

# Kjemirettede oppgaver til INF1100

Sverre Løyland

Høst 2015

[illegible]



# Innhold

<b>Forord</b>	<b>v</b>
<b>Oppgaver til kapittel 1</b>	<b>1</b>
Oppgave 1.1: Glukose – <code>glucose.py</code> . . . . .	1
Oppgave 1.2: Nernst likning – <code>nernst_equation.py</code> . . . . .	1
Oppgave 1.3: Ideelle gasser – <code>ideal_gas.py</code> . . . . .	1
Oppgave 1.4: Reelle gasser – <code>real_gas.py</code> . . . . .	2
<b>Oppgaver til kapittel 2</b>	<b>3</b>
Oppgave 2.1: Energinivåer – <code>energy_levels.py</code> . . . . .	3
Oppgave 2.2: Elektronskall – <code>electrons_in_shell.py</code> . . . . .	3
<b>Oppgaver til kapittel 3</b>	<b>5</b>
Oppgave 3.1: Nernst likning som funksjon – <code>nernst_function.py</code> . . . . .	5
Oppgave 3.2: Funksjon for energi i elektronskall – <code>electron_shell_function.py</code> . . . . .	5
Oppgave 3.3: pH-titrering – <code>pH_titration.py</code> . . . . .	5
Oppgave 3.4: Orbitalsannsynlighet – <code>orbital_probability.py</code> . . . . .	6
<b>Oppgaver til kapittel 4</b>	<b>7</b>
Oppgave 4.1: Forbedring av skript for Nernst likning – <code>nernst_equation2.py</code> . . . . .	7
Oppgave 4.2: Forbedring av skript for titrering – <code>pH_titration2.py</code> . . . . .	7
<b>Oppgaver til kapittel 5</b>	<b>9</b>
Oppgave 5.1: Statistikkfunksjoner – <code>statistic_functions.py</code> . . . . .	9
Oppgave 5.2: Stirlings approksimasjon – <code>stirling.py</code> . . . . .	10
Oppgave 5.3: Spektrofotometri – <code>spectrofotometry.py</code> . . . . .	10
Oppgave 5.4: pH-måling – <code>pH_measuring.py</code> . . . . .	11
Oppgave 5.5: Bjerrumplot – <code>bjerrum_plot.py</code> . . . . .	12
Oppgave 5.6: Haber-Bosch-prosessen – <code>Haber-Bosch.py</code> . . . . .	12
Oppgave 5.7: Damptrykk – <code>vapor_pressure.py</code> . . . . .	13
Oppgave 5.8: Aktiveringsenergi – <code>activation_energy.py</code> . . . . .	13
Oppgave 5.9: Bølgefunksjon – <code>wave_function.py</code> . . . . .	14
Oppgave 5.10: Bohrs atommodell – <code>bohres_model.py</code> . . . . .	14
Oppgave 5.11: Orbitalplott – <code>orbital_plot.py</code> . . . . .	15
<b>Oppgaver til kapittel 6</b>	<b>17</b>
Oppgave 6.1: Autotitrator – <code>autotitrator.py</code> . . . . .	17
Oppgave 6.2: Grunnstoffdata – <code>element_data.py</code> . . . . .	17

<b>Oppgaver til kapittel 7</b>	<b>19</b>
Oppgave 7.1: Grunnstoff – <code>element.py</code> . . . . .	19
Oppgave 7.2: Periodesystemet – <code>periodic_table.py</code> . . . . .	19
Oppgave 7.3: Elektronkonfigurasjon – <code>electron_configuration.py</code> . . . . .	19
Oppgave 7.4: Bindingstype – <code>binding_type.py</code> . . . . .	20
Oppgave 7.5: Molekyl – <code>molecule.py</code> . . . . .	20
Oppgave 7.6: Periodiske trender – <code>periodic_trends.py</code> . . . . .	20
Oppgave 7.7: Diffusjon – <code>diffusion.py</code> . . . . .	21
<b>Oppgaver til kapittel 8</b>	<b>23</b>
Oppgave 8.1: Radioaktiv stråling – <code>radioactive_radiation.py</code> . . . . .	23
Oppgave 8.2: Rene isotoper – <code>pure_isotopes.py</code> . . . . .	23
<b>Oppgaver til kapittel 9</b>	<b>25</b>
Oppgave 9.1: Enheter – <code>units.py</code> . . . . .	25
Oppgave 9.2: Størrelser – <code>quantities.py</code> . . . . .	25
<b>Oppgaver til kapittel E</b>	<b>27</b>
Oppgave E.1: Partikkelakselerator – <code>particle_accelerator.py</code> . . . . .	27
Oppgave E.2: Reaksjonshastighet – <code>reaction_rate.py</code> . . . . .	27
Oppgave E.3: Likevekt – <code>equilibrium.py</code> . . . . .	28
Oppgave E.4: Enzym – <code>enzyme.py</code> . . . . .	28

# Forord

Dette heftet er skrevet som et tilbud til bachelorstudentene i kjemi 2015 og 2016. De tidligere kjemistudentene hadde emnet programmeringsemnet *GEO-KJM1040 – Grunnkurs i programmering for problemstillinger i geofag og i kjemi*, men ble erstattet med *INF1100 – Grunnkurs i programmering for naturvitenskapelige anvendelser* på grunn av at det tidligere emnet ikke ga de nødvendige programmeringsferdighetene som trengs senere i studiet. Siden *INF1100* i utgangspunktet er beregnet for studenter på programmene *Elektronikk og datateknologi*, *Fysikk*, *astronomi og meteorologi*, *Matematikk og økonomi*, *Matematikk, informatikk og teknologi* og delvis *Materialer, energi og nanoteknologi*, har oppgavene liten tilknytning til kjemi.

Mange av oppgavene er basert på oppgaver fra et tilsvarende hefte for *GEO-KJM1040*, *GEO1040*, *Kjemirettede oppgaver* av Håkon Beckstrøm, Bård André Bendiksen og Espen Hagen Blokkdal og noen er inspirert av oppgaver fra *General Chemistry: The Essential Concepts* av Kenneth Goldsby og Raymond Chang.

Mange av tallene i oppgavene er funnet på siden det er vanskelig å finne reelle tall på noen av oppgavene.

Har du spørsmål om oppgavene, forslag til forbedringer eller andre henvendelser til dette heftet, ta kontakt med Sverre Løyland på email, du finner den på uio.no.



# Oppgaver til kapittel 1

## Oppgave 1.1: Glukose

På en vekt måler en kjemiker opp 3,243 g glukose,  $C_6H_{12}O_6$ , som skal brukes i en reaksjon.

a) Lag et Python-skript som beregner og printer den molare massen til glukose fra de molare massene til karbon, hydrogen og oksygen.

b) Modifiser skriptet ditt så den beregner og printer hvor mange mol glukose det er.

c) Modifiser skriptet ditt så den beregner og printer omtrent hvor mange atomer glukose det er.

Filnavn: `glucose.py`.

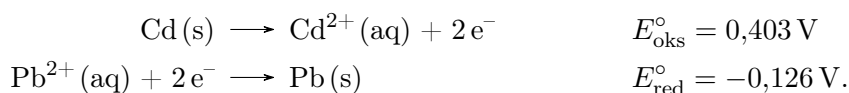
## Oppgave 1.2: Nernst likning

Nernst likning er gitt ved

$$E_{\text{celle}} = E_{\text{celle}}^{\circ} - \frac{RT}{neN_A} \ln Q$$

og beskriver cellepotensialet til en elektrokjemisk celle der  $E_{\text{celle}}$  er cellepotensialet,  $E_{\text{celle}}^{\circ} = E_{\text{oks}}^{\circ} + E_{\text{red}}^{\circ}$  er standard cellepotensialet,  $R = 8,31 \text{ J K}^{-1} \text{ mol}^{-1}$  er gasskonstanten,  $n$  er antallet elektroner som blir overført i redoksreaksjonen,  $e = 1,60 \cdot 10^{-19} \text{ C}$  er elementærladningen,  $N_A = 6,02 \cdot 10^{23} \text{ mol}^{-1}$  er Avogadros tall og  $Q$  er reaksjonskoeffisienten.

I en bestemt elektrokjemisk celle skjer halvreaksjonene



Bruk Python til å beregne og printe  $E_{\text{celle}}$  for en celle ved  $25^{\circ}\text{C}$  der konsentrasjonene av  $\text{Cd}^{2+}$  og  $\text{Pb}^{2+}$  er henholdsvis 0,0656 M og 0,192 M. Bruk variabler.

Filnavn: `nernst_equation.py`.

## Oppgave 1.3: Ideelle gasser

For en ideell gass gjelder den ideelle gassloven

$$pV = RnT,$$

der  $p$  er trykk,  $V$  er volum,  $n$  er stoffmengde,  $R = 8,314 \text{ J K}^{-1} \text{ mol}^{-1}$  og  $T$  er absolutt temperatur.

**a)** Finn stoffmengden til en idealgass som oppbevares i en ballong på 4 L under standardbetingelser\*. Regn ved hjelp av variabler.

**b)** Den tomme ballongen ble veid til 4,376 g før den ble fylt, mens den fylte ballongen ble veid til 11,45 g. Hva er den molare massen til gassen?

**c)** Hvilken gass tror du det er?

Filnavn: `ideal_gas.py`.

(Hentet fra *GEO-KJM1040*, oppgave 2.2.)

## Oppgave 1.4: Reelle gasser

Det molare volumet til en gass er definert som

$$V_m = \frac{V}{n} = \frac{M_m}{\rho}$$

der  $V_m$  er det molare volumet,  $V$  er volumet,  $n$  er stoffmengden,  $M_m$  er den molare massen og  $\rho$  er tettheten.

**a)** Beregn det molare volumet til  $\text{SF}_6$  med  $\rho = 6,602 \text{ g L}^{-1}$ .

**b)** Den ideelle gasslikningen på molar form er

$$pV_m = RT.$$

der  $p$  er trykk,  $R$  er gasskonstanten og  $T$  er absolutt temperatur. Bestem trykket til  $\text{SF}_6$  ved  $25^\circ\text{C}$  når man antar at det er ideell gass.

**c)** van der Waals likning på molar form er

$$\left(p + \frac{a}{V_m^2}\right)(V_m - b) = RT.$$

For  $\text{SF}_6$  er  $a = 7,857 \cdot 10^{-1} \text{ Pa} \cdot \text{m}^6/\text{mol}^2$  og  $b = 0,08786 \cdot 10^{-3} \text{ m}^3 \text{ mol}^{-1}$ . Bestem trykket.

**d)** Holder det å bruke idealgasslikningen eller må man bruke van der Waals likning?

Filnavn: `real_gas.py`.

(Hentet fra *GEO-KJM1040*, oppgave 2.4.)

---

\*International Union of Pure and Applied Chemistry (IUPAC) definerer standard trykk og temperatur som henholdsvis  $100\,000 \text{ Pa}$  og  $0^\circ\text{C}$ .



# Oppgaver til kapittel 2

## Oppgave 2.1: Energinivåer

Det  $n$ -te energinivået til et elektron i et hydrogenatom er gitt ved

$$E_n = -\frac{m_e e^4}{8\varepsilon_0^2 h^2} \cdot \frac{1}{n^2}$$

der  $m_e = 9,1094 \cdot 10^{-31}$  kg er elektronmassen,  $e = 1,6022 \cdot 10^{-19}$  C er elementærladningen,  $\varepsilon_0 = 8,8542 \cdot 10^{-12}$  C<sup>2</sup> s<sup>2</sup> kg<sup>-1</sup> m<sup>-3</sup> er den elektriske tomromspermittiviteten og  $h = 6,6261 \cdot 10^{-34}$  J s.

**a)** Lag et skript som printer energien til energinivåene for  $n = 1, \dots, 20$ .

Energien som frigjøres når et elektron går fra det  $n_i$ -te energinivået til det  $n_f$ -te energinivået er gitt ved

$$\Delta E = -\frac{m_e e^4}{8\varepsilon_0^2 h^2} \cdot \left( \frac{1}{n_i^2} - \frac{1}{n_f^2} \right).$$

**b)** Modifiser skriptet slik at det lager en fin tabell der cellen den  $i$ -te raden og  $f$ -te kolonnen viser energien som frigjøres når et elektron går fra det  $n_i$ -te energinivået til det  $n_f$ -te energinivået for  $i, f = 1, \dots, 5$ .

Filnavn: `energy_levels.py`.

## Oppgave 2.2: Elektronskall

Antallet elektroner som det er mulig å ha i det  $n$ -te elektronskallet er gitt ved

$$2n^2.$$

**a)** Lag et Python-skript som printer ut  $n$  og antallet elektroner det er plass til i det  $n$ -te fint formatert for  $n = 1, \dots, 7$ .

**b)** Det første skallet kalles K-skallet, det andre, L-skallet, det tredje, M-skallet og så videre. Modifiser skriptet ditt slik at navnet på det  $n$ -te skallet printes i stedet for  $n$ .

**Hint.** Du kan du konvertere en bokstav, for eksempel K, til et representativt tall med funksjonen `ord('K')` og tilbake med funksjonen `chr(75)`.

**c)** Elektronskall består av underskall som igjen består av orbitaler. Det  $n$ -te skallet består av  $n$  underskall. Det  $l$ -te underskallet består av  $2l + 1$  orbitaler der  $l = 0, \dots, n - 1$ . Hver orbital kan holde to elektroner. Modifiser skriptet ditt slik at den beregner antallet elektroner hvert skall kan holde fra informasjonen i denne oppgaven. Stemmer det overens med svaret i **a** og **b**?

Filnavn: `electrons_in_shell.py`.

# Oppgaver til kapittel 3

## Oppgave 3.1: Nernst likning som funksjon

Lag en funksjon `E_cell(E0_cell, n, Q, T=273.15)` som beregner cellepotensialet til en celle med standard cellepotensial `E0_cell` der `n` elektroner blir overført og reaksjonskoeffisienten er `Q` ved temperaturen `T`, se oppgave 1.2.

Test funksjonen ved å løse oppgave 1.2 og sammenlikne svarene.

Filnavn: `nernst_function.py`.

## Oppgave 3.2: Funksjon for energi i elektronskall

Lag en funksjon `DeltaE(ni, nf=None)` som returnerer energien som frigjøres når et elektron går fra det `ni`-te skallet i hydrogenatomet til det `nf`-te skallet. Dersom `nf` ikke spesifiseres skal funksjonen kun returnere energien til elektronet i det `ni`-te skallet. `ni` og `nf` skal være både tall og navnet til skallet, se oppgave 2.1.

Test funksjonen ved å putte inn noen verdier og sammenligne med oppgave 2.1.

Filnavn: `electron_shell_function.py`.

## Oppgave 3.3: pH-titrering

Når 250 mL 0,3 M eddiksyre titreres med 0,5 M natriumhydroksid er  $pH$ en gitt ved

$$pH = \begin{cases} \frac{1}{2} (pK_a - \lg c_a) & \text{hvis } n_b = 0 \\ pK_a + \lg \frac{n_b}{n_a - n_b} & \text{hvis } 0 < n_b < n_a \\ 14 + \frac{1}{2} \lg \left( \frac{c_a c_b}{c_a + c_b} K_b \right) & \text{hvis } n_b = n_a \\ 14 + \frac{1}{2} pK_a + \lg \frac{n_b - n_a}{V_a + V_b} & \text{hvis } n_b > n_a \end{cases}$$

der  $pK_a = 4.76$ ,  $n$  er stoffmengde,  $c$  er konsentrasjon og  $V$  er volum.  $a$  representerer syre og  $b$  representerer base.

Lag en funksjon `pH(Vb)` som returnerer  $pH$ en i løsningen der  $Vb$  er volum tilsatt base.

Test funksjonen ved å putte inn noen verdier som tilsvarer  $pH$  ved start, halvtitreringspunktet og ekvivalenspunktet.

Filnavn: `pH_titration.py`.

### Oppgave 3.4: Orbitalsannsynlighet

Sannsynligheten for å finne elektronet i 1s-orbitalen i hydrogenatomet i en avstand mindre enn  $x$  er gitt ved

$$\int_0^x r^2 \cdot \left( 2a_0^{-\frac{3}{2}} \cdot e^{-\frac{r}{a_0}} \right)^2 dr$$

der  $a_0 = 5,292 \cdot 10^{-11}$  m er Bohrradien.

**a)** Bruk trapesmetoden fra oppgave 3.6 i *A Primer on Scientific Programming with Python* til å bestemme medianavstanden, den avstanden der det er like sannsynlig å finne elektronet innenfor som utenfor.

**b)** Hvor langt unna kjernen må man for at det skal være 99 % sannsynlig å finne elektronet innenfor.

Filnavn: `orbital_probability.py`.

(Hentet fra *GEO-KJM1040*, oppgave 11.1.)

# Oppgaver til kapittel 4

## Oppgave 4.1: Forbedring av skript for Nernst likning

**a)** Lag en ny versjon av funksjonen i oppgave 1.2 der det sjekkes om temperaturen som oppgis er negativ. Hvis den er det, lag en `ValueError`.

**b)** Gjør sånn at skriptet kan kjøres fra kommandolinjen med parametrene `-E0`, `-n`, `-Q` og alternativt `-T`.

Test funksjonen ved å sammenlikne svarene med oppgave 1.2 og 3.1.  
Filnavn: `nernst_equation2.py`.

## Oppgave 4.2: Forbedring av skript for titrering

**a)** Lag en ny versjon av funksjonen i oppgave 3.3 der det sjekkes om  $V_b$  som oppgis er negativ. Hvis den er det, lag en `ValueError`.

**b)** Gjør sånn at skriptet spør om konsentrasjonen til syren. Hvis brukeren ikke gir meningsfylt input, spør igjen inntil det gis noe meningsfylt.

Test funksjonen ved å sammenlikne med svarene med oppgave 3.3.  
Filnavn: `pH_titration2.py`.



# Oppgaver til kapittel 5

## Oppgave 5.1: Statistikkfunksjoner

I denne oppgaven skal du lage dine *egne* funksjoner for å gjøre enkle statistiske beregninger. I alle deloppgaven skal du ta utgangspunkt i at du får en liste  $x_1, x_2, \dots, x_n$  med data som input.

**a)** Det aritmetiske gjennomsnittet er gitt ved

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Lag en funksjon `mean` som regner ut gjennomsnittet.

**b)** Den empiriske variansen til et datasett er gitt ved

$$V_x = \frac{1}{n} \sum_{i=1}^n (\bar{x} - x_i)^2.$$

Lag en funksjon `variance` som regner ut variansen.

**c)** Det empiriske standardavviket er gitt ved

$$s_x = \sqrt{V}.$$

Lag en funksjon `standard_deviation` som regner ut standardavviket.

**d)** Korrelasjonen mellom to datasett er gitt ved

$$r_{xy} = \frac{1}{n-1} \cdot \frac{1}{s_x s_y} \cdot \sum_{i=1}^n (\bar{x} - x_i) (\bar{y} - y_i).$$

Lag en funksjon `correlation` som regner ut korrelasjonen.

**e)** Relativt standardavvik er gitt ved

$$s_{x,r} = \frac{s_x}{\bar{x}}.$$

Lag funksjonen `relative_standard_deviation` som regner ut det relative standardavviket.

**f)** Funksjonsuttrykket for enkel, lineær regresjon mellom to datasett er

$$y = \alpha + \beta x,$$

der  $\alpha = \bar{y} - \beta \bar{x}$  og  $\beta = r_{xy}s_y/s_x$ . Lag en funksjon `simple_linear_regression` som gitt et datasett og en  $x$ -verdi regner ut en passende  $y$ -verdi.

**g)** Lag en funksjon `simple_linear_regression_plot` som plotter datasettene og lineærregresjonen.

Filnavn: `statistic_functions.py`.

## Oppgave 5.2: Stirlings approksimasjon

Boltzmanns formel for absolutt entropi er

$$S = k_B \ln W,$$

der  $S$  er entropi,  $k_B$  er Boltzmanns konstant og  $W$  er den statistiske vekten til en konfigurasjon. I forbindelse med utregning av  $W$  møter man ofte på uttrykket  $N!$ . En av grunnene til at  $S$  er definert med  $\ln$  er at det er lett å gjøre approksimasjoner av uttrykket  $\ln N!$ . En av disse approksimasjonene er *Stirlings approksimasjon* som er

$$\ln N! \approx N \ln N - N$$

der feilen blir mindre når  $N$  blir større.

**a)** Lag et program som plotter  $\ln N!$  og Stirlings approksimasjon i samme plot for  $N = 1, \dots, 100$ .

**b)** Plot feilen i Stirlings approksimasjon for  $N = 1, \dots, 100$ . 100 partikler er ikke mange, er det stort sett akseptabelt å bruke Stirlings approksimasjon?

Filnavn: `stirling.py`.

(Hentet fra *GEO-KJM1040*, oppgave 6.3.)

## Oppgave 5.3: Spektrofotometri

Man kan bruke kolorimetri for å bestemme konsentrasjonen til en løsning hvis løsningen har farge. Dette gjøres ved at man lager flere løsninger med kjent konsentrasjon, kalt standardløsninger, og måler hvor mye lys de absorberer. Så kan man gjøre lineærregresjon på dataene. Når man skal finne konsentrasjonen til en ukjent løsning, måler man absorpsjonen og kan ut fra regresjonen lese ut tilnærmet konsentrasjon.

Konsentrasjon og absorpsjon til seks standardløsninger måles til



$c \text{ [mg L}^{-1}\text{]}$	absorbans
0	0.001
0.2	0.184
0.4	0.346
0.6	0.465
0.8	0.686
1.0	0.727

- a)** Bruk funksjonene fra oppgave 5.1 til å lage en kalibreringskurve.
- b)** Korrelasjonen kvadrert sier noe om hvor bra en lineær regresjonslinje passer. Hvis  $r^2$  er nærme 1 passer den bra, men hvis den er nærme 0 passer den dårlig. Hva er  $r^2$ ?
- c)** Tre prøver fortynnes 250 ganger. Absorbansen måles til 0,495, 0,493 og 0,498. Regn ut konsentrasjonene i de tre prøvene. Hva er gjennomsnittet og standardavvikene?  
 Filnavn: `spectrofotometry.py`.  
 (Hentet fra *GEO-KJM1040*, oppgave 9.2.)

## Oppgave 5.4: pH-måling

De tre vanligste måten for å bestemme pH i en løsning er med pH-papir, titrering og pH-elektrode. En kjemiker måler pH-en i Coca Cola med de tre metodene, og hvert eksperiment blir gjennomført 100 ganger. Resultatene fra målingene er på [http://heim.ifi.uio.no/~inf1100/kjemi/pH\\_data.dat](http://heim.ifi.uio.no/~inf1100/kjemi/pH_data.dat).

- a)** Beregn gjennomsnittet for hvert av eksperimentene.
- b)** Beregn standardavviket for hvert av eksperimentene.
- c)** Beregn det relative gjennomsnittet for hvert av eksperimentene.
- d)** Kjemikeren godtar bare resultater der det relative standardavviket er mindre enn 5 %. Oppfyller noen av eksperimentene dette kravet?
- e)** Plot histogrammene til de tre eksperimentene med 20 stolper.  
 Filnavn: `pH_measuring.py`.  
 (Hentet fra *GEO-KJM1040*, oppgave 12.1.)

## Oppgave 5.5: Bjerrumplot

Likevektskonsentrasjonene til  $\text{CO}_2(\text{aq})$ ,  $\text{HCO}_3^-(\text{aq})$  og  $\text{CO}_3^{2-}(\text{aq})$  er avhengig av pHen i løsningen. Andelen av konsentrasjonene til hver er gitt ved

$$\begin{aligned} [\text{CO}_2(\text{aq})] &= \frac{[\text{H}^+(\text{aq})]^2}{[\text{H}^+(\text{aq})]^2 + K_1 [\text{H}^+(\text{aq})] + K_1 K_2}, \\ [\text{HCO}_3^-(\text{aq})] &= \frac{K_1 [\text{H}^+(\text{aq})]}{[\text{H}^+(\text{aq})]^2 + K_1 [\text{H}^+(\text{aq})] + K_1 K_2}, \\ [\text{CO}_3^{2-}(\text{aq})] &= \frac{K_1 K_2}{[\text{H}^+(\text{aq})]^2 + K_1 [\text{H}^+(\text{aq})] + K_1 K_2} \end{aligned}$$

der  $K_1 = 5,01 \cdot 10^{-7}$  og  $K_2 = 4,79 \cdot 10^{-11}$  er likevektskonstantene for reaksjonene  $\text{CO}_2 + \text{H}_2\text{O} \rightleftharpoons \text{H}^+ + \text{HCO}_3^-$  og  $\text{HCO}_3^- \rightleftharpoons \text{H}^+ + \text{CO}_3^{2-}$ .

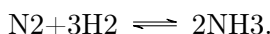
**a)** Plot konsentrasjonene av  $\text{CO}_2(\text{aq})$ ,  $\text{HCO}_3^-(\text{aq})$  og  $\text{CO}_3^{2-}(\text{aq})$  i samme vindu mot pHen. La  $\text{pH} \in [4, 12]$ .

**b)** Hvor skjærer kurvene til  $[\text{CO}_2(\text{aq})]$  og  $[\text{HCO}_3^-(\text{aq})]$  hverandre og hvor skjærer kurvene til  $[\text{HCO}_3^-(\text{aq})]$  og  $[\text{CO}_3^{2-}(\text{aq})]$  hverandre?

Filnavn: `bjerrum_plot.py`.

## Oppgave 5.6: Haber-Bosch-prosessen

Haber-Bosch-prosessen bruker for å fremstiller ammoniakk ved likevektsreaksjonen



Likevekter oppfyller likningene

$$\begin{aligned} \Delta_r G &= -RT \ln K_{\text{eq}} \\ \Delta_r G &= \Delta_r H - T \Delta_r S \end{aligned}$$

der  $R$  er gasskonstanten og  $T$  er temperaturen.

Data for temperatur og likevektskonstanter er i filen `http://heim.ifi.uio.no/~inf1100/kjemi/data\_121.dat`.

**a)** Beregn  $\Delta_r G$  for hver temperatur.

**b)** Plot  $\Delta_r G$  mot  $T$ .

**c)** Utfør lineærregresjon og bestem  $\Delta_r H$  og  $\Delta_r S$ .

d) Det viser seg at  $\Delta_r H = -92,22 \text{ kJ mol}^{-1}$ . Hva er den relative feilen? Kan du tenke deg hva den kommer av?

Filnavn: `Haber_Bosch.py`.

(Hentet fra *GEO-KJM1040*, oppgave 12.2.)

## Oppgave 5.7: Damptrykk

Alle kondenserte faser, det vil si væsker og faste stoffer, har *damptrykk*. Damptrykk er at molekyler flykter fra overflaten og blir til gass. Denne gassen skaper et trykk, damptrykket. Kokepunkt er definert som den temperaturen der damptrykket er lik trykket til omgivelsene. Damptrykket kan beskrives ved Clausius-Clapeyrons likning,

$$\underbrace{\ln P}_y = - \underbrace{\frac{\Delta_{\text{vap}} H}{R}}_a \cdot \underbrace{\frac{1}{T}}_x + \underbrace{C}_b$$

der  $P$  er damptrykket,  $\Delta_{\text{vap}} H$  er fordampningsentalpien,  $R$  er gasskonstanten,  $T$  er temperaturen og  $C$  er en konstant.

a) Last inn temperatur og damptrykk fra filen [http://heim.ifi.uio.no/~inf1100/kjemi/vap\\_pres.dat](http://heim.ifi.uio.no/~inf1100/kjemi/vap_pres.dat) og utfør lineærregresjon.

b) Hva er kokepunktet til Hg?

Filnavn: `vapor_pressure.py`.

(Hentet fra *GEO-KJM1040*, oppgave 14.3.)

## Oppgave 5.8: Aktiviseringsenergi

Ved hjelp av logaritmer kan Arrhenius likning skrives som

$$\underbrace{\ln k}_y = - \underbrace{\frac{E_a}{R}}_a \cdot \underbrace{\frac{1}{T}}_x + \underbrace{\ln A}_b$$

der  $k$  er reaksjonshastigheten,  $E_a$  er aktiveringsenergien,  $R$  er gasskonstanten,  $T$  er temperaturen og  $A$  er en konstant.

Nedbryting av lystgass,  $\text{N}_2\text{O}$ , til nitrogen og et oksygenatom er en andreordens reaksjon. Reaksjonshastigheten blir målt ved forskjellige temperaturer:

$k$ [ $\text{mol L}^{-1} \text{s}^{-1}$ ]	$t$ [ $^{\circ}\text{C}$ ]
$2,374 \cdot 10^{-3}$	600
$1,196 \cdot 10^{-2}$	650
$4,681 \cdot 10^{-2}$	700
$1,390 \cdot 10^{-1}$	750
$6,856 \cdot 10^{-1}$	800

**a)** Plot dataene og finn et uttrykk for den best tilpassede linjen .

**b)** Hva er  $E_a$  og  $A$ ?

Filnavn: `activation_energy.py`.

## Oppgave 5.9: Bølgefunksjon

For en bestemt partikkel er bølgefunksjonen gitt ved

$$\psi(x) = Ae^{-2|x|}e^{-3i}$$

der  $A$  er en konstant og  $i$  er den imaginære enheten ( $i^2 = -1$ ). For  $\psi$  skal være en gyldig bølgefunksjon, må

$$\int_{-\infty}^{\infty} \psi^*(x) \cdot \psi(x) \, dx = 1$$

der  $\psi^*$  er komplekskonjugatet av  $\psi$ . Beregn  $A$ .

Filnavn: `wave_function.py`.

## Oppgave 5.10: Bohrs atommodell

I denne oppgaven skal vi se på bevegelsen til et elektron rundt kjernen i Bohrs atommodell. Dette er en *svært* forenklet modell, men kan illustrere enkelte eksempler. Konvertering fra polare koordinater til kartesiske koordinater gjøres ved

$$\begin{aligned}x &= r \cos \theta, \\ y &= r \sin \theta\end{aligned}$$

der  $r$  er avstanden til origo og  $\theta$  er vinkelen fra  $x$ -aksen til punktet.

**a)** Plot en sirkel ved å holde  $r$  konstant og variere  $\theta$ .

**b)** I følge de Broglie oppfører elektroner seg som stående bølger som vil si at nullpunktene til bølgen ikke endrer seg. Legg inn en slik bølge ved å la  $r = r_0 + A \sin(n\theta)$  der  $r_0$  er gjennomsnittlig avstand til origo,  $A$  er amplituden og  $n$  er antall nullpunkter per omdreining.  $A$  bør være liten i forhold til  $r_0$ .

**c)** Hva skjer om  $n$  ikke er et heltall?

**Hint.** Prøv med flere omdreininger.

Filnavn: `bohrs_model.py`.

(Hentet fra *GEO-KJM1040*, oppgave 6.5.)

## Oppgave 5.11: Orbitalplott

Tverrsnitt av orbitalene til et atom med ett elektron kan lett plottes i polarkoordinater siden bølgefunksjonen i et bestemt plan er gitt ved

$$\psi(r, \theta) = R(r) \cdot Y_l(\theta).$$

Følgende tabell gir  $R$  og  $Y_l$  for noen utvalgte orbitaler.

Orbital	$R(r)$	$Y_l(\theta)$
$1s$	$\left(\frac{Z}{a_0}\right)^{\frac{3}{2}} \cdot 2 \cdot e^{-\frac{\rho}{2}}$	$\pi^{-\frac{1}{2}}$
$2p_0$	$\left(\frac{Z}{a_0}\right)^{\frac{3}{2}} \cdot \frac{1}{2\sqrt{6}} \cdot \rho \cdot e^{-\frac{\rho}{2}}$	$\left(\frac{3}{8\pi}\right)^{\frac{1}{2}} \cos \theta$
$3d_0$	$\left(\frac{Z}{a_0}\right)^{\frac{3}{2}} \cdot \frac{1}{9\sqrt{30}} \cdot \rho^2 \cdot e^{-\frac{\rho}{2}}$	$\left(\frac{5}{16\pi}\right)^{\frac{1}{2}} (3 \cos^2 \theta - 1)$

I tabellen er  $\rho = \frac{2Zr}{na_0}$ ,  $a_0 = 52,92$  pm,  $Z$  er atomnummer og  $n$  er skallnummer.

Plot  $1s$ -orbitalen til hydrogen,  $2p_0$ -orbitalen til karbon og  $3d_0$ -orbitalen til jern med `contourf`.

Filnavn: `orbital_plot.py`.

(Hentet fra *GEO-KJM1040*, oppgave 13.2.)



# Oppgaver til kapittel 6

## Oppgave 6.1: Autotitrator

En autotitrator er et instrument som kan gjøre titreringer, for eksempel pH-titreringer. En autotitrator har titrert 0,05 M NaOH mot en ukjent, enprotisk syre og skriver pH-verdiene og volum tilsatt base til en filen <http://heim.ifi.uio.no/~inf1100/kjemi/autotitrator.dat>. Vi definerer ekvivalenspunktet til det midten av intervallet der stigningen er størst og halvtitrerpunktet til midten av det intervallet der stigningen er minst.

- a) Hva er ekvivalenspunktet?
  - b) Hva er halvtitrerpunktet?
  - c) Hva er forskjellen mellom halvtitrerpunktet og halve ekvivalenspunktet?
  - d) Hva er  $pK_a$  til syren?
- Filnavn: `autotitrator.py`.

## Oppgave 6.2: Grunnstoffdata

Filen [http://heim.ifi.uio.no/~inf1100/kjemi/periodic\\_table.csv](http://heim.ifi.uio.no/~inf1100/kjemi/periodic_table.csv) inneholder data for alle grunnstoffene.

Lag et skript med funksjonen `element_data(element, data)` der `element` er et atomnummer og `data` er en tekststreng som tilsvarer en “overskrift” i datafilen som returnerer dataen for grunnstoffet.

Filnavn: `element_data.py`.





# Oppgaver til kapittel 7

## Oppgave 7.1: Grunnstoff

Lag en klasse `Element` med klassevariablene `a_num` som er atomnummeret, `sym` som symbolet, `name` som er navnet, `mass` som er atommassen i u, `e_config` som er elektronkonfigurasjonen, `ox_num` som er en liste av vanlige oksidasjonstall, `melt_pnt` som er smeltepunkt i K, `boil_pnt` som er kokepunkt i K og `el_neg` som er elektronegativiteten.

“Lag” karbon ved å lage en instans av klassen med riktig data som du kan finne det i et bra periodesystem eller på Internett.

Filnavn: `element.py`.

## Oppgave 7.2: Periodesystemet

Filen [http://heim.ifi.uio.no/~inf1100/kjemi/periodic\\_table.csv](http://heim.ifi.uio.no/~inf1100/kjemi/periodic_table.csv) inneholder data for grunnstoffene. De kommaseparerte kolonnene inneholder data for atomtall, symbol, engelsk navn, atommasse i u, smeltepunkt (K) og kokepunkt (K).

**a)** Lag en klasse `PeriodicTable` med metoden `__init__(self, src)`, der `src` er banen til filen `periodic_table_simple.csv`, som leser filen og lager instanser av klassen `Element` fra oppgave 7.1 for hvert grunnstoff.

**b)** Test at skriptet fungerer ved å printe ut de kjemiske forkortelsene til alle grunnstoffene.

Filnavn: `periodic_table.py`.

## Oppgave 7.3: Elektronkonfigurasjon

**a)** Lag et skript med funksjonen `e_config_long(a_num)` der `a_num` er et atomnummer som bruker skriptet i oppgave 7.2 til å returnere den utvidede elektronkonfigurasjonen, for eksempel  $1s^2 2s^2 2p^6 3s^2 3p^6 3d^6 4s^2$  i stedet for  $[Ar] 3d^6 4s^2$ .

**b)** Test at skriptet fungerer ved å printe ut den fullstendige elektronkonfigurasjonen til bly og sjekke at det stemmer.

Filnavn: `electron_configuration.py`.

## Oppgave 7.4: Bindingstype

Bindingstypen mellom to atomer kan grovt sett klassifiseres ut fra forskjellen i elektronegativitet,  $\Delta$ .

Forskjell i elektronegativitet	Type binding
$\Delta \geq 2,0$	ionisk
$2,0 > \Delta > 1,6$	mellom ionisk og polar kovalent
$1,6 \geq \Delta \geq 0,5$	polar kovalent
$0,5 > \Delta > 0,3$	mellom polar kovalent og upolar kovalent
$0,3 \geq \Delta$	upolar kovalent

**a)** Lag en funksjon `binding_type(element1, element2)` som tar to instanser av klassen `Element` fra oppgave 7.1 og returnerer en tekststreng med bindingstypen.

**b)** Hva slags binding er det i LiF, BrCl, mellom K og O i  $K_2O$  og mellom H og C i  $CH_4$ ?  
Filnavn: `binding_type.py`.  
(Hentet fra *GEO-KJM1040*, oppgave 10.1.)

## Oppgave 7.5: Molekyl

Lag en klasse `Molecule` med metoden `__init__(formula)` der `formula` er en tekststreng for molekylformelen (for eksempel ' $CH_3Br$ ') som har variabelen `molar_mass` som returnerer den molare massen.

Test skriptet ved å regne ut den molare massen til  $C_{12}H_{22}O_{11}$  og  $HCl$  ved hjelp av skriptet og for hånd.

Filnavn: `molecule.py`.

## Oppgave 7.6: Periodiske trender

**a)** Plot av elektronegativiteten mot atomnummer.

**b)** Gjør det samme som i **a**, men med antallet vanlige oksidasjonstall.

**c)** Gjør det samme som i **a**, men med kokepunktet.

**d)** Gjør det samme som i **a**, men med elektronegativiteten.

Filnavn: `periodic_trends.py`.

(Hentet fra *GEO-KJM1040*, oppgave 10.2 og 10.3.)

## Oppgave 7.7: Diffusjon

Bevegelsen til enkeltpartikler kan ofte beskrives som Brownske bevegelser eller virrevandring. På en vannoverflate legges 1 000 pollenkorn i et punkt. Bevegelsen til pollenkornene kan modelleres ved at for hvert sekund som går beveger hvert korn seg en tilfeldig avstand som er normalfordelt med forventning 0 mm og standardavvik 0,05 mm.

- a)** Lag video av pollenkornenes posisjon fra 0 til 100 sekunder.
- b)** Lag et plot av gjennomsnittlig avstand til origo kvadrert mot tid. Hva ser du?  
Filnavn: `diffusion.py`.



# Oppgaver til kapittel 8

## Oppgave 8.1: Radioaktiv stråling

Sannsynligheten for at en radioaktive kjerne henfaller i løpet av én tidsenhet kalles  $\lambda$ , og dens sammenheng med halveringstiden,  $T_{1/2}$ , er

$$\lambda \cdot T_{1/2} = \ln 2.$$

**a)** Lag et Python-skript som simulerer henfallet av 10 000 atomer  $^{99}\text{Mo}$  til  $^{99\text{m}}\text{Tc}^*$  og plotter hvor mye det er igjen av  $^{99}\text{Mo}$  og hvor mye  $^{99\text{m}}\text{Tc}$  som er produsert i løpet av én uke. Halveringstiden til  $^{99}\text{Mo}$  er 2,7489 dager.

**b)**  $^{99\text{m}}\text{Tc}$  henfaller igjen til  $^{99}\text{Tc}$  som er tilnærmet stabilt på denne tidsskalaen. Halveringstiden til  $^{99\text{m}}\text{Tc}$  er 6,0058 timer. Modifiser skriptet ditt slik at det plotter hvor mye  $^{99\text{m}}\text{Tc}$  det er mot tid.

**c)** Lag et plott av hvor mange desintegrasjoner det er per tidsenhet.  
Filnavn: `radioactive_radiation.py`.

## Oppgave 8.2: Rene isotoper

Til et eksperiment trengs det radioaktivt  $^{108}\text{Ag}$ , og for eksperimentet er det viktig at det ikke er noen andre radioaktive isotoper til stede.

$^{108}\text{Ag}$  kan produseres ved å bestråle naturlig sølv med nøytroner, men  $^{110}\text{Ag}$  produseres også siden naturlig sølv består av 52 %  $^{107}\text{Ag}$  og 48 %  $^{109}\text{Ag}$ . 50 g naturlig sølv bestråles i 30 min. Reaksjonshastigheten er

$$R = \sigma \phi N_T,$$

der  $\sigma_{^{107}\text{Ag}} = 37,65 \text{ b}$ ,  $\sigma_{^{109}\text{Ag}} = 90,26 \text{ b}$ ,  $\phi = 1 \cdot 10^{12} \text{ nøytroner s}^{-1} \text{ cm}^{-2}$  og  $N_T$  er antallet atomer av isotopet som bestråles.  $1 \text{ b} = 1 \cdot 10^{-24} \text{ cm}^2$ .  $^{108}\text{Ag}$  en halveringstid på 2,37 min og  $^{110}\text{Ag}$  har halveringstid på 24,56 s der begge henfaller til stabile isotoper av kadmium.

**a)** Plot mengden  $^{108}\text{Ag}$  og  $^{110}\text{Ag}$  mot tid fra starten av strålingen til det praktisk talt ikke er noe  $^{108}\text{Ag}$  eller  $^{110}\text{Ag}$  igjen.

**b)** Hvor lenge etter endt bestråling må man vente for mengden  $^{108}\text{Ag}$  er mer enn 99,9 % av mengden av radioaktive isotoper til stede?

Filnavn: `pure_isotopes.py`.

---

\*m-en betyr at technetiumet er *metastabilt* som betyr at kjernen er eksitert.



# Oppgaver til kapittel 9

## Oppgave 9.1: Enheter

Lag en modul `unit` med klassen `Unit` som kan gjøre enkel aritmetikk med enheter. Klassen skal initialiseres med metoden `__init__(self, baseunit)`. Klassen skal kunne gjøre følgende type aritmetikk.

```
$ python -i units.py
>>> J = kg * m**2 / s**2
>>> print J
m^2*kg*s^-2
```

Filnavn: `units.py`.

## Oppgave 9.2: Størrelser

Lag en module `quantity` med klassen `Quantity` som skal kunne gjøre enkel aritmetikk med tall med enheter. Klassen skal initialiseres med metoden `__init__(self, value, unit)`. Klassen skal kunne gjøre følgende type aritmetikk.

```
$ python -i quantities.py
>>> J = kg * m**2 / s**2
>>> print J
1.0 m^2*kg*s^-2
>>> R = 8.3145 * J / K / mol
>>> print R
8.3145 m^2*kg*s^-2*K^-1*mol^-1
>>> NA = mol**-1 * 6.0221e23
>>> kB = R / NA
>>> print kB
1.38066455223261e-23 m^2*kg*s^-2*K^-1
>>> a = 5.3 * mol; b = 1.5 * mol
>>> print a + b
6.8 mol
>>> print a - b
3.8 mol
>>> atm = 101325 * kg / m / s**2; L = 1e-3 * m**3
```

```
>>> R_alt = 0.082057 * L * atm / K / mol
>>> abs(R - R_alt) < 1e-3
True
```

Filnavn: quantities.py.



# Oppgaver til kapittel E

## Oppgave E.1: Partikkelakselerator

I en kjernereaktor kan man bestråle naturlig  $^{98}\text{Mo}$  med nøytroner, og det dannes  $^{99}\text{Mo}$ . Hastigheten til produksjonen er

$$R = \sigma \phi N_T,$$

der  $R$  er reaksjonshastigheten i atomer  $\text{s}^{-1}$ ,  $\sigma$  er tverrsnittet\* i  $\text{b} = 10^{-24} \text{cm}^2$ ,  $\phi$  er partikkelfluksen† i nøytroner  $\text{cm}^{-2} \text{s}^{-1}$  og  $N_T$  er antallet atomer som bestråles.

**a)** 5,00 g  $^{98}\text{Mo}$  bestråles av en nøytronståle med fluks lik  $10^{12}$  nøytroner  $\text{s}^{-1} \text{cm}^{-2}$  i én uke. Tverrsnittet til  $^{98}\text{Mo}$  er 412 mb. Lag et Python-skript som plotter hvor mye molybden-99 som er blitt produsert.

**b)**  $^{99}\text{Mo}$  er radioaktivt og henfaller til  $^{99\text{m}}\text{Tc}$  med en halveringstid på 2,7 dager. Lag et plot av hvor mye  $^{99}\text{Mo}$  som er til stede.

**c)** Etter bestrålingen skal produktet fraktes til et sykehus. Dette tar 2 dager. Lag et plot av hvor mye  $^{99}\text{Mo}$  som er til stede. Hvor mye  $^{99}\text{Mo}$  får sykehuset levert?

Filnavn: `particle_accelerator.py`.

## Oppgave E.2: Reaksjonshastighet

Syklopropan er ustabil og brytes ned til propen ved en førsteordens reaksjon. Ved  $500^\circ\text{C}$  er hastighetskonstanten  $6,7 \cdot 10^{-4} \text{s}^{-1}$ .

**a)** I en beholder er konsentrasjonen av syklopropan 0,25 M. Plot konsentrasjonen av syklopropan og propen mot tid over én time.

Hydrogen og nitrogenmonoksid reagerer og blir til nitrogen og vann ved en reaksjon tredjeordens reaksjon. Reaksjonen er førsteordens med hensyn på hydrogen og andreordens med hensyn på nitrogenmonoksid. Ved  $1280^\circ\text{C}$  er reaksjonshastigheten  $2,5 \cdot 10^2 \text{M}^{-2} \text{s}^{-1}$ .

**b)** I en beholder er det hydrogen og nitrogenmonoksid konsentrasjoner henholdsvis lik  $2,92 \cdot 10^{-2} \text{M}$  og 5,61 M. Plot konsentrasjonen av hydrogen, nitrogenmonoksid, nitrogen

---

\*Tverrsnittet til en kjerne sier noe om hvor lett den reagerer i kjernereaksjoner.

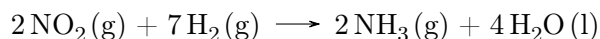
†Hvor mange partikler som strømmer gjennom et areal per tidsenhet.

og vann mot tid over 1 millisekund.

Filnavn: `reaction_rate.py`.

## Oppgave E.3: Likevekt

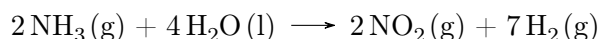
Reaksjonenhastigheten til reaksjonen



er gitt ved

$$r_f = k_f [\text{NO}_2]^2 [\text{H}_2]^7.$$

Reaksjonenhastigheten til den motsatte reaksjonen



er gitt ved

$$r_r = k_r [\text{NH}_3]^2.$$

Målinger viser at  $k_f = 9,71 \cdot 10^{-3} \text{ M}^{-8} \text{ s}^{-1}$  og  $k_r = 6,18 \cdot 10^{-2} \text{ M}^{-1} \text{ s}^{-1}$ .

I en lukket beholder er konsentrasjon til  $\text{NO}_2$ ,  $\text{H}_2$  og  $\text{NH}_3$  henholdsvis 1,32 M, 0,98 M og 1,06 M.

**a)** Lag er skript som plotter konsentrasjonene av stoffene over ett sekund.

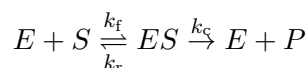
**b)** Plot  $[\text{NH}_3]^2 / ([\text{NO}_2]^2 [\text{H}_2]^7)$  mot tid.

**c)** Hva er likevektskonstanten for denne reaksjonen, og hvilken sammenheng har den med oppgave **b**?

Filnavn: `equilibrium.py`.

## Oppgave E.4: Enzym

I den enzymkatalyserte reaksjonen



er alle trinnene førsteordens med hensyn på konsentrasjonen til reaktantene i trinnet. For et bestemt enzym er  $k_f = 8,32 \cdot 10^4 \text{ M}^{-1} \text{ s}^{-1}$ ,  $k_r = 1,82 \cdot 10^{-2} \text{ s}^{-1}$  og  $k_c = 5,06 \cdot 10^{-2} \text{ s}^{-1}$ .

Til en beholder ble enzymer og substrater tilsatt slik at konsentrasjonene ble henholdsvis 1,22  $\mu\text{M}$  og 2,02  $\mu\text{M}$ .

Plot konsentrasjonen av substrat, produkt, enzym og enzym-substratkompleks mot tid i samme vindu.

Filnavn: `enzyme.py`.