.day==periode[i].day
)].Cs.values[0]
        cs_year=cs_0*np.exp(-np.log(2)/30.17*aar)
    val=np.append(val,cs_year) # Array med dep-verdier for hele perioden
```

In [5]:

```
df2=pd.DataFrame(columns=['dato', 'Cs'])
df2['dato']=pd.to_datetime(periode)
df2['Cs']=val
```

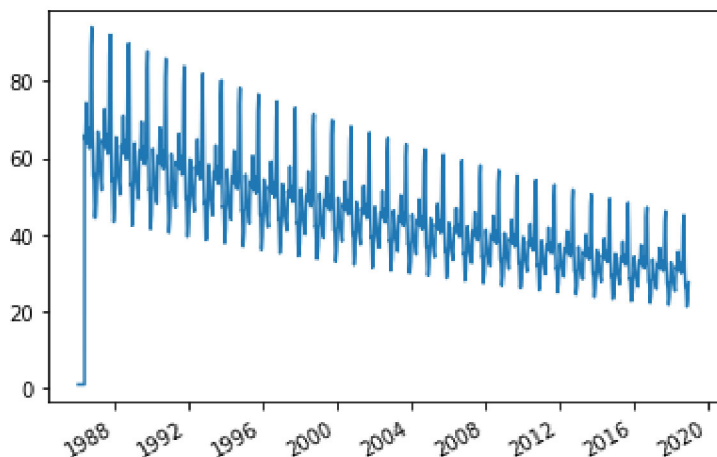
Legg inn omtrentlige verdier for eksisterende forureining i området fram til det nye nedfallet (i ulykkesåret)

In [6]:

```
# For at modellering av konsentrasjonar i nedfallsåret skal bli riktig, og
# sidan nedfallet neppe skjer 1. januar, må det settes inn ei rekke med
# "bakgrunnsverdier" fram til nedfallet skjer. I eksempelet her er Cs-verdiane
# satt til 1 kBq/m2 fram til slutten av april (120 dagar inn i 1986): val[0:120]=1
val[0:120]=1
df2['Cs']=val
```

In [7]:

```
# Plott for sjekke at utrekning av deposisjonsverdier framover i heile
# den ønska tidsperioden har blitt riktig
plt.plot(df2.dato,df2.Cs)
plt.gcf().autofmt_xdate()
plt.show()
```



Neste steg er utrekning av inntak m.m. Her er det ingenting som må endrast

In [8]:

```
# Verdier for ulike parametrar frå Skuterud m.fl.
intake=np.array([1.5,1.5,1.5,1.5,2.5,2.5,3.3,3.3,2.5,2.5,2.5,1.5,1.5]) # kg per dag for
kvar måned
At=0.61 # Absorpsjon i mage-tarm-kanalen
T_bio=[30,30,30,30,15,7,7,7,10,20,20,20,25] # Biologisk halveringstid i dyra og korleis
den endrar seg gjennom året
months=range(1,14) # Ekstra måned lagt til pga. deling av dietten i november
# Diett gjennom året (prosentdel av intake):
diet=pd.DataFrame(data={'Lichen': [0.866,0.866,0.80,0.7333,0.60,0.1154,0.0909,0.1515,0.
3077,0.60,0.6923,0.6875,0.80],
                        'Grass': [0.066,0.066,0.10,0.1333,0.10,0.3932,0.404,0.
579,0.5518,0.2526,0.1868,0.1897,0.1176],
                        'Woody_plants': [0.067,0.067,0.10,0.1333,0.30,0.1966,
0.101,0.1198,0.1405,0.1474,0.1209,0.1228,0.0824],
                        'Herbs': [0,0,0,0,0,0.2949,0.404,0.1497,0,0,0,0,0]})
```

In [9]:

```
# Overføring frå jord til vegetasjon, korrigert med effektive halveringstider
# Talverdiar for TF og Teff frå tabell 1 i Skuterud m.fl.)
# NB: Nedgangen i lav skjer mot ein konstantverdi (0.0217). Dersom halveringstider skal
# endrast
# bør også denne konstanten endrast - den er gitt i eksponensiell likning lenger ned.
TFs=pd.DataFrame(data={'vegetation': ["Lichen", "Grass", "Woody_plants", "Herbs"], 'TF':
                        [0.762,0.0431,0.0218,0.0797], 'T_eff': [2.87,15.5,22.2,13.8
]}, columns=['vegetation', 'TF', 'T_eff'])
```

In [10]:

```
# Utvider dataframe og sjekker at diett-, halveringstid, og totalinntak gjennom året er
rett
diet['T_bio']=T_bio
diet['intake']=intake
diet
```

Out[10]:

	Lichen	Grass	Woody_plants	Herbs	T_bio	intake
0	0.8660	0.0660	0.0670	0.0000	30	1.5
1	0.8660	0.0660	0.0670	0.0000	30	1.5
2	0.8000	0.1000	0.1000	0.0000	30	1.5
3	0.7333	0.1333	0.1333	0.0000	30	1.5
4	0.6000	0.1000	0.3000	0.0000	15	2.5
5	0.1154	0.3932	0.1966	0.2949	7	2.5
6	0.0909	0.4040	0.1010	0.4040	7	3.3
7	0.1515	0.5790	0.1198	0.1497	7	3.3
8	0.3077	0.5518	0.1405	0.0000	10	2.5
9	0.6000	0.2526	0.1474	0.0000	20	2.5
10	0.6923	0.1868	0.1209	0.0000	20	2.5
11	0.6875	0.1897	0.1228	0.0000	20	1.5
12	0.8000	0.1176	0.0824	0.0000	25	1.5

In [11]:

```
import datetime
startyear = 1986 # Må sette inn nytt nedfallsår
startmonth = 1
endyear = 1986 # Må sette inn nytt nedfallsår
endmonth = 11
alle_datoer=[]
for i in range(0,35): # Talet (35) må stemme med lengda på alle_datoer
    datoer=[datetime.date(m//12, m%12+1, 1) for m in range((startyear+i)*12+startmonth-
1, (endyear+i)*12+endmonth)]
    datoer.append(datetime.date (startyear+i, 11, 15))
    datoer.append(datetime.date (startyear+i, 12, 1))
    alle_datoer.extend(datoer)
```

In [12]:

```
# Multipliserer opp diet for alle år som skal modellerast.
total_diet=pd.concat([diet]*35, ignore_index=True) # Talet (35) må stemme med lengda på
alle_datoer
total_diet['dato']=alle_datoer
total_diet['dato'] = pd.to_datetime(total_diet['dato'])
total_diet.index = total_diet['dato']
del total_diet['dato']
```

In [13]:

```
df_interpol=total_diet.resample('D').mean()
df_interpol['Lichen'] = df_interpol['Lichen'].interpolate()
df_interpol['Grass'] = df_interpol['Grass'].interpolate()
df_interpol['Woody_plants'] = df_interpol['Woody_plants'].interpolate()
df_interpol['Herbs'] = df_interpol['Herbs'].interpolate()
df_interpol['T_bio'] = df_interpol['T_bio'].interpolate()
df_interpol['intake'] = df_interpol['intake'].interpolate()
df_interpol.head(4) # Sjekker at datoar og diett m.m. er riktig
```

Out[13]:

	Lichen	Grass	Woody_plants	Herbs	T_bio	intake
dato						
1986-01-01	0.866	0.066	0.067	0.0	30.0	1.5
1986-01-02	0.866	0.066	0.067	0.0	30.0	1.5
1986-01-03	0.866	0.066	0.067	0.0	30.0	1.5
1986-01-04	0.866	0.066	0.067	0.0	30.0	1.5

Om modellresultata skal ta omsyn til effekten av soppinntak, må det skrivast inn kategoriar (lite, middels eller mye) her:

In [14]:

```
# Det må vere kategoriar for Like mange år som det modellen skal køyrast.
# Verdiane (200, 2000, 5000) nedanfor er henta frå Skuterud m.fl.,
# og desse verdiane kan settast til 0 dersom modellen skal køyrast utan sopp
# (ev. kan "Lite" brukast som kategori for alle år)
# NB: I neste celle må det også spesifiserast årstal for nedfall osv.
soppAr=["Lite" ,"Lite","Mye","Lite","Lite","Middels","Lite","Middels","Middels","Lite",
"Lite","Middels"
,"Middels","Lite","Mye","Mye","Lite","Lite","Lite","Middels","Mye","Lite","Midd
els","Mye"
,"Middels","Middels","Lite","Lite","Mye","Lite","Lite","Lite","Middels","Lite",
"Lite"]
df_sopp=pd.DataFrame(columns=['forekomst','value'])
df_sopp['forekomst']=soppAr
df_sopp['value']=soppAr
df_sopp.replace({'value': {'Lite': 200, 'Middels': 2000, 'Mye': 5000}}, inplace=True)
```

In [15]:

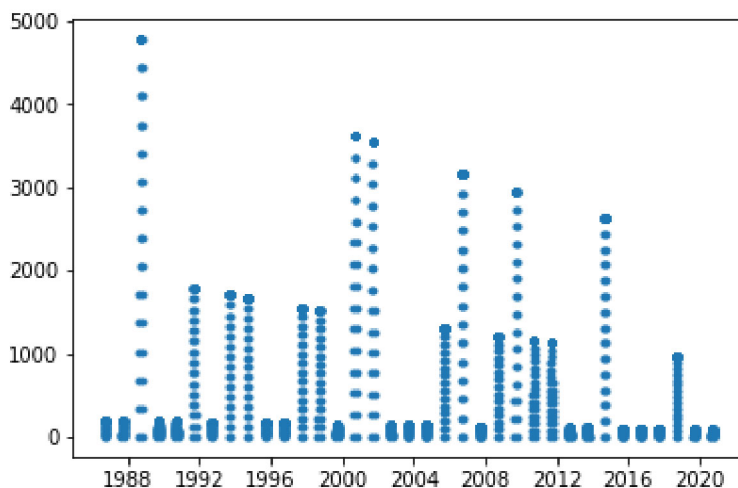
```
# Modellerer effekt av soppinntak frå 15. aug. til 15. okt. kvart år
# NB: Nedfallsåret må presiserast
from datetime import datetime
numb=df_sopp['value'].values
starten=datetime(1986, 8, 15) # Må sette inn nytt nedfallsår
enden=datetime(1986,10,15) # Må sette inn nytt nedfallsår
periode=[]
skalering=[]

for i in range(0,len(numb)): # Lager liste for åra med tilhøyrande soppårskalering
    start= starten +relativedelta(years=i)
    end = enden +relativedelta(years=i)
    periode= periode + [tid.date() for tid in rrule.rrule(rrule.DAILY, dtstart=start, u
ntil=end)]
    if numb[i]==1.0:
        skalering.extend([1]*62) # varigheit av soppsesong (dagar)
    else:
        tmp=[x for x in np.linspace(start=1, stop=numb[i], num=15)]+ [numb[i]]*32 + [x
for x in np.linspace(start=numb[i],stop=1, num=15)]
        skalering.extend(tmp)

sopp=pd.DataFrame(periode,columns=['dato'])
sopp['dato']=pd.to_datetime(sopp['dato'])
sopp['years']=sopp['dato'].dt.year -1986 # Må sette inn nytt nedfallsår
sopp['scale']=skalering*np.exp(-np.log(2)/30.17*(sopp.years))
```

In [16]:

```
# Plott for sjekke at utrekning av soppbidrag for heile den ønska
# tidsperioden har blitt riktig
plt.plot(sopp.dato,sopp.scale, '.')
plt.show()
```



Til slutt må det veljast ein inngangsverdi for Cs-konsentrasjon i dyra før modellen kan køyrast:

In [17]:

```
# Her var valt 700 Bq/kg (per 1. jan. 1986)
# (verdien må vere rundt forventa konsentrasjon utan nytt nedfall)
def start_rein2(start_value=700):
    '''Reknar ut initielt totalinnhald (Q) av Cs i rein basert på valt konsentrasjon'''
# Funksjonsforklaring
vekt=70 # kg
Me=0.83*vekt # Me er kroppsvekt utan mage-tarm-innhald
start= Me* start_value/1000 *0.57 # Q_0
return start
```

In [18]:

```
def rein_cs(rein,At,df_interpol,TFs,t=0):
    Q=[start_rein2()]
    for index, row in rein.iterrows():
        t+=1
        cesium= row['Cs']
        dato= row['dato']
        aar= dato.year-1986 # Må sette inn nytt nedfallsår
        intak=df_interpol.loc[dato]

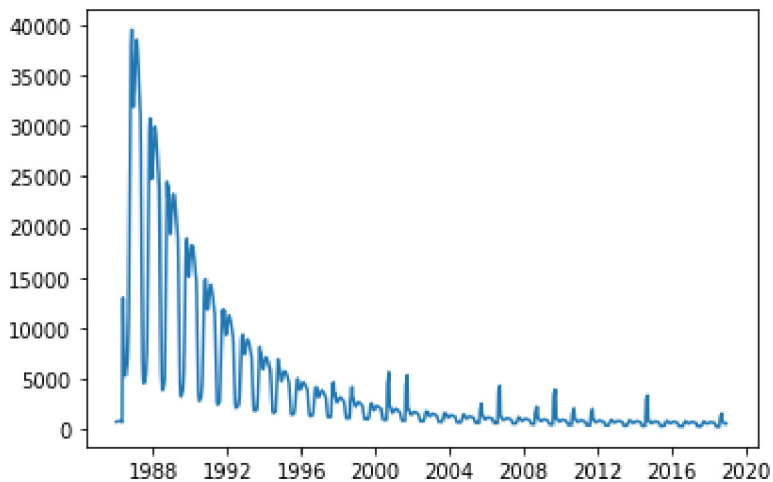
        i_m_Li= intak['intake']*((TFs['TF'][0]*np.exp(-(np.log(2)/TFs['T_eff'][0])*aar)
+0.0217)*intak['Lichen'])
        # Talverdien 0.0217 er konstantleddet i likninga for nedgangen i Lav. Bør endra
        # halveringstider blir endra.
        i_m_GrWpHe=intak['intake']*(TFs['TF'][1]*np.exp(-(np.log(2)/TFs['T_eff'][1])*aa
r) * intak['Grass']
                + TFs['TF'][2]*np.exp(-(np.log(2)/TFs['T_eff'][2])*aar) * intak['Wo
ody_plants']
                + TFs['TF'][3]*np.exp(-(np.log(2)/TFs['T_eff'][3])*aar) * intak['He
rbs'])
        sum_intak= (i_m_Li + i_m_GrWpHe)*cesium
        k=np.log(2)/intak['T_bio']
        Q.append((sum_intak*At + Q[t-1]*np.exp(-k)))
    del Q[0]
    return Q
```

In [19]:

```
mergedStuff = pd.merge(df2,sopp, on=['dato'], how='left')
mergedStuff.fillna(1,inplace=True)
df3=mergedStuff
vekt=70 # Kroppsvekt 70 kg
Me=0.83*vekt # Me er kroppsvekt utan mage-tarm-innhald
Q=rein_cs(df3,At,df_interpol,TFs)
dictionary=dict(zip(df3['dato'],[x/(Me*0.57)*1000 for x in Q]+df3['scale']))
```

In [20]:

```
Cs_rein = pd.DataFrame(data={1:dictionary})
plt.plot(Cs_rein) # Plotter resultatet (Bq/kg i dyra) for heile modellperioden:
Cs_rein.reset_index(inplace=True)
Cs_rein.columns=['Dato', 'Bq/kg']
# Cs_rein # Skriv ut datoane med modellresultat
```



Skriv resultatet (Bq/kg i dyra) for heile modellperioden til csv-fil:

In [21]:

```
Cs_rein.to_csv('Cs_rein.csv',index=False) # Filnamnet (mellom apostrofane) kan endast
```

In [22]:

```
# Om det er interessant å separere soppbidraget (tillegget Bq/kg i dyra frå sopp) for h
eile
# modellperioden, kan det skrivast til csv-fil her:
# sopp.to_csv('soppbidrag.csv')
```

4 Referansar

Skuterud L, Hevrøy T, Thørring H, Ytre-Eide M. Modelling the dynamic behaviour of radiocaesium in reindeer grazing Chernobyl contaminated pastures. Artikkel under utarbeiding.

Skuterud, L., Thørring H (2021). Caesium-137 in mountain flora with emphasis on reindeer's diet – Spatial and temporal trends. *Journal of Environmental Radioactivity* 231: 106551.

Thørring H, V. C. Baranwal VC, Ytre-Eide MA, Rønning JS, Mauring A, Stampolidis A, Drefvelin J, Watson RJ, Skuterud L (2019). Airborne radiometric survey of a Chernobyl-contaminated mountain area in Norway – using ground-level measurements for validation. *Journal of Environmental Radioactivity* 208-209: 106004.

Åhman, B. (2007). Modelling radiocaesium transfer and long-term changes in reindeer. *Journal of Environmental Radioactivity* 98(1-2): 153-165.

