Oppgavesamling

IN-KJM1900

Andreas Haraldsrud

Kjemisk institutt
Universitetet i Oslo
Høst 2020

Innhold

Veiledning til oppgavene

Oppgavene i denne oppgavesamlingen er ment å vise deg hvordan programmering kan være nyttig i kjemi og matematikk. Oppgavene i del 1 er supplerende, frivillige oppgaver til ukesobligene i fellesdelen før midtveiseksamen. Det er lurt å gjøre noen av disse, i hvert fall de stjernemerkede oppgavene. De tre oppgavene i del 2 (oblig 7, 8 og 9) er obligatoriske, og du må få godkjent hver oppgave for å kunne gå opp til eksamen. For å få godkjent oppgavene må du følge kriteriene nedenfor:

Kriterier

1. Programmene bør ha en hensiktsmessig mengde kommentarer som gjør koden enkel å lese, og som viser at at du forstår det du selv har laget.
2. Hver oppgave skal leveres som en rapport i form av en Jupyter Notebook-fil som har god struktur og inneholder:
   (a) hensiktsmessige overskrifter.
   (b) kode med kommentarer.
   (c) drøfting av resultater.
3. Algoritmene og de numeriske metodene bør forklares i Markdown.
4. Resultatene bør komme tydelig fram, og output må kommenteres.
5. Alle plott skal ha aksetitler og merkelapper (labels) der det er relevant.
6. Alle numeriske metoder må implementeres som Python-funksjoner. De bør også inneholde:
   (a) relevant feilhåndtering
   (b) maks antall iterasjoner og toleranse der det er relevant
7. Det skal komme godt fram, både gjennom programmet og i teksten i Markdown, at du forstår det kjemiske og matematiske innholdet.
8. Ikke kopier kode direkte. Du kan bruke kode fra læreverk, internett og liknende som inspirasjon, men modifiser og forklar koden slik at det kommer godt fram at du forstå alt som skjer.

Del I

Ekstraoppgaver til fellesdelen

Oppgave 1: Tall og variabler

Læringsmål 1.1: Tall og variabler

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Definere variabler av ulik variabeltype.
2. Bruke kommentarer på en hensiktsmessig måte.
3. Utføre grunnleggende regneoperasjoner.
4. Importere og bruke bibliotekter.
5. Skrive ryddig og oversiktlig kode.

Oppgave 1.1*

a) Lag et program der du definerer innholdet av H-, C- og O-atomer i et karbohydrat i hver sine variabler. Den molare massen til hver av grunnstoffene skal også tilordnes hver sin variabel. Print så ut den molare massen til sukrose ($C_{12}H_{22}O_{11}$).

b) Lag en variabel som inneholder massen av innveid sukrose. Modifiser programmet fra oppgave a) slik at det printer ut hvor mange mol sukrose dette er, og hvor mange sukrosemolekyler dette tilsvarer.

Oppgave 1.2

Når vi fortynner en løsning, er antall mol av stoffet vi fortynner likt før og etter fortynningen: $n_0 = n_1$. Siden $n = c \cdot V$, der c er konsentrasjonen i mol/L og V er volumet i L, får vi følgende sammenheng:

$$c_1 V_1 = c_2 V_2 \tag{1.0.1}$$

Dette kaller vi *fortynningsloven*.

Lag et program som regner ut den nye konsentrasjonen hvis du fortynner 25 mL løsning med konsentrasjon 0,50 mol/L til 1,0 L.

Oppgave 1.3

Mellom et elektron og et proton i et hydrogenatom virker det hovedsakelig to krefter: Én elektrisk kraft og én gravitasjonskraft. Den elektromagnetiske kraften kan uttrykkes litt forenklet ved Coloumbs lov:

$$F_e = k_e \frac{q_1 q_2}{r^2} \tag{1.0.2}$$

hvor k_e er Coloumbs konstant ($8.987742438 \cdot 10^9$ $\mathrm{Nm^2C^{-2}}$), q_1 og q_2 er ladningen til de to partiklene, og r er avstanden mellom partiklene. Siden vi har å gjøre med et proton og et elektron, er ladningen lik elementærladningen e: $q_{proton} = e = 1.602176634 \cdot 10^{-19}$ C og $q_{elektron} = -e$.

Gravitasjonskraften kan vi uttrykke ved hjelp av Newtons 2. lov:

$$F_g = G \frac{m_1 m_2}{r^2} \tag{1.0.3}$$

der F_G er gravitasjonskreftene som virker mellom to legemer med henholdsvis masse m_1 og m_2 med avstand r mellom hverandres tyngdesentre. G er gravitasjonskonstanten ($G = 6.674 \cdot 10^{-11}$ $\mathrm{Nkg^{-2}m^2}$), og avstanden mellom et proton og et elektron i hydrogenatomet er omtrent lik Bohr-radiusen, $a_0 = 5.292 \cdot 10^{-11}$ m. Massen til et proton og et elektron er henholdsvis $m_p = 1.672 \cdot 10^{-27}$ kg og $m_e = 9.109 \cdot 10^{-31}$ kg.

a) Legg alle konstanter i hver sine variable. Bruk kommentarer til å spesifisere hva de ulike variablene betyr og hvilken enhet de har. Benytt også gode variabelnavn.

b) Lag to nye variabler der du beregner den elektriske kraften og gravitasjonskraften som virker mellom et elektron og et proton i hydrogenatomet.

c) Print ut forholdet mellom kreftene med en beskrivende svarsetning. Hva sier dette deg? Legg svaret på dette spørsmålet i en flerlinjekommentar til slutt i programmet.

Oppgave 2: Løkker og lister

Læringsmål 2.1: Løkker og lister

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Definere og trekke ut informasjon fra lister.
2. Iterere over lister med løkker.
3. Benytte for-løkker med et gitt antall iterasjoner.
4. Benytte while-løkker med et gitt kriterium.

Oppgave 2.1*

Lag et program som printer ut pH-en til løsninger med $[H_3O^+] \in [1 \cdot 10^{-15}, 1]$ mol/L med 0.1 mol/L mellom hver verdi.

Oppgave 2.2

Ideelle gasser kan beskrives med følgende likning:

$$PV = nRT \tag{2.0.1}$$

der P er trykket i pascal, V er volumet i m^3, n er stoffmengden i mol, T er temperaturen i kelvin og $R = 8.31\ \mathrm{m^2 kg s^{-2} K^{-1} mol^{-1}}$ er gasskonstanten. Vi ønsker å regne ut volumet gitt et sett med trykkverdier av 1 mol gass ved $T = 300$ K.

a) Du skal lage en liste `P` med N likt fordelte verdier av $P \in [100, 400)$ kPa og en annen liste `V` med det tilsvarende volumet. Lag først en steglengde, `steg`, som sier noe om hvor langt det er mellom hver trykkverdi i intervallet. Husk at steglengden må være et heltall. Fyll så verdier i listene ved hjelp av en løkke.

b) Sjekk om listene er N lange ved å bruke `len`-funksjonen. Rett opp i programmet hvis de ikke er det.

c) Plott P som funksjon av V. Tips til hvordan du kan plotte, finner du nedenfor.

```
import matplotlib.pyplot as plt

plt.plot(P,V)
plt.show()
```

Oppgave 2.3

Vi kan bruke Bohrs formel til å regne ut frekvensen til et foton som emitteres når et elektron i et hydrogenatom deeksiterer fra skall n til m:

$$f = \frac{B}{h} \cdot \left(\frac{1}{m^2} - \frac{1}{n^2} \right) \tag{2.0.2}$$

der $B = 2.18 \cdot 10^{-18} J$ er Bohrs konstant, $h = 6.63 \cdot 10^{-34}$ $\mathrm{m^2 kg\ s^{-1}}$. Vi har også en sammenheng mellom frekvens og bølgelengden til fotonene:

$$\lambda = \frac{c}{f} \tag{2.0.3}$$

der $c = 3.00 \cdot 10^8$ m/s er lysets hastighet i vakuum.

a) Lag en funksjon `hydrogenfotoner` som tar n og m som parametere og returnerer bølgelengden i nm til fotonet som emitteres.

b) Lag en testkjøring av funksjonen som skriver ut bølgelengden ved deeksitasjon fra $n = 5$ til $n = 2$. Dette skal gi $\lambda = 434\ nm$.

c) Lag et program som regner ut alle bølgelengdene til fotonene som emitteres når atomet deeksiterer fra et skall n til alle mulige energinivåer m. Gi n som variabel i starten av programmet.

Oppgave 3: Beslutninger og funksjoner

Læringsmål 3.1: Beslutninger og funksjoner

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Identifisere og programmere ulike utfall med if-tester.
2. Definere funksjoner med riktige parametere og returverdier.
3. Kalle på funksjoner.
4. Lage hensiktsmessige testkjøringer av funksjoner.

Oppgave 3.1*

pH-en i en buffer med bufferparene HA (svak syre) og A^- (korresponderende svak base) er gitt ved bufferlikningen:

$$pH = pK_a + log\left(\frac{[A^-]}{[HA]}\right) \tag{3.0.1}$$

a) Lag en funksjon som tar pK_a og konsentrasjonen av syra og korresponderende base som parametere. Funksjonen skal returnere pH-en i løsningen.

b) Test funksjonen for en eddiksyre-acetat-buffer ($pK_a = 4.76$). Kall på funksjonen med $[CH_3COOH] = 2.3 \cdot 10^{-3}$ mol/L og $[CH_3COO^-] = 2.0 \cdot 10^{-3}$ mol/L. Testen bør gi pH = 4.82.

Oppgave 3.2

En kjemisk reaksjon er spontan dersom graden av uorden (entropi) øker i systemet (positiv endring i entropi), og/eller reaksjonen er tilstrekkelig eksoterm (gir fra seg varme, dvs. negativ endring i entalpi). Vi gir endringen i entropi symbolet ΔS og endringen i entalpi symbolet ΔH.

For å holde styr på hvilke reaksjoner som går spontant, kan vi bruke Gibbs frie energi som tar høyde for både entropien og entalpien i reaksjonen. Dersom endringen i Gibbs frie energi i en reaksjon er negativ, er reaksjonen spontan. Sammenhengen er slik:

$$\Delta G = \Delta H - T\Delta S \tag{3.0.2}$$

Her er ΔG Gibbs frie energi, ΔH er entalpien, T er temperaturen i kelvin og ΔS er entropien. Vi har to lister med ulike verdier for entalpi og entropi:

```
S_liste = [-10, 0.9, 5.1, 1.5, -0.1, 3, -0.4, 0.5, 0.8, 0.1]
H_liste = [100, 12, -100, -75, -20, 10, 20, -35, 49, -10]
```

Lag et program som sjekker om en gitt kombinasjon av ΔS og ΔH fra listene ovenfor gir en spontan reaksjon eller ikke for $T = 298$ K. Alle kombinasjoner av listeelementene skal sjekkes. Programmet skal telle opp hvor mange reaksjoner som er spontane og skrive dette ut. I dette tilfellet skal 71 av 100 reaksjoner være spontane.

Oppgave 3.3

I en løsning er pH et mål på konsentrasjonen av oksoniumioner:

$$pH = -log([H_3O^+]) \tag{3.0.3}$$

a) Lag en funksjon som tar som parameter konsentrasjonen av oksoniumioner og returnerer pH-en i løsningen. Test funksjonen med $[H_3O^+] = 1 \cdot 10^{-5}$ mol/L, noe som vil gi $pH = 5$.

b) Utvid funksjonen slik at den også skriver ut hvorvidt løsningen er sur, nøytral eller basisk.

   En løsning vil sjeldent ha pH eksakt lik 7, og programmet vil derfor nesten aldri skrive ut at løsningen er nøytral. Vi kan kompensere for dette på ulike måter. Vi kan se på to mulige løsninger her. Den første kan være å runde av pH-en til et visst antall desimaler. Dette kan for eksempel gjøres med funksjonen `round(verdi, antall desimaler)`. Den andre løsningen går på å legge til en toleranseparameter, `tol`, slik at nøytral pH defineres som $pH = 7 \pm tol$.

c) Ta utgangspunkt i funksjonen fra b) og bruk avrundingsfunksjonen slik at vi får litt slingringsmonn på nøytral pH. Rund av til ett desimal og gjør justeringer i koden slik at f.eks. $[H_3O^+] = 1.1 \cdot 10^{-7}$ gir utskriften `«Løsningen er nøytral»`.

d) Ta utgangspunkt i funksjonen fra b) og legg inn en toleranseparameter i funksjonen som gjør at $[H_3O^+] = 1.3 \cdot 10^{-7}$ gir utskriften `«Løsningen er nøytral»` dersom toleransen er 0.2.

e) Hvilke fordeler er det med å bruke en toleranseparameter i slike beregninger?

Oppgave 4: Feilhåndtering og fildata

Læringsmål 4.1: Feilhåndtering og fildata

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Ta input med input-funksjonen og fra kommandolinja.
2. Bruke if-tester i feilhåndtering.
3. Bruke `exceptions` i feilhåndtering.
4. Bruke `assertions` i feilhåndtering.
5. Lese fra og skrive til fil.

Oppgave 4.1

Modifiser bufferprogrammet fra oppgave 3.1 slik at det tar alle parametere som input fra brukeren ved hjelp av (lag to ulike programmer):

a) Input-funksjonen.

b) Argumenter fra kommandolinja.

Oppgave 4.2

En bufferløsning har bufferområde lik $pH = pK_a \pm 1$. Dersom pH-en beveger seg utenfor dette området, er bufferen *sprengt* og mister sine bufferegenskaper. Modifiser programmet fra oppgave 3.1 slik at du får en feilmelding dersom opplysningene du gir, fører til at bufferen sprenges.

Oppgave 4.3*

Den molare massen til et grunnstoff kan beregnes ved å ta summen av den molare massen til hvert av isotopene til grunnstoffet multiplisert med andelen/forekomsten av denne istopen:

$$\sum_i m_i w_i \tag{4.0.1}$$

Her er m_i massen til den i-te isotopen av grunnstoffet, og w_i er forekomsten av denne istopen (fra 0 til 1, der 1 er 100 prosent forekomst). Fila `tinn.txt` inneholder forekomsten i % av ulike isotoper av tinn, som er grunnstoffet med mest stabile isotoper.

a) Les av fila og legg nuklidetallet i en liste, massen i en annen liste og forekomsten i prosent/100 i en tredje liste.

b) Bruk informasjonen til å regne ut den molare massen til tinn. Print ut massen med fire desimaler og korrekt enhet.

c) Legg inn en feilhåndtering med try-except som gir en feil dersom fila ikke kan leses.

Oppgave 4.4

Fila `titrering.txt` inneholder en kolonne med tilsatt volum NaOH i mL til en løsning med eddiksyre og en kolonne med pH.

Les av fila og lag to lister eller arrayer med navn *volum* og *pH* med de aktuelle verdiene i. Plott titrerkurven (se oppgave 2).

Oppgave 5: Arrayer, dictionarier og plotting

Læringsmål 5.1: Arrayer og plotting

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Opprette og utføre operasjoner på arrayer.
2. Opprette og utføre operasjoner på dictionarier.
3. Vektorisere kode.
4. Plotte data og modifisere plottene med tittel, aksetekst, graftekst, farge, markører og ulik linjestil.

Oppgave 5.1

Et program skal regne ut `c = a + b` og `c = a * b`. Forklar hva c vil bli dersom

1. `a = [1,2,3,4]` og `b = [1,2,3,4]`
2. `a = [1,2,3,4]` og `b = 2`
3. `a = array([1,2,3,4])` og `b = array([1,2,3,4])`
4. `a = array([1,2,3,4])` og `b = 2`

Beskriv hva som skjer i de ulike tilfellene.

Oppgave 5.2

Arrayer kan i mange tilfeller gi kode som er raskere og kortere enn tilsvarende kode med lister. Men arrayer kan også behandles komponentvis gjennom løkker slik som lister. I programmet nedenfor gjøres dette. For å gjøre koden raskere, kan vi *vektorisere* koden. Det vil si å utføre operasjoner gruppevis på et element istedenfor å gjøre det komponentvis.

a) Vektoriser koden nedenfor.

```
import numpy as np
N = 1000
a = np.linspace(0, 10, N)
b = np.linspace(1, 11, N)
c = np.zeros(N)

for i in range(N):
    c[i] = a[i] + b[i]
```

b) Følgende kode regner ut pH i en løsning gitt ulike verdier av konsentrasjonen av oksoniumioner. Vektoriser koden.

```
import numpy as np
H3O = [1E-10, 2.4E-9, 1E-8, 3.5E-7, 7E-6, 1.2E-5, 1E-4, 1E-2, 1.2]

def surhet(oksonium):
    pH = -np.log10(oksonium)
    return pH

pH = []

for kons in H3O:
    pH.append(surhet(kons))
```

c) Lag en array med 1000 verdier av $[H_3O^+]$ i intervallet $[1 \cdot 10^{-16}, 1]$. Lag de tilsvarende pH- og pOH-verdiene i hver sin array. Plott pH og pOH som funksjon av $[H_3O^+]$. Lag aksetitler, graftekst (legend) og tittel, og pynt plottet slik at det blir en fin framstilling. Lag også et plott med logaritmisk x-akse.

Oppgave 5.3*

Vi ser på en dictionary som skal lagre informasjon om grunnstoffer.

a) Lag en dictionary med navnet til de åtte første grunnstoffene som nøkler og den tilhørende massen som verdier.

b) Legg inn grunnstoff 9 og 10 uten å redigere rett i dictionarien.

c) Skriv ut alle nøklene og alle verdiene.

d) Bruk ei løkke og skriv ut alle nøklene med tilhørende masser på følgende format:

```
Massen til H er 1.008
Massen til He er 4.003
```

```
Massen til Li er 6.94
Massen til Be er 9.012
...
```

e) Utvid dictionarien slik at en ny dictionary er verdien til hver av nøklene. Disse *nøstede* dictionariene er nyttige når vi skal tilknytte mer informasjon til hver nøkkel. Legg nye nøkler for masse, atomnummer, gruppe og periode inn i disse dictionariene. Skriv så ut atomnummeret til fluor og massen til karbon.

Oppgave 5.4

Elektroner har bølgeegenskaper, blant annet en frekvens (svingninger per sekund) og en bølgelengde (avstand mellom tilsvarende steder på bølgen, f.eks. mellom bølgetopper). En annen egenskap kalles *bølgetallet* k. Mens frekvensen beskriver antall bølgelengder (målt i sykluser/radianer) per tidsenhet, beskriver bølgetallet antall bølgelenger over en viss distanse:

$$k = \frac{2\pi}{\lambda} \tag{5.0.1}$$

Siden både frekvens og bølgelengden til en bølge har sammenheng med energien til bølgen, har også bølgetallet det. I faste stoffer er elektronene delokaliserte (fordelt i en «elektronsjø») og ligger nært hverandre i energi. Dette danner en såkalt *båndstruktur* med energibånd. Disse båndene er separert med *båndgap*. Dette er energiintervaller med energier som elektronet ikke kan ha. Fila `filebands.txt` inneholder bølgetall og tilsvarende energier for de tre første båndene til et fast stoff.

Lag et program som leser av fila og plotter energien som funksjon av k. Lag aksetitler og pynt på grafen slik at den blir fin. Hva forteller plottet deg?

Oppgave 6: Diskrete beregninger

Læringsmål 6.1: Diskrete beregninger

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Beregne rekkesummer iterativt.
2. Beregne neste ledd i en matematisk følge og differenslikning.
3. Benytte løkker til å løse iterative prosesser (differenslikninger) fra ulike fagområder.

Oppgave 6.1

Du har følgende geometriske rekke:

$$3 + 1 + \frac{1}{3} + \frac{1}{9} + \frac{1}{27} + ... \tag{6.0.1}$$

a) Finn summen av rekka.

b) Modifiser programmet slik at det avgjør hvorvidt rekka konvergerer eller divergerer.

Oppgave 6.2

Fibonnaci-tallene er bygd opp slik at hvert tall i rekka er summen av de to foregående tallene i rekka:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... \tag{6.0.2}$$

Lag et program som finner tall nummer N i Fibonnaci-følgen.

Oppgave 6.3

Du tar opp et lån på L kroner med nedbetalingstid N måneder. Den månedlige nedbetalingen er L/N i tillegg til renter av lånet, p prosent per år. Vi regner derfor med en fast rente på

p/12 per måned. Vi innfører en vekstfaktor $k = \frac{p}{12 \cdot 100}$. Da er summen av avdrag og renter hver måned gitt ved:

$$y_{n+1} = k \cdot x_n + \frac{L}{N} \tag{6.0.3}$$

Verdien til lånet kan da beskrives slik:

$$x_{n+1} = x_n + k \cdot x_n - y_{n+1} \tag{6.0.4}$$

Lag en Python-funksjon som regner ut hvor mye du betaler hver måned dersom du tar opp et samlet studielån på 450 000 kroner med en rente på 2.5 % og nedbetalingstid på 30 år. Renter og avdrag første måned er lik 0. Plott y som funksjon av tida t i måneder. Sjekk om lånet er nedbetalt ved å plott lånets verdi x som funksjon av tida t.

Oppgave 6.4

En harepopulasjon p som utvikler seg uten ressursbegrensninger og uten rovdyr kan beskrives med følgende modell:

$$p_{t+\Delta t} = p_t + k \cdot p_n \tag{6.0.5}$$

Her er $p_{t+\Delta t}$ antall harer ved neste tidssteg, $t + \Delta t$, p_n antall harer ved tida t og k vekstfaktoren (i prosent/100).

a) Lag et plott som beskriver antall harer etter tre år med en vekst på 25 % per måned. Vi starter med 42 harer.

Dersom vi har faktorer som begrenser utviklingen av antall harer i økosystemet, kan vi innføre en bæreevne B som bremser utviklingen. Bæreevnen er maks antall individer av en art i et økosystem. Da kan utviklingen beskrives med denne modellen:

$$p_{t+\Delta t} = p_t + k \cdot p_t \left(1 - \frac{p_t}{B}\right) \tag{6.0.6}$$

b) Modifiser programmet ovenfor slik at det tar hensyn til en bæreevne $B = 1030$ harer. Plott utviklingen.

Oppgave 6.5

Vi kan tilnærme funksjonen $f(x) = e^x$ med et Taylorpolynom av grad n slik:

$$e^x \approx \sum_{n=0}^{\infty} \frac{x^n}{n!} \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + ... \tag{6.0.7}$$

a) Lag en funksjon `exp_taylor` som tar et punkt x og graden til polynomet n som parametere. Funksjonen skal returnere funksjonsverdien til Taylor-polynomet i punktet ved å benytte summeformelen ovenfor.

b) Lag en løkke som regner ut verdien til e^x for alle heltall i intervallet $x \in [1, 10]$ med funksjonen din. Regn også ut den «eksakte» verdien ved hjelp av `exp`-funksjonen til numpy. Skriv ut den absolutte feilen ved å ta differansen mellom den eksakte verdien og den estimerte.

En annen kjent funksjon vi kan tilnærme med Taylor-polynomer, er $f(x) = cos(x)$:

$$cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + ... \tag{6.0.8}$$

c) Lag en funksjon `cos_taylor` som tar et punkt x og graden til polynomet n som parametere. Funksjonen skal returnere funksjonsverdien til Taylor-polynomet i punktet ved å benytte summeformelen ovenfor.

d) Lag en array `x` med 1000 likt fordelte verdier i intervallet $x \in [-5, 5]$ og plott $f(x) = cos(x)$. Plott også en Taylor-polynomtilnærming med seks ledd i samme koordinatsystem. Bruk merkelapper (labels/legend) for å skille på kurvene.

e) Bruk funksjonen din til å plotte fire ulike tilnærminger til $cos(x)$ ved hjelp av Taylor-polynomfunksjonen din i samme koordinatsystem som i d). De fire tilnærmingene skal ha grad fra 2–5, og merkelappene skal inneholde informasjon om graden til polynomet.

Del II

Obligatoriske oppgaver

Oblig 7: Datahåndtering

Læringsmål 7.1: Statistikk

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Lage egne funksjoner til statistisk beregning.
2. Bruke biblioteker til statistisk beregning.
3. Beregne gjennomsnitt, standardavvik og relativ og absolutt feil.
4. Utføre regresjonsanalyse på eksperimentelle data.
5. Benytte maskinlæringsalgoritmer til å utforske datasett og danne hypoteser om sammenhenger.

Oppgave 7.1

Vi bruker et datasett vi har fått fra en analyse av innholdet av Pb^{2+} i bekkevann som utgangspunkt. Til dette er det brukt spektroskopisk analyse. I spektroskopiske analyser finner vi konsentrasjonen til et stoff ved å undersøke hvor mye lys stoffet absorberer av en bestemt bølgelengde. For å finne konsentrasjonen i en ukjent løsning, lager vi først en *standardkurve* ut fra absorpsjonen til løsninger med kjent konsentrasjon. Vi analyserer en rekke løsninger og får følgende resultater:

Konsentrasjon (ppm)	**Absorbans**
0.0	0.0
0.100	0.116
0.200	0.216
0.300	0.310
0.400	0.425
0.500	0.520

a) Lag et program som gjør lineær regresjon på dataene og plotter datapunktene og den tilpassede regresjonskurven i samme koordinatsystem.

b) Analysen ved 283 nm av vannprøva ga absorbans på 0.340. Bruk standardkurven og programmet til å bestemme konsentrasjonen av blyioner i vannprøva i ppm.

Oppgave 7.2

a) Lag et program som inneholder en funksjon *gjennomsnitt* og en funksjon *standardavvik* som regner ut gjennomsnittet og standardavviket gitt en liste med datapunkter som parameter. Test funksjonene på `liste = [1,2,2,1,3,3]` og sammenlikn med numpy-funksjonene `mean` og `std`.

b) Benytt funksjonene du lagde i a) til å regne ut gjennomsnittet og standardavviket av følgende målinger gjort av koffein i te med væskekromatografi:

Injeksjon	**Konsentrasjon (mg/mL)**
1	245
2	272
3	252
4	264
5	261
6	272
7	255
8	260
9	268
10	259

Oppgave 7.3

Bruk pandas-biblioteket til denne oppgaven. Fila `vin.csv` beskriver ulike kjemiske parametre i 1500 rødviner, i tillegg til en vurdering av vinkvaliteten, på en skala fra 1–8.

1. Les fila og beskriv de ulike kategoriene.
2. Lag et korrelasjonsplott med utgangspunkt i alle kategoriene i datasettet. Hvilke faktorer ser ut til å korrelere med god vinkvalitet? Gi også eksempler på faktorer som ikke korrelerer og faktorer som har negativ korrelasjon. Forklar hva dette betyr. Prøv gjerne å forklare noen av korrelasjonene.
3. Lag et søylediagram med vinkvalitet på førsteaksen og det totale innhold med svoveldioksid på andreaksen. Hva kan årsaken være til denne fordelingen? Sammenlikn med korrelasjonen mellom disse faktorene.
4. Lag en ny kolonne i datasettet som inneholder «kvalitetskategorien» til vinen. Den skal inneholde 0 hvis vinen har under 6 i kvalitet, og 1 hvis den har 6 eller mer.
5. Lag en modell som skal forutsi kvalitetskategorien til vinen. Bruk en bestemmelsestre-algoritme som grunnlag for modellen.
6. Test og valider modellen din. Kommenter resultatet. Hvorfor brukte vi en egen kvalitetskategori og ikke vinkvaliteten på en skala fra 1–8 direkte?

Oblig 8: Optimering og løsing av likninger

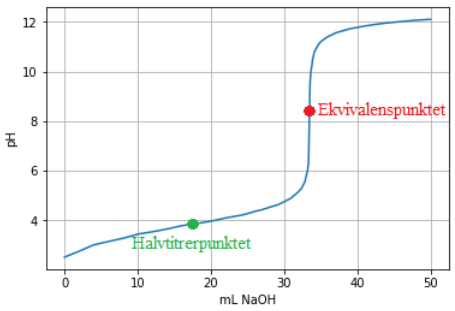
Læringsmål 8.1: Optimering og løsing av likninger

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Bruke numerisk derivasjon til å analysere eksperimentelle data.
2. Finne ekstremalpunkter vha. numerisk derivasjon, og tolke resultatene.
3. Implementere halveringsmetoden som en funksjon med relevant feilhåndtering.
4. Bruke halveringsmetoden for å finne nullpunkter og løse likninger.

Teoretisk bakgrunn

Titrering er en kvantitativ analysemetode der vi bestemmer konsentrasjonen av et ukjent stoff (*analytten*) ved å tilsette et stoff med kjent konsentrasjon (*titranten*). Titranten tilsettes ofte fra en byrette, og vi kan notere oss pH i analytten underveis ettersom vi tilsetter et visst volum titrant. Her er en titreringskurve for titrering av en svak syre med en sterk base.



Figur 8.0.1: Titreringskurve der en svak syre titreres med en sterk base.

Ved ekvivalenspunktet er grafen brattest, og her er stoffmengdene av syre og base *ekvivalente* (og dermed like hvis forholdet er 1:1 i reaksjonslikningen). Dette kan vi bruke til å finne konsentrasjonen av analytten. Ved halvtitrerpunktet er $pH = pK_a$, og vi har en bufferløsning. Derfor endrer pH-en seg ikke så mye rundt dette punktet. Vi skal se på noen metoder for å finne ekvivalenspunktet og pH-en ved ekvivalenspunktet i en slik titrering.

pH ved ekvivalenspunktet

Vi kan utlede likninger som gir oss pH-en ved tilsetting av et visst volum syre til base, eller motsatt. Når vi titrerer, tilsette vi sterke syrer til baser eller sterke baser til syrer. Derfor konsenterer vi oss om å utlede likninger som kan beskrive pH-en i slike løsninger. Vi tar utgangspunkt i en generell syre-base-titrering der vi titrerer en svak syre (HA) med en sterk base (BOH):

$$HA(aq) + BOH(aq) \rightleftharpoons B^+(aq) + A^-(aq) + H_2O(l) \tag{8.0.1}$$

Etter at en slik titrering er fullført, kjenner vi til følgende:

1. Konsentrasjonen av basen vi brukte (*titranten*): c_b.
2. Volumet base vi brukte for å nå ekvivalenspunktet: V_b.
3. Volumet syre vi titrerte (*analytten*): V_a.
4. Den tidligere ukjente konsentrasjonen av syra: c_a.

Siden vi kjenner disse verdiene, er de å regne som konstanter når vi skal finne pH i løsningen ved ekvivalenspunktet. Vi trenger å formulere en likning som inneholder $[H^+]$, og ingen andre ukjente. La oss prøve å utlede en slik likning. Et godt utgangspunkt kan være at summen av negative ioner til enhver tid skal være lik summen av positive ioner, siden løsningen alltid er elektrisk nøytral:

$$[H^+] + [B^+] = [OH^-] + [A^-] \tag{8.0.2}$$

Denne sammenhengen kan vi ta utgangspunkt i, men vi trenger å eliminere alle de ukjente konsentrasjonene bortsett fra $[H^+]$, som vi jo skal finne. Merk at $[B^+]$ og $[A^-]$ er konsentrasjonen i løsningen etter tilsatt titrant, mens c_b og c_a er konsentrasjon av base i titranten og syre i analytten *før* titreringen starter (initialbetingelser).

Konsentrasjonen av den korresponderende syra, $[B^+]$, til den sterke basen i *løsningen* ved enhver tid er gitt ved antallet mol base tilsatt delt på totalvolumet av løsningen. Dette er fordi vi regner med at all basen vi tilsetter, har reagert. Antall mol base tilsatt er det samme som konsentrasjonen av basen ved start (c_b) multiplisert med volumet av base som er tilsatt. Deretter får vi konsentrasjonen av base i løsningen ved å dele på totalvolumet av løsningen, $V_t = V_a + V_b$. Da får vi at:

$$[B^+] = \frac{c_b V_b}{V_t} \tag{8.0.3}$$

Da har vi altså fått et uttrykk for konsentrasjonen til den sterke basen, som vi kan sette inn i likning 8.0.2. Vi mangler da blant annet et uttrykk for $[OH^-]$. Dette kan vi få fra uttrykket for *vannets egenprotolyse*:

$$K_w = [OH^-][H^+] \tag{8.0.4}$$

der $k_w = 10^{-14}$. Da har vi nok et uttrykk som vi kan sette inn i likning 8.0.2:

$$[OH^-] = \frac{K_w}{[H^+]} \tag{8.0.5}$$

Vi mangler nå et uttrykk for $[A^-]$. Vi kan utlede dette fra følgende sammenheng:

$$n_{A^-} = n_{syre,start} - n_{HA} \tag{8.0.6}$$

Denne sammenhengen sier at antall mol korresponderende base til en svak syre er lik antall mol syre vi startet med minus antall mol syre vi har igjen. Dette gir følgende sammenheng:

$$[A^-] \cdot V_t = c_a V_a - [HA] \cdot V_t \tag{8.0.7}$$

Et problem er at vi nå har en ukjent til i likning 8.0.7, nemlig $[HA]$. Denne kan vi derimot eliminere ved å benytte oss av uttrykket for likevektskonstanten for en svak syre:

$$K_a = \frac{[H^+][A^-]}{[HA]} \tag{8.0.8}$$

Dette gir følgende uttrykk for konsentrasjonen av syra, $[HA]$:

$$[HA] = \frac{[H^+][A^-]}{K_a} \tag{8.0.9}$$

Hvis vi substituerer $[HA]$ fra 8.0.9 inn i 8.0.7, får vi:

$$[A^-] = \frac{c_a V_a}{V_t[H^+]/K_a + V_t} = \frac{c_a V_a K_a}{V_t([H^+] + K_a)} \tag{8.0.10}$$

Nå har vi uttrykk for $[A^-]$, $[OH^-]$ og $[B^+]$ som kun avhenger av én ukjent, nemlig $[H^+]$. Da kan vi sette alt inn i likning 8.0.2:

$$[H^+] + \frac{c_b V_b}{V_t} = \frac{K_w}{[H^+]} + \frac{c_a V_a K_a}{V_t([H^+] + K_a)} \tag{8.0.11}$$

Vi samler alle ledd på samme side og ganger alle ledd med V_t for å fjerne flest mulige brøker:

$$[H^+]V_t + c_b V_b - \frac{K_w V_t}{[H^+]} - \frac{c_a V_a K_a}{[H^+] + K_a} = 0 \tag{8.0.12}$$

Nå har vi en likning som vi kan løse ved å finne nullpunktene til følgende funksjon:

$$T([H^+]) = [H^+]V_t + c_b V_b - \frac{K_w V_t}{[H^+]} - \frac{c_a V_a K_a}{[H^+] + K_a} \tag{8.0.13}$$

Funksjonen 8.0.13 er det derimot problematisk å finne nullpunktene til. Vi får nemlig asymptoter rundt nullpunktet fordi den ikke er definert når $[H^+] = 0$. Dette kan vi komme unna ved å gange med fellesnevner, $[H^+] \cdot (K_a + [H^+])$, i alle ledd. Da får vi følgende funksjon:

$$T([H^+]) = V_t([H^+] + K_a)[H^+]^2 + c_b V_b([H^+] + K_a)[H^+] - K_w V_t([H^+] + K_a) - c_a V_a K_a[H^+] \tag{8.0.14}$$

Vi ser at dette er en tredjegradsfunksjon som ikke så lett lar seg løses analytisk (men det er mulig!). Vi kan derfor bruke numeriske løsninger, for eksempel Newtons metode, for å løse denne.

Merk at likningen funksjonen ikke har noen fysisk eller kjemisk betydning – men et av nullpunktene til funksjonen gir pH-en i en blanding der sterk base er tilsatt svak syre.

Oppgaver

Oppgave 8.1: Beregning av ekvivalenspunktet

a) Les og plott dataene fra fila *titreringsdata.txt*, som viser titreringsdata for titrering av glykolsyre med NaOH. Sørg for at datapunktene vises i plottet.

b) Deriver pH-en numerisk med hensyn på volumet og legg den deriverte pH-en i ei ny liste. Forklar hva den deriverte av pH-en kan fortelle oss.

c) Lag en funksjon som finner den største deriverte i den deriverte lista. Sammenlikn gjerne med numpy-funksjonen *max*. La programmet skrive ut hvilket volum dette tilsvarer. Dette er volumet sterk base som er tilsatt ved ekvivalenspunktet. Finn også pH ved ekvivalenspunktet ved hjelp av programmet ditt.

Oppgave 8.2: Beregning av pH ved ekvivalenspunktet

Titreringen er utført slik at 0.0100 L glykolsyre ($K_a = 1.50 \cdot 10^{-4}$) ble titrert med 0.500 M NaOH. Forbruket av NaOH var 33.52 mL.

a) Regn ut konsentrasjonen av syra og lag alle variablene du trenger for å kunne regne med likning 8.0.14.

b) Tilpass kodesnutten under til egen kode og regn ut pH-en ved ekvivalenspunktet. Hvordan passer dette med utregningen i B.1 (pH skal bli 8.70)? Kommenter resultatet.

```
# Utregna pH-verdi ved ekvivalenspunktet
Kb = 6.67E-11
V_total = (V_NaOH + V_glykolsyre)
c_glykolsyre_ekv = n_glykolsyre/V_total
c_OH = np.sqrt(Kb*c_glykolsyre_ekv)
pOH = -np.log10(c_OH)
pH = 14-pOH
print(f'pH ved ekvivalenspunktet er {pH}')
```

c) Lag en Python-funksjon av funksjonen i 8.0.14. Plott funksjonen i intervallet $[-1, 1]$. Hva ser du? Prøv å plotte i intervallet $[-1 \cdot 10^{-8}, 1 \cdot 10^{-8}]$. Hvordan ser det ut nå? Negativ konsentrasjon gir ingen mening reint kjemisk, men vi bruker det kun teoretisk her for å finne nullpunktene til funksjonen vår.

d) Forklar hvordan Newtons metode fungerer.

e) Lag en funksjon som benytter Newtons metode til å finne nullpunktene til 8.0.14. Begrunn valget av startpunktet a.

f) Beregn pH-en ut fra det du fikk i forrige oppgave. Kommenter eventuelle avvik eller likheter med de andre beregningene.

Oblig 9: Dynamiske systemer og integrasjon

Læringsmål 9.1: Dynamiske systemer og numerisk integrasjon

I disse oppgavene skal du lære og vise at du behersker følgende:

1. Plotte og illustrere utviklingen i et dynamisk system.
2. Simulere fartslov.
3. Sammenlikne eksperimentelle data med simuleringer.
4. Forstå hva som menes med numerisk integrasjon, og gjøre rede for ulike metoder som kan brukes til dette.

Teoretisk bakgrunn

Chapman-modellen kan benyttes for å simulere produksjon og nedbrytning av ozon i stratosfæren. Den er basert på følgende reaksjonslikninger med tilhørende reaksjonskoeffisienter:

$$\mathrm{O_2} + h\nu \rightarrow 2\mathrm{O} \qquad k_1 = 3.0 \times 10^{-12}\ \mathrm{s^{-1}}, \tag{9.0.1}$$

$$\mathrm{O_2} + \mathrm{O} + \mathrm{M} \rightarrow \mathrm{O_3} + M \qquad k_2 = 1.2 \times 10^{-33}\ \mathrm{cm^6\ molec^{-2}\ s^{-1}}, \tag{9.0.2}$$

$$\mathrm{O_3} + h\nu' \rightarrow \mathrm{O} + \mathrm{O_2} \qquad k_3 = 5.5 \times 10^{-4}\ \mathrm{s^{-1}}, \tag{9.0.3}$$

$$\mathrm{O} + \mathrm{O_3} \rightarrow 2\mathrm{O_2} \qquad k_4 = 6.9 \times 10^{-16}\ \mathrm{cm^3\ molec^{-1}\ s^{-1}}, \tag{9.0.4}$$

hvor O, $\mathrm{O_2}$ og $\mathrm{O_3}$ er henholdsvis oksygen, dioksygen og ozon. M er en ikke-reagerende støtpartner[1], mens $h\nu$ og $h\nu'$ er energi tilført av UV-stråling med bølgelengde, λ, under 242 nm og 336 nm, henholdsvis.

(9.0.1) beskriver spaltingen av $\mathrm{O_2}$ til 2 O-atomer som resultat av UV-stråling. (9.0.2) er den påfølgende reaksjonen mellom $\mathrm{O_2}$ og O som krever en kollisjon med M for å danne $\mathrm{O_3}$, mens (9.0.3) og (9.0.4) viser hvordan $\mathrm{O_3}$ brytes ned henholdsvis som resultat av UV-stråling for å danne O og $\mathrm{O_2}$, og gjennom reaksjon med O for produksjon av 2 $\mathrm{O_2}$-molekyler.

[1] S. S. Zumdahl, D. J. DeCoste, *Chemical Principles* **2015**, 18.11 for mer om ikke-reagerende støtpartner.

Ratelikningene for [O], [O_2] og [O_3] er gitt ved henholdsvis

$$\frac{d[\mathrm{O}]}{dt} = 2k_1[\mathrm{O_2}] - k_2[\mathrm{O_2}][\mathrm{O}][\mathrm{M}] + k_3[\mathrm{O_3}] - k_4[\mathrm{O}][\mathrm{O_3}], \tag{9.0.5}$$

$$\frac{d[\mathrm{O_2}]}{dt} = -k_1[\mathrm{O_2}] - k_2[\mathrm{O_2}][\mathrm{O}][\mathrm{M}] + k_3[\mathrm{O_3}] + 2k_4[\mathrm{O}][\mathrm{O_3}], \tag{9.0.6}$$

$$\frac{d[\mathrm{O_3}]}{dt} = k_2[\mathrm{O_2}][\mathrm{O}][\mathrm{M}] - k_3[\mathrm{O_3}] - k_4[\mathrm{O}][\mathrm{O_3}]. \tag{9.0.7}$$

Konsentrasjonene er gitt i molekyler per kubikkcentimerer (molec/cm^3). Steady-state-approksimasjonen[2] sier at konsentrasjonen av intermediatene i en reaksjon er konstant, hvilket leder til uttrykkene (9.0.8)–(9.0.9) for [O_3] og [O] dersom vi regner med at [O_2] er konstant lik $[\mathrm{O_2}]_{t=0}$ under hele forløpet.

$$[\mathrm{O_3}] = \sqrt{\frac{k_1 k_2}{k_3 k_4}}[\mathrm{O_2}][\mathrm{M}]^{\frac{1}{2}}, \tag{9.0.8}$$

$$[\mathrm{O}] = \frac{k_3[\mathrm{O_3}]}{k_2[\mathrm{O_2}][\mathrm{M}]}. \tag{9.0.9}$$

Ratekonstantene i (9.0.1)–(9.0.4) er gitt ved omtrent 25 km høyde, hvor $[\mathrm{M}] \approx 9.0 \times 10^{17}$ molec cm^{-3}. Systemet har følgende initialbetingelser:

$$[\mathrm{O_2}]_{t=0} = 0.21[\mathrm{M}], \tag{9.0.10}$$

$$[\mathrm{O}]_{t=0} = 0, \tag{9.0.11}$$

$$[\mathrm{O_3}]_{t=0} = 0. \tag{9.0.12}$$

Oppgave 9.1

a) Skriv et program som beregner og plotter [O_3] og [O] som funksjon av tid i intervallet $t \in [0, 100]$ ved å benytte Forward Euler-algoritmen på fartslovene 9.0.5–9.0.7 med initialbetingelsene (9.0.10)–(9.0.12) og tidssteg $h = 0.001$. Plott med logaritmisk skala på y-aksen (`plt.yscale('log')`).

b) Beregn og plott de samme verdiene med en backward-metode ('BDF) ved å bruke funksjonen `scipy.integrate.solve_ivp` fra Scipy-biblioteket for $t \in [0, 10^8]$. Evaluer punktene i `t = np.linspace(t0,tid_slutt,int(1E6))`.

c) Sammenlikn de beregnede verdiene med BDF for [O_3] og [O] (de siste i arrayene) ved $t = 10^8$ s med verdiene fra steady-state-approksimasjonen. Beregn og kommenter det prosentvise avviket mellom steady-state og BDF-beregningene.

[2] S. S. Zumdahl, D. J. DeCoste, *Chemical Principles* **2015**, 15.7 for mer om steady-state-approksimasjonen.

Oppgave 9.2

a) Forklar hvorfor fartslover er differensiallikninger. Hva finner vi når vi integrerer dem med Eulers metode?

b) Implementer rektangelmetoden og trapesmetoden for numerisk integrasjon som Python-funksjoner. Bruk begge metodene til å integrere en funksjon der du kjenner den analytisk integrerte. Sammenlikn de to metodene