

# **En enkel forsøringsmodell**

Prosjektoppgave i IN-KJM1900, Høst 2018

**Simen Kvaal og Christian Wilhelm Mohr**

**Revidert av Audun Skau Hansen**

Basert på en oppgave av Nils Christophersen og Are Magnus Bruaset



**UiO : Universitetet i Oslo**

Kjemisk Institutt  
Universitetet i Oslo  
2018

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>2</b>
1.1	Innledning . . . . .	2
1.2	Utføring og innlevering av oppgaven . . . . .	4
<b>2</b>	<b>Birkenes-modellen</b>	<b>5</b>
2.1	Hydrologisk delmodell . . . . .	5
2.2	Kjemisk delmodell . . . . .	9
<b>3</b>	<b>Delprosjekt I</b>	<b>13</b>
3.1	Oppgave 1: Newtons metode . . . . .	13
3.2	Oppgave 2: Innlesing av måledata . . . . .	16
3.3	Oppgave 3: Løsning av difflikninger med Eulers metode . . . . .	18
<b>4</b>	<b>Delprosjekt II</b>	<b>21</b>
4.1	Oppgave 4: Transport av vann . . . . .	21
4.2	Oppgave 5: Transport av sulfat . . . . .	25
4.3	Oppgave 6: Den fullstendige forsuringsmodellen . . . . .	27
<b>A</b>	<b>Eksempelsimuleringer til kontroll</b>	<b>29</b>

# Kapittel 1

## Introduksjon

### 1.1 Innledning

Forsuring av jord og ferskvann i Sør-Norge som følge av sur nedbør var et av de største miljøproblemene på 70- og 80-tallet. Sur nedbør fører til utlekking av labilt aluminium fra skogsjord til ferskvann. I høy nok konsentrasjon er labilt aluminium toksisk for fisk, da den feller ut i gjellene og kveler fisken. Fiskebestandene ble enten borte eller sterkt redusert i områder på til sammen over 36000 km<sup>2</sup> i våre fire sørligste fylker.

Det sure nedfallet i Sør-Norge var hovedsakelig fortynt svovelsyre (H<sub>2</sub>SO<sub>4</sub>), som dannes i atmosfæren når svoveldioksid (SO<sub>2</sub>) reagerer med vann, oksygen og frie radikaler. Svoveldioksidutslipp kommer hovedsakelig fra kullkraftverk. Mesteparten av det sure nedfallet kom fra store industriland i Europa, bl.a. Tyskland og Storbritannia. På grunn av stadig strengere regelverk i EU vedrørende utslipp, har det blitt en utfasing av kullkraftverk og teknisk forbedring i fjerning av svoveldioksid fra utslipp. Som resultat av dette er situasjonen langt bedre i Sør-Norge, med en nedgang på 70-80% ikke-marint sulfat i ferskvann. Dette har også medført en sterk reduksjon i konsentrasjonen av labilt aluminium i ferskvann<sup>1</sup>.

Årsakssammenhengen mellom sur nedbør, sur jord og surt overflatevann kan synes enkel, men et nærmere studium viser at forholdene er kompliserte. Skogsjord er naturlig sur og det skjer et vell av kjemiske prosesser fra vannet treffer vegetasjonen, renner gjennom jord, og til slutt ender opp i bekker og innsjøer. Betydningen av nedbørens kjemiske sammensetning er derfor ikke opplagt.

På syttitallet og begynnelsen av åttitallet pågikk en omfattende vitenskapelig debatt i Norge og andre land om hvilken rolle sur nedbør spiller for forsuringen av ferskvann<sup>2</sup>. I denne sammenhengen har matematiske modeller av hydrologiske og kjemiske proses-

---

<sup>1</sup>Ø. Garmo *et al.*, *Overvåkning av langtransportert forurenset luft of nedbør*, NIVA Årsrapport – Vann-kjemiske effekter (2013)

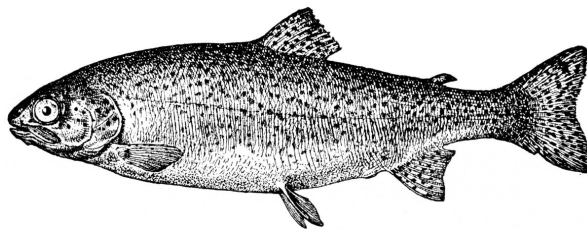
<sup>2</sup>N. Roll-Hansen, *Sur nedbør — et storprosjekt i norsk miljøforskning*. Fase I av “Sur nedbørs virkning på skog og fisk” (SNSF) 1972–1976, (NAVFs utredningsinstitutt, melding 4, Oslo, 1986), N. Roll-Hansen og G. Hestmark, *Miljøforskning mellom vitenskap og politikk*. En studie av forskningspolitikken omkring andre fase av storprosjektet “Sur nedbørs virkning på skog og fisk” (SNSF) 1976–1980, (NAVFs utredningsinstitutt, melding 2, Oslo, 1990).

ser i naturlige nedbørsfelter vært av betydning. (Uttrykket “hydrologi” dekker i denne sammenheng vannets strømningsmønster gjennom ulike jordlag fra det treffer bakken til det renner ut i bekken.) Modellene gjør det mulig å teste ut ulike antatte årsaksme-kanismer. Videre kan de (i den grad de er pålitelige) brukes til å forutsi virkningene av endret nedfall av sure komponenter.

Det foreligger en omfattende litteratur om vann- og jordforsuring særlig fra Europa og Nord-Amerika. Mye er publisert i form av artikler i vitenskapelige tidsskrifter og som rapporter. Når det gjelder forsuring i Norge gir de årlige overvåkingsrapportene finansiert av Miljødirektoratet en oversikt over hvordan situasjon er rundt omkring i landet og hvordan det endrer seg.

I denne oppgaven skal dere programmere en forenklet versjon av en modell som reproducerer vannføring og konsentrasjoner av viktige kjemiske komponenter i en bekk i et forsøksfelt — Birkenes — på Sørlandet. Modellen “drives” av nedbørsmengde, nedbørens kjemiske sammensetning og temperatur. Vi vil først introdusere modellen med de likningene som skal løses. Deretter vil programmeringsoppgavene følge fortløpende. Modellen representerer en *numerisk tilnærming*: likningene er for kompliserte til å løses eksakt på papiret (analytisk løsning), og istedet må man skrive dataprogrammer som for en stor del bruker *flyttall* – trunkerte representasjoner av reelle tall.

Birkenesmodellen blir i dag ansett som for enkel<sup>3</sup> da den ikke helt er i stand til å reproducere observerte korrelasjoner. Likevel består den av flere numeriske komponenter som inngår rutinemessig i moderne naturvitenskapelig forskning, slik som Newtons- og Eulers metode, og vil derfor kunne fungere som en nyttig introduksjon til beregningskjemi.



---

<sup>3</sup>Se N. Christophersen *et al.*, *Modelling the hydrochemistry of catchments: a challenge for the scientific method*, *Journal of Hydrology* **152** (1993), 1–12.

## 1.2 Utføring og innlevering av oppgaven

Prosjektoppgaven har to del-innleveringer, delprosjekt I og II med henholdsvis 3 og 2 ukers varighet. Frist for del I er **4. november klokka 23.59**, mens frist for del II er **18. november klokka 23.59**. Ytterligere rammer for prosjektet vil bli annonsert på kurssets semestersider.

Hver del-innlevering skal bestå av:

- En kort rapport med
  - Svar på tekstoppgaver
  - Plott og visualiseringer med tittle og aksetekst.
- Utskrift av eksempelkjøringer
- Kjørbar kildekode
- birkenes.data-filen

Koden skal være godt leselig og dokumentert. Det skal være kommentarer i koden som viser at dere forstår hva de ulike stegene gjør og hvorfor de virker som de skal.

Unngå mye printing i endelig innlevering, da det kan gjøre rettingen uoversiktlig.

I delprosjekt I skal flere verktøyfunksjoner implementeres og brukes videre i del II. Derfor er det nyttig å opprette en Python-modul med disse funksjonene. I delprosjekt II trenger man ideelt sett ikke endre modulen, kun importere og anvende.

I delprosjekt II skal du skrive Python-skript der verktøymodulen må importeres med en `import`-kommando sammen med andre moduler du måtte trenge. Du kan skrive ett skript for hele delprosjekt II om du ønsker dette, eller flere. Oppgavene er laget så de kan løses etter tur, og de gjenbraker data fra de forrige oppgavene på en naturlig måte.

*Lykke til!*

## Kapittel 2

# Birkenes-modellen

Den matematiske modellen for Birkenes-feltet består av en hydrologisk og en kjemisk delmodell. Den hydrologiske delen beskriver hvordan vannet beveger seg gjennom nedbørsfeltet som funksjon av tid. Når vannbevegelsene er bestemt, kan man legge til en kjemisk del som beregner konsentrasjonene av de viktigste kjemiske komponentene i jordvæske og overflatevann.

### 2.1 Hydrologisk delmodell

#### Reservoarer og strømmer

Den hydrologiske delmodellen er visualisert i Figur 2.1. Modellen består av to *reservoarer*, eller “bokser”, som representerer jordlag med ulike hydrologiske og kjemiske egenskaper. Disse boksene er koblet sammen med drenering, og boksene drenerer også ut i en bekk. Man kan tenke på reservoarene som beholdere med vann, og dreneringen som ledninger som forbinder disse.

De primære variablene for modellen er vannstanden  $A$  og  $B$  i reservoar **A** og **B** som funksjon av tiden  $t$ . De ytre påvirkningene som driver modellen er nedbøren  $P$  (“precipitation” på engelsk) og temperaturen  $T$ . Bekkens vannstrøm er  $Q$ .

La oss gå igjennom modellen. Vi beskriver først vannbevegelsen i reservoar **A**.

På ethvert tidspunkt er det en nedbørsmengde  $P$  som fyller det øverste reservoaret **A**. Så lenge vannstanden er under terskelen  $A_{\min}$  skjer det ingen avrenning derfra. Terskelverdien svarer til den fuktigheten jorda kan holde uten å gi fra seg vann. Overstiges  $A_{\min}$ , drenerer vannet ned i reservoaret **B** og til bekken. Dreneringen fra **A** er  $Q_A$ . Denne dreneringen går delvis over i **B** og delvis rett over i bekken, og variabelen  $A_{\text{sig}}$  bestemmer hvor mye av  $Q_A$ , som går til **B**-reservoaret.  $A_{\text{sig}}$  avhenger av vannstanden  $B$ .

Så over til **B**: Vannet som renner inn i dette reservoaret kommer fra **A** som beskrevet over. Når  $B$  overstiger terskelen  $B_{\min}$ , dreneres noe over i bekken,  $Q_B$ . Reservoaret **B** har også en øvre grense  $B_{\max}$  på vannstanden. Alt vann som overstiger denne grensen går direkte til bekken ( $Q_{\text{over}}$ ). (Tenk på dette som at beholderen renner over.)



Felles for begge reservoarene er at de kan miste vann på grunn av opptak gjennom vegetasjonen ( $E_A, E_B =$  evapotranspirasjon). Størrelsen på  $E_A$  og  $E_B$  avhenger av temperaturen  $T$  og vannstanden **A** og **B**.

Alle vannstrømmene eller vannfluksene ( $P, E, Q_A, Q_B, Q_{\text{over}}$  og  $Q$ , jf. figur 2.1) regnes i mm/dag, og vannstandene  $A$  og  $B$  i reservoarene regnes i mm. (Merk at 1 mm er lik 1 L/m<sup>2</sup>. Alle vannstrømmer og vannmengder uttrykkes derfor pr. flateenhet.) Døgnlige middeltemperatur,  $T$ , er i °C. Vannstrømmen i bekken er  $Q$ , og vi har

$$Q = (1 - A_{\text{sig}}) \cdot Q_A + Q_B + Q_{\text{over}}. \quad (2.1)$$

Vi skal nå utlede differensiallikningene som beskriver endringene i vannstandene  $A$  og  $B$  som funksjon av tid  $t$ .

### Massebalanse og tidsutvikling

Betrakt **A**-reservoaret over en kort tid  $\delta t$ . Over denne tiden endres vannmengden med  $\delta A$ . Vi antar at over tiden  $\delta t$  er vannfluksene  $P, E_A$  og  $Q_A$ , som bringer vann til **A**-reservoaret eller fjerner vann fra dette reservoaret, tilnærmet konstante. Vi kan da sette opp et "vannbudsjett" ( gjerne kalt en massebalanse) for **A**-reservoaret over tiden  $\delta t$ :

$$\delta A = \delta t \cdot P - \delta t \cdot E_A - \delta t \cdot Q_A. \quad (2.2)$$

For reservoaret **B** tenker vi tilsvarende, men vi ser bort fra  $Q_{\text{over}}$  i første omgang. I løpet av tidsrommet  $\delta t$  får vi

$$(\delta B)_0 = \delta t \cdot A_{\text{sig}} \cdot Q_A - \delta t \cdot E_B - \delta t \cdot Q_B. \quad (2.3)$$

Dersom  $B + (\delta B)_0 \leq B_{\text{max}}$ , så er  $Q_{\text{over}} = 0$ , og likningen over er det totale vannbudsjettet. Dersom  $B + (\delta B)_0 > B_{\text{max}}$ , så tømmes det overskytende vannet rett over i bekken,

$$\delta t \cdot Q_{\text{over}} = B + (\delta B)_0 - B_{\text{max}}. \quad (2.4)$$

Merk at  $Q_{\text{over}}$  er en strøm, så vi må multiplisere med  $\delta t$  for å få vannmengden. Det totale vannbudsjettet blir altså

$$\delta B = (\delta B)_0 - \delta t \cdot Q_{\text{over}} \quad (2.5a)$$

$$= \delta t \cdot A_{\text{sig}} \cdot Q_A - \delta t \cdot E_B - \delta t \cdot Q_B - \delta t \cdot Q_{\text{over}}, \quad (2.5b)$$

der

$$Q_{\text{over}} = \max[\delta t^{-1}(B - B_{\text{max}}) + A_{\text{sig}} \cdot Q_A - Q_B - E_B, 0]. \quad (2.6)$$

Vi ser altså på endringen i reservoar **B** i løpet av tidsrommet  $\delta t$  som en *tostegsprosess*, der tømming av overskytende vann over i bekken skjer etter de andre prosessene.<sup>1</sup>

<sup>1</sup>Dersom vi i stedet ser på massebalansen som en ettstegsprosess får vi det alternative og noe enklere uttrykket

$$Q_{\text{over}}^0 = \max[\delta t^{-1}(B - B_{\text{max}}), 0]. \quad (2.7)$$



La oss beskrive de ulike modellparametrene. Vi har her

$$Q_A = K_A \cdot \max(A - A_{\min}, 0), \quad K_A = 0.8 \text{ dag}^{-1}, \quad A_{\min} = 13 \text{ mm}, \quad (2.9a)$$

$$Q_B = K_B \cdot \max(B - B_{\min}, 0), \quad K_B = 0.045 \text{ dag}^{-1}, \quad B_{\min} = 40 \text{ mm}. \quad (2.9b)$$

Verdiene av konstantene er funnet ved tidligere tilpassing av beregnet avrenning fra feltet ( $Q$ ) til observert avrenning.

Parameteren  $A_{\text{sig}}$  er gitt som en stykkevis lineær funksjon. For  $B \leq B_{\min}$  går alt vannet fra  $A$ -reservoaret ned i  $B$ -reservoaret. Øker  $B$  over dette nivået går en større og større del av  $Q_A$  direkte til bekken,

$$A_{\text{sig}} = \begin{cases} 1 & B \leq B_{\min} \\ 1 - 0.25 \cdot \frac{B - B_{\min}}{B_{\max} - B_{\min}} & B_{\min} < B \leq B_{\max} \\ 0 & B_{\max} < B \end{cases}, \quad B_{\max} = 80 \text{ mm}. \quad (2.10)$$

Evapotranspirasjonen avhenger også av vannstandene i reservoarene og er gitt svært forenklet som

$$E_A = \begin{cases} 0.2 \cdot T & A > 1 \text{ mm} \\ 0 & \text{ellers,} \end{cases} \quad (2.11)$$

$$E_B = \begin{cases} 0.2 \cdot T & A \leq 1 \text{ mm og } B_{\min} < B \leq B_{\max} \\ 0 & \text{ellers.} \end{cases}$$

La nå  $t_j = j \cdot \delta t$ , med  $j = 0, 1, 2, \dots$  et heltall. Vi deler altså opp tidsaksen i mange like store intervaller med lengde  $\delta t$ . Vi ønsker å bestemme vannstanden  $A$  og  $B$  på tidspunktene  $t_j$ , og vi skriver

$$A_j = A(t_j), \quad B_j = B(t_j), \quad (2.12)$$

og tilsvarende for andre størrelser som er funksjon av  $t$ . Med vannbudsjettene (2.2) og (2.5b) får vi formlene

$$A_{j+1} = A_j + \delta t \cdot P_j - \delta t \cdot E_{A,j} - \delta t \cdot Q_{A,j}. \quad (2.13a)$$

$$B_{j+1} = B_j + \delta t \cdot A_{\text{sig},j} \cdot Q_{A,j} - \delta t \cdot E_{B,j} - \delta t \cdot Q_{B,j} - \delta t \cdot Q_{\text{over},j}. \quad (2.13b)$$

Vannstanden  $A_{j+1}$  og  $B_{j+1}$  ved  $t_j + \delta t$  er altså en funksjon av vannstandene ved  $t_j$ . Likningene (2.13) er de sentrale likningene som skal programmeres for den hydrologiske delen.

Et tredje alternativ er uttrykket

$$Q_{\text{over}}^1 = \begin{cases} Q_A \cdot A_{\text{sig}} - Q_B - E_B & B \geq B_{\max} \text{ og } Q_A \cdot A_{\text{sig}} \geq Q_B + E_B \\ 0 & \text{ellers} \end{cases} \quad (2.8)$$

Dette ble brukt i de publiserte simuleringene av birkenesmodellen, og er inkludert her kun til informasjon. Det samme gjelder uttrykket (2.7). Den ivrige studenten står fritt til å eksperimentere, men dette er ikke påkrevd.

Tidsintervallet  $\delta t$  var over vilkårlig. Dividerer vi vannbudsjettene med  $\delta t$  på begge sider og lar  $\delta t$  gå mot null, får vi differensiallikningene

$$\frac{dA}{dt} = P - E_A - Q_A \quad (2.14a)$$

$$\frac{dB}{dt} = Q_A \cdot A_{\text{sig}} - E_B - Q_B - Q_{\text{over}}. \quad (2.14b)$$

Likningene (4.2) representerer *Eulers metode* for difflikningene (2.14a) og (2.14b). For  $\delta t$  tilstrekkelig liten burde vi intuitivt få en tilnærming til de eksakte løsningene til disse kontinuerlige likningene.

Når vi løser differensiallikningene (2.14a) og (2.14b) med Eulers metode kan vi samtidig beregne vannføringen  $Q$  i bekken etter formelen (2.1). Denne kan sammenlignes med den observerte avrenningen  $Q_{\text{obs}}$ . For å kunne løse likningene og beregne avrenningen trengs døgnlig nedbør og døgnlig middeltemperatur som ligger på filen `birkenes.data`. Her finnes også målt døgnlig avrenning  $Q_{\text{obs}}$ . Denne kan sammenliknes med den beregnede avrenningen  $Q$ .

## 2.2 Kjemisk delmodell

### Begreper og bakgrunn

Alle kjemiske komponenter i denne oppgaven er uorganiske ioner. Ioner er kjemiske stoff med en eller flere elektriske elementærladninger (positive eller negative). Vi opererer med følgende typer ioner der den kjemiske formelen står i parentes: sulfat ( $\text{SO}_4^{2-}$ ), bikarbonat ( $\text{HCO}_3^-$ ), hydronium ( $\text{H}^+$ ), kalsium ( $\text{Ca}^{2+}$ ) og aluminium ( $\text{Al}^{3+}$ ).

Dette er en sterk forenkling av den vannkjemiske sammensetningen, men er greit nok for denne enkle hydrokjemiske forsyningsmodellen.

Det såkalte elektronøytralitetsprinsippet sier at i enhver mengde vann vil summen av de negative elektriske ladningene være lik summen av de positive. Konsentrasjonen av ionene (angitt som  $[\text{H}^+]$  etc.) bestemmes i mol/L, der 1 mol tilsvarer Avogadros tall ( $6.022 \cdot 10^{23}$ ). Elektronøytralitetsprinsippet gir da for en vilkårlig vannmengde:

$$[\text{H}^+] + 2[\text{Ca}^{2+}] + 3[\text{Al}^{3+}] = 2[\text{SO}_4^{2-}] + [\text{HCO}_3^-] \quad (2.15)$$

Det er en høy konsentrasjon av ionene  $\text{H}^+$  og  $\text{Al}^{3+}$  som kan gjøre ferskvann giftig for fisk og andre organismer. Svært grovt kan en si at konsentrasjonene av disse ionene bør være  $\text{H}^+ < 2 \cdot 10^{-5}$  mol/L og  $\text{Al}^{3+} < 4 \cdot 10^{-6}$  mol/L for at ørret skal kunne overleve. Mye kalsium kan gjøre at fisken tåler mer  $\text{H}^+$  og  $\text{Al}^{3+}$ , mens sulfat og bikarbonat i seg selv betyr lite.

### Sulfattransport

Som beskrevet i innledningen er "sur nedbør" i denne sammenheng fortynt svovelsyre ( $\text{H}_2\text{SO}_4$ ) som kommer ned med nedbøren. Vi antar at  $\text{SO}_4^{2-}$  ikke inngår i kjemiske reaksjoner med jord og vegetasjon, noe som er en meget sterk forenkling. Sulfat antas da bare å følge vannet gjennom feltet med det unntak at dette ionet ikke tas opp av

vegetasjonen ved evapotranspirasjon. Antar vi at hvert reservoar er ideelt blandet med hensyn på sulfat, leder en massebalansebetraktning (altså et "sulfatbudsjett") til følgende masseregnskap for sulfatmassene  $M_A$  og  $M_B$  i reservoarene **A** og **B**, henholdsvis:

$$\delta M_A = \delta t \cdot P \cdot C_P - \delta t \cdot Q_A \cdot C_A \quad (2.16)$$

$$\delta M_B = \delta t \cdot A_{\text{sig}} \cdot Q_A \cdot C_A - \delta t \cdot (Q_B + Q_{\text{over}}) \cdot C_B \quad (2.17)$$

Her er  $C_P$  er konsentrasjonen av sulfat i nedbøren, og  $C_A$  og  $C_B$  konsentrasjonen av sulfat i reservoarene **A** og **B** (det vil si  $C_A = M_A/A$  og  $C_B = M_B/B$ ). Sulfatmengde regnes her i mol/m<sup>2</sup>, mens sulfatkonsentrasjonen er i mol/L.

Lar vi  $\delta t \rightarrow 0$  får vi følgende differensiallikning for massetransport:

$$\frac{d}{dt} M_A = P \cdot C_P - Q_A \cdot C_A \quad (2.18)$$

$$\frac{d}{dt} M_B = A_{\text{sig}} \cdot Q_A \cdot C_A - (Q_B + Q_{\text{over}}) \cdot C_B. \quad (2.19)$$

Simulert konsentrasjon av sulfat i avrenningen blir:

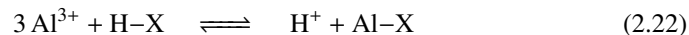
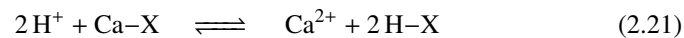
$$C_Q = \frac{1}{Q} [(1 - A_{\text{sig}}) \cdot Q_A \cdot C_A + (Q_B + Q_{\text{over}}) \cdot C_B]. \quad (2.20)$$

Her beregnes sulfatkonsentrasjonen som en veiet sum av konsentrasjonene i **A** og **B** ved tidspunkt der  $Q > 0$ . Når  $Q = 0$ , setter vi  $C_Q = 0$ .<sup>2</sup>

Studer uttrykkene og overbevis deg selv om at du forstår det. For å løse differensiallikningene for sulfat og beregne konsentrasjonen i avrenningen trengs døgnlig  $C_P$  som ligger på filen `birkenes.data` (under kolonnen 'cps04').

### Ionebyttereaksjoner

Når det gjelder de positivt ladede ionene — kationene — antar vi det motsatte av det vi gjorde for sulfat, nemlig at konsentrasjonene her er fullstendig kontrollerte av kjemiske likevektsreaksjoner i nedbørsfeltet. For begge reservoarene er disse likevektsreaksjonene antatt å kun være kationbyttereaksjoner mellom jordpartiklene og jordvæsken:



Den venstre siden i (2.21) representerer to  $\text{H}^+$ -ioner i jordvæsken som byttes ut med et kalsiumion som sitter på den faste fasen i jorda (symbolisert med  $\text{Ca-X}$ ). Resultatet (høyre side) er at kalsiumionet går ut i jordvæsken mens  $\text{H}^+$ -ionene blir sittende på den faste fasen. Ionebytte mellom  $\text{H}^+$  og  $\text{Al}^{3+}$  (2.22) tolkes på tilsvarende måte. Symbolet  $\rightleftharpoons$  betyr at reaksjonen kan gå i begge retninger. Når reaksjonen er i likevekt (egentlig

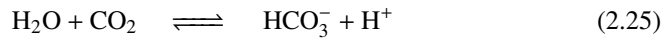
<sup>2</sup>I realitet er konsentrasjonen ikke 0, men udefinert. Det vil si at det er ikke snakk om at konsentrasjonen er under deteksjonsgrense, men at det er fysisk ikke mulig å ta en "måling" av konsentrasjonen fordi det ikke er noe vann. Vi skal se bort fra dette i denne prosjekt oppgave, men det er noe man må være klar over dersom det skulle utføres videre statistisk analyse.

dynamisk likevekt) betyr det at “forover”- og “bakover”-reaksjonen går med samme hastighet. For å beregne konsentrasjonene i likevekt trenger vi likevektskonstanter:

$$K_{HCa} = \frac{[H^+]^2}{[Ca^{2+}]} \quad (2.23)$$

$$K_{AlH} = \frac{[Al^{3+}]}{[H^+]^3} \quad (2.24)$$

Sulfat er ikke den eneste anion vi må ta hensyn til. Konsentrasjonen av bikarbonat ( $HCO_3^-$ ) er avhengig av konsentrasjonen av  $H^+$ :



Denne reaksjonen er ikke en ionebytte-reaksjon, men en reaksjon mellom løst  $CO_2$ -gass i jordvæsken og vann. Likevekten bestemmes av konstanten  $K_H$ :

$$K_H = [H^+][HCO_3^-] \quad (2.26)$$

Det er mer løst  $CO_2$  i jordvæsken enn i overflatevann. Dette skyldes respirasjon fra mikroorganismer og røtter i jorda. Når jordvæsken kommer ut i atmosfæren, vil  $CO_2$  fordampe, og reaksjon (2.25) vil gå mot venstre.

For at vi skal kunne beregne konsentrasjonen av ionene  $H^+$ ,  $Al^{3+}$ ,  $Ca^{2+}$  og  $HCO_3^-$  i avrenningen på lik måte som for sulfat (2.20), er vi først nødt til å beregne konsentrasjonene i reservoarene. Avhengig av konsentrasjonen av sulfat i reservoarene ved et gitt tidspunkt, vil konsentrasjon av resterende ioner måtte bestemmes ved likevekt. Dette gjøres først ved å ta hensyn til elektronøytralitetsprinsippet (2.15). Som likningen står kan den ikke løses fordi den inneholder flere ukjente. Likningen kan derimot løses hvis vi skriver om slik at den har kun en ukjent. Alle likevektslikninger (2.23, 2.24 og 2.26) kan løses med hensyn på  $[H^+]$ . Dermed kan elektronøytralitetslikningen (4.15) omskrives ved å erstatte  $[Ca^{2+}]$ ,  $[Al^{3+}]$  og  $[HCO_3^-]$ :

$$[H^+] + 2\frac{[H^+]^2}{K_{HCa}} + 3K_{AlH}[H^+]^3 = 2[SO_4^{2-}] + \frac{K_H}{[H^+]} \quad (2.27)$$

Vi kan skrive dette som:

$$f(x) = 3K_{AlH}x^3 + 2K_{HCa}^{-1}x^2 + x - 2[SO_4^{2-}] - K_Hx^{-1} = 0. \quad (2.28)$$

For en gitt verdi av  $[SO_4^{2-}]$  kan man løse likningen  $f(x) = 0$  numerisk med f.eks. Newtons metode.

Når konsentrasjonen av  $H^+$  er beregnet for et gitt sulfatkonsentrasjon kan de andre ionene beregnes lett med likevektslikningene (2.23) (2.24) og (2.26).

Konsentrasjonen av kalsium i avrenningen beregnes ved å veie konsentrasjonene i **A**- og **B**-reservoarene med avrenningen fra disse reservoarene (som for sulfat, likning (2.20)). Konsentrasjonene  $[H^+]$ ,  $[Al^{3+}]$  og  $[HCO_3^-]$  i avrenningen beregnes først som for kalsium, men må deretter justeres for å ta hensyn til at  $CO_2$  damper av etter at jordløsningen er blitt til overflatevann. Når  $CO_2$  damper av forskyves likningen (2.25)

til venstre og konsentrasjonen  $[H^+]$  og  $[HCO_3^-]$  synker. Når  $[H^+]$  synker forskyves likning 2.22 til høyere. Selv om det ikke lenger er snakk om jordvæske, så finnes det jordpartikler i avrenningen som vil binde til seg  $Al^{3+}$  ved samme ionebyttereaksjon (2.22). Videre vil en økning i pH gjøre  $Al^{3+}$  om til  $AlOH^{2+}$ . Dette går utover prosjekt-oppgaven da aluminiums spesiering i miljøet er kompleks. Vi må derfor bare akseptere at konsentrasjonen av  $Al^{3+}$  må justeres for å oppnå elektronøytralitet i løsningen. Justeringen i avrenningen løses på likt måte som for reservoarene med Newtons metode, men her holdes  $[Ca^{2+}]$  og  $[SO_4^{2-}]$  konstant:

$$[H^+] + 2[Ca^{2+}] + 3K_{AlH}[H^+]^3 = 2[SO_4^{2-}] + \frac{K_H}{[H^+]} \quad (2.29)$$

$$g([H^+]) = 3K_{AlH}[H^+]^3 + [H^+] + 2([Ca^{2+}] - [SO_4^{2-}]) - K_H[H^+]^{-1} = 0 \quad (2.30)$$

Som for sulfat må vi fortsatt sette konsentrasjonene i avrenningen til null ved tidspunkt der  $Q = 0$ . Ytterligere detaljer beskrives i programmeringsoppgavene under.

# Kapittel 3

## Delprosjekt I

I delprosjekt I skal du utvikle en Python-modul, som vi her kaller `birkutil` for enkelhets skyld. Du står fritt til å kalle denne modulen noe annet om du ønsker. Du skal også anvende denne modulen til å løse øvrige tekstoppgaver.

### 3.1 Oppgave 1: Newtons metode

#### Om metoden

I mange situasjoner har vi en likning på formen:

$$\text{finn } x \text{ slik at } f(x) = 0,$$

hvor  $f$  er en glatt (flere ganger deriverbar) funksjon  $f : [a, b] \rightarrow \mathbb{R}$ . Dersom  $f$  er, f.eks., en lineær eller kvadratisk funksjon kan vi lett finne løsningene, men for mer kompliserte funksjoner må vi ty til numeriske teknikker.

En populær metode for dette problemet er *Newtons metode*. Dette er en iterativ prosedyre og basert på følgende intuitive bilde, se figur 3.1: Gitt et startgjett  $x_0$  for  $x$ , konstruer tangenten til  $f$  i  $x_0$ ,

$$f(x) \approx f(x_0) + f'(x_0) \cdot (x - x_0).$$

Dersom  $x_0$  er tilstrekkelig nær den eksakte roten  $x$ , så bør punktet der tangenten skjærer  $x$ -aksen være en bedre approksimasjon enn  $x_0$ . Dette punktet er gitt ved

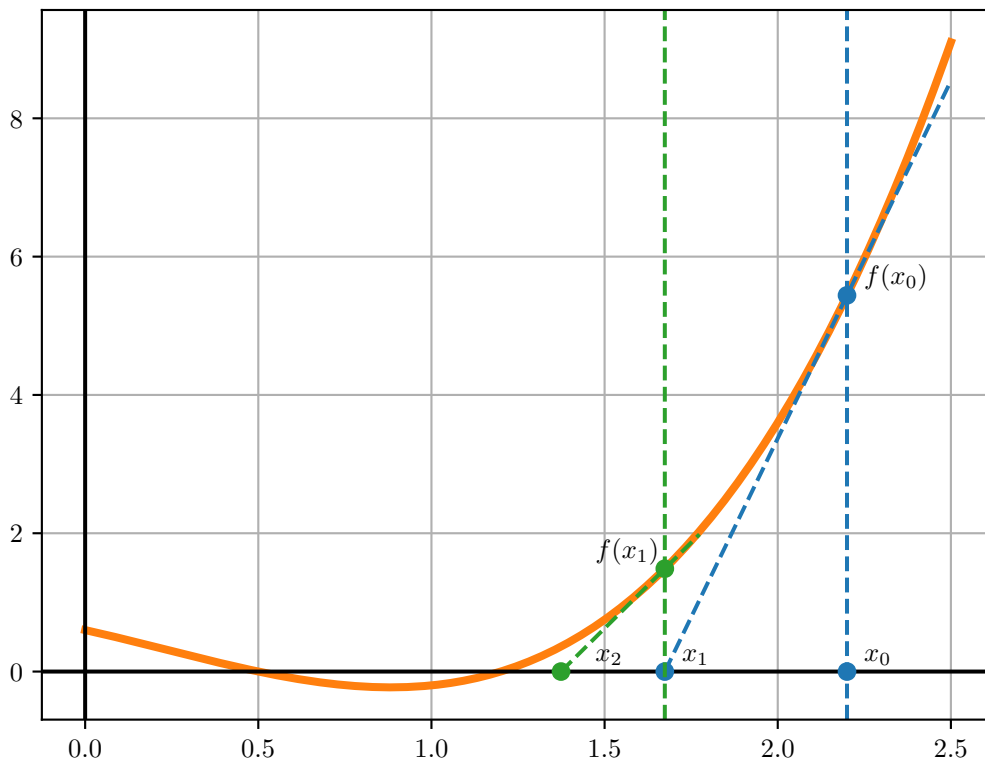
$$x_1 = x_0 - f'(x_0)^{-1} f(x_0).$$

Dette definerer den grunnleggende iterasjonen in Newtons metode: Etter å ha beregnet  $x_k$ , beregner vi  $x_{k+1}$  på samme måte, etter formelen

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k).$$

Vi avslutter iterasjonen når tilstrekkelig presisjon er nådd, det vil si når

$$|f(x_k)| < \epsilon,$$



Figur 3.1: Illustrasjon av Newtons metode

der  $\epsilon$  er en toleranseparameter vi står fritt til å velge. Det er viktig å merke seg at vi kan ikke sette  $\epsilon = 0$ , ettersom alle beregninger med flyttall er approksimative. Typiske verdier er  $\epsilon = 10^{-12}$ .

Noen ganger konvergerer ikke Newtons metode. Da må vi sikre at iterasjonen stopper likevel, for eksempel ved å terminere etter et visst antall iterasjoner `maxit`. Det kan også hende at  $f'(x_k) = 0$ . Da er ikke  $x_{k+1}$  veldefinert, og vi må også terminere.

Newton's metode konvergerer veldig raskt når den først konvergerer. Grovt sagt, dersom  $n$  siffer er riktig etter  $k$  iterasjoner, så vil rundt  $2n$  siffer være riktig etter  $k + 1$  iterasjoner. Dette kalles "kvadratisk konvergens".

### Oppgave 1a: Implementering av Newtons metode

**Implementering.** Du skal levere en Python-funksjon i verktøymodulen som implementerer Newtons metode for likningen  $f(x) = 0$ , der den deriverte funksjonen  $f'(x)$  antas kjent. Funksjonen skal oppfylle følgende:

- Funksjonen skal returnere den siste approksimative roten funnet ved iterasjon

inntil tilstrekkelig nøyaktighet er oppnådd eller til et maksimalt antall iterasjoner er tatt.

- Funksjonen skal ta en funksjon og dens deriverte som parametre samt et startgjett, et maksimalt antall iterasjoner og en toleranseparameter.
- Funksjonen skal kunne gjenkjenne det singulære tilfellet  $f'(x_k) = 0$  og terminere ved å kaste en exception av type `ValueError`. Funksjonen skal også kaste en exception dersom antall `maxit` iterasjoner er nådd uten konvergens.
- Funksjonen skal ha en docstring og ellers skrives på en lett forståelig måte med kommentarer.

Det er viktig å tenke på at funksjonen skal inngå i en større sammenheng, det vil si at den skal være lett å anvende på ulike problemer. Et eksempel på et funksjonskall kan være

```
1 from birkutil import newton_solve
2 (x, n) = newton_solve(f, df, x0, maxit = 50, epsilon = 1e-8)
```

**Testing.** Skriv en test-funksjon i en testfil: opprett en `.py`-fil med navn som starter på `test_`, f.eks. `test_newton.py`. Filen skal inneholde en eller flere funksjoner med navn som starter på `test_`. Se ellers læreboka/forelesningene om testfunksjoner.

Testfunksjonene skal teste implementeringen av Newtons metode på  $f(x) = x^2 - 4$ , med initialverdier  $x_0 = -1$  og  $1$ , og også  $0$ . Newton-iterasjonen skal konvergere til  $-2$  og  $2$  for de to første tilfellene, og for det siste tilfellet skal du sjekke at funksjonen kaster en `ValueError`, siden  $f'(0) = 0$ .

Bruk `maxit = 50` og `epsilon = 10-12` i testfunksjonen. Se for øvrig forelesningene/læreboka om *unit testing* og *exception handling*.

### Oppgave 1b: Newtons metode i Birkenesmodellen

I denne oppgaven skal du levere et skript som studerer elektronøytralitetsbetingelsen i likning (2.15), som er på formen  $f(x) = 0$  med

$$f(x) = 3K_{AlH}x^3 + 2K_{HCa}^{-1}x^2 + x - 2s - K_Hx^{-1}. \quad (3.1)$$

Her er  $s = [SO_4^{2-}]$ , og er en parameter til likningen på samme måte som likevektskonstantene. I birkenesmodellen trenger vi å finne konsentrasjonen  $x = [H^+]$  for ulike kombinasjoner av likevektskonstanter og verdier av  $s$ . Da trenger vi å vite hva som er en rimelig verdi for startgjettet  $x_0$  som vi kan bruke i alle tilfeller. Det er også nyttig å ha en følelse med hvilket intervall  $x$  kan regnes med å befinne seg.

Vi kan anta at  $s$  ligger i intervallet  $[2 \cdot 10^{-6}, 200 \cdot 10^{-6}]$ . Likevektskonstantene tar i reservoar **A** verdiene  $K_{AlH} = 10^9$ ,  $K_{HCa} = 10^{-2.2}$ , og  $K_H = 2.5 \cdot 10^{-10}$ . I **B** har vi endringen  $K_{HCa} = 10^{-3.2}$ , men ellers de samme verdiene.

Siden  $x$  er en konsentrasjon, er den alltid positiv.



**Studie av funksjonen  $f(x)$ .** For hvert reservoar: Lag et plott av  $f(x)$  for 4 representative valg av  $s$  i samme koordinatsystem. (Altså 2 plott med 4 funksjoner i hver.) Alle linjene skal ha merkelapper (legend). La  $x$  variere i et passende positivt intervall, slik at skjæringspunktene med  $x$ -aksen er tydelige i alle tilfeller. (Finn dette intervallet ved prøving og feiling. Bruk samme intervall for begge reservoar.)

Finn et rimelig valg for  $x_0$  uavhengig av reservoar og  $s$ . Test valget ditt av  $x_0$  ved å rekne ut  $x_*$  slik at  $f(x_*) = 0$  med Newtons metode. (Vi bruker her symbolet  $x_*$  for å skille det fra  $x$  som her er en aksevariabel.) Gjør dette for alle verdiene av  $s$  du valgte, og begge reservoar. Testen gjøres visuelt ved å plote punktene  $x = x_*, y = f(x_*)$  i samme aksystem som  $f(x)$ . (I alt 8 punkter.)

**Implementering av modulfunksjon.** Skriv en funksjon som løser  $f(x) = 0$  gitt verdier på de ulike parameterne. Funksjonen skal legges i verktøymodulen. Funksjonen skal tilfredsstillende følgende:

- Funksjonen skal bruke implementeringen av Newtons metode fra forrige deloppgave. Bruk  $x_0$  som du fant over.
- Funksjonen ta likevektskonstantene og sulfatkonsentrasjonen  $s$  som parametre.
- Funksjonen skal kaste en `ValueError`-exception dersom det endelige svaret  $x$  er negativt (alle konsentrasjoner er positive).
- Funksjonen skal ha en docstring og ellers være pent og oversiktlig skrevet.

For eksempel kan et funksjonskall se slik ut:

```
1 from birkutil import finn_konsentrasjon
2 x = finn_konsentrasjon(s, K_AlH, K_HCa, K_H)
```

**Testing.** Du skal skrive en testfunksjon/testfil for implementeringen. For reservoar **A** skal du teste at  $s = 2 \cdot 10^{-6}$  gir  $x = 1.38919609145 \cdot 10^{-5}$ , og for reservoar **B**,  $x = 1.37581974866 \cdot 10^{-5}$ .

**Plott av konsentrasjoner.** Lag plott som viser  $[H^+]$ ,  $[Al^{3+}]$ ,  $[Ca^{2+}]$  og  $[HCO_3^-]$  som funksjon av sulfatkonsentrasjonen  $s \in [10 \cdot 10^{-6}, 200 \cdot 10^{-6}]$ . Konsentrasjonene utover  $[H^+]$  kan finnes ved å løse likningene (2.23), (2.24) og (2.26). Variabler som har omtrent samme verdiområde kan presenteres i samme diagram. Kommenter hvordan konsentrasjonene endrer seg fra **A** til **B** som følge av endringen av parameteren  $K_{HCa}$ .

## 3.2 Oppgave 2: Innlesing av måledata

### Om oppgaven

Å lese inn numeriske data fra måleinstrumenter eller programoutput er en oppgave som man ofte støter på. Særlig måledata fra eldre instrumenter kommer i ikke-standardiserte

format, og krever “litt massasje” før de egner seg for prosessering. Noen programmer produserer også output med komplisert formatering. I denne prosjektoppgaven skal du lese inn data fra en tekstfil og lagre dataene i en Python-array slik at de lett kan brukes i simuleringene.

Du skal ikke skrive en fullstendig generell dataimporter, men koden du skriver skal i det minste kunne kjøre på andre datasett med samme format.

Datene ligger i filen `birkenes.data` og består av en tabell på følgende format:

	<code>cps04</code>	<code>nedboer</code>	<code>temp</code>	<code>avrenn</code>	<code>dato</code>
	mol/L	mm/dag	Deg. C	mm/dag	
1	-5.00e-04	0	1.78	7.909	01-May-87
2	4.30e-05	5.2	4.25	6.62	02-May-87
3	2.03e-05	3.8	4.05	5.132	03-May-87
4	-5.00e-04	0	4.13	4.39	04-May-87
5	.	.	.	.	.
6	.	.	.	.	.
7	2.81e-06	11.1	15.03	0.47	06-Aug-87
8	1.34e-05	70.6	15.23	84.199	07-Aug-87
9	1.34e-05	53.8	11.54	nan	08-Aug-87
10	1.03e-05	4.2	13.26	54.09	09-Aug-87
11	2.58e-05	0.8	15.96	7.307	10-Aug-87
12	.	.	.	.	.
13	.	.	.	.	.
14	.	.	.	.	.
15	.	.	.	.	.

Vi merker oss at det er to “headerlinjer” med henholdsvis navn og enhet på størrelsen som måles. Videre er sufatkonsentrasjonene satt til  $-5 \times 10^{-4}$  for de dagene det ikke er nedbør – konsentrasjonen er ikke veldefinert for disse dagene, ettersom det ikke er nedbør. Ett eller flere av målepunktene mangler, med verdien `nan` i stedet for et tall.

## Oppgave 2: Implementering av modulfunksjon

**Implementering av modulfunksjon.** Du skal skrive en funksjon for lesing av `birkenes.data`. Filen skal legges i verktøy-modulen, og tilfredsstill følgende:

- Funksjonen tar minst én parameter som skal være et filnavn, med tanke på anvendelse på andre måleserier fra birkenesfeltet.
- Funksjonen leser filen og lagrer alle numeriske data i minnet. Ignorer datokolonnen. Headerens første linje, altså merkelappene (utenom `dato`), skal også lagres.
- Funksjonen skal returnere et dictionary med merkelappene som nøkler og de tilhørende dataene organisert i en liste eller NumPy -array som verdier. Det vil si, at dersom du returnerer `d`, og `'temp'` er en av kolonnene, så er `d['temp']` en liste eller en array med verdier.

- Funksjonen skal ha en docstring og ellers være skrevet på en pen og oversiktlig måte, med kommentarer der det er naturlig.

**Testing.** Du skal skrive en testfunksjon/testfil. Test funksjonen din på følgende data, som du kan skrive manuelt inn i en editor og lagre som f.eks. `test_data.data`.

```

1 cps04 nedboer temp avrenn dato
2 mol/L mm/dag Deg. C mm/dag
3 1 2 3 4 1-Jan-17

```

### 3.3 Oppgave 3: Løsning av difflikninger med Eulers metode

I mange modelleringsproblemer ender man opp med *ordinære differensiallikninger* (ODEer). Dette er likninger, der den ukjente er en *funksjon*  $y(t)$  for  $t$  i et intervall  $[0, T]$ , på formen

$$\frac{d}{dt}y(t) = f(y(t), t), \quad y(0) = u. \quad (3.2)$$

Spesielt kalles et problem som dette et initialverdiproblem, siden vi ønsker å finne  $y(t)$  når  $y(0)$  er kjent. Differensiallikninger er ofte så kompliserte at man må ty til numeriske metoder for å finne approksimative løsninger. En slik metode er Eulers metode. Den er kanskje den aller enkleste metoden, og er basert på definisjonen av den deriverte,

$$\frac{d}{dt}y(t) := \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} [y(t + \delta t) - y(t)]. \quad (3.3)$$

Dersom vi bytter ut grensen  $\delta t \rightarrow 0$  med en fiksert konstant verdi  $\delta t > 0$  får vi

$$\frac{1}{\delta t} (y_{j+1} - y_j) \approx f(y_j, t_j). \quad (3.4)$$

der  $y_j = y(t_j)$ , med  $t_j = j\delta t$ . Løser vi for  $y_{j+1}$  får vi et såkalt eksplisitt oppdaterings-skjema,

$$y_{j+1} \approx y_j + \delta t f(y_j, t_j), \quad y_0 = u. \quad (3.5)$$

I praksis beregner man  $y_j$ ,  $j = 0, 1, \dots$  i en løkke. Merk at man kun trenger kunnskap om “hva som hendte i fortiden”,  $y_j$ , for å rekne ut  $y_{j+1}$ . Det er viktig å være oppmerksom på at Eulers metode kun er en approksimasjon, og at vi ikke har noen garanti for at den faktisk “virker” uten mer informasjon om høyresiden  $f$ .

Funksjonen  $y(t)$  kan godt være en vektorfunksjon, dvs, vi kan generalisere til problemet

$$\frac{d}{dt}\vec{y}(t) = \vec{f}(\vec{y}, t). \quad (3.6)$$

Her er

$$\vec{y} = (y^1, y^2, \dots, y^n), \quad (3.7)$$

og tilsvarende for  $\vec{f}(\vec{y}, t)$ , hvor hver  $f^j$  kan avhenge av alle  $y^k$ .<sup>1</sup> Vi får et system av ODEer:

$$\begin{aligned} \frac{d}{dt}y^1(t) &= f^1(y^1(t), y^2(t), \dots, y^n(t), t) \\ \frac{d}{dt}y^2(t) &= f^2(y^1(t), y^2(t), \dots, y^n(t), t) \\ &\vdots \\ \frac{d}{dt}y^n(t) &= f^n(y^1(t), y^2(t), \dots, y^n(t), t). \end{aligned} \quad (3.8)$$

Likningene (2.14a)–(2.14b) er et eksempel på et system av ODEer med  $n = 2$ , og med  $y^1(t) = A(t)$  og  $y^2(t) = B(t)$ .

**Oppgave 3: Eulers metode for et testproblem.** I denne oppgaven ska du skrive et selvstendig Python-skript. I motsetning til de tidligere oppgavene, velger vi å ikke skrive en modulfunksjon. I delprosjekt II skal du implementere Eulers metode for birkenesmodellen, og det er i det tilfellet mer hensiktsmessig å implementere Eulers metode direkte i en egen funksjon sammen med resten av det som skal beregnes.

For å bli kjent med Eulers metode skal du implementere et testproblem som har en eksakt løsning. For  $t \geq 0$ , betrakt ODEen

$$\frac{d}{dt}y^1(t) = \pi y^2(t) \quad (3.9)$$

$$\frac{d}{dt}y^2(t) = -\pi y^1(t) \quad (3.10)$$

med initialverdier  $y^1(0) = 0$  og  $y^2(0) = 1$ .<sup>2</sup> Likningen har analytisk løsning

$$y^1(t) = \sin(\pi t) \quad (3.11)$$

$$y^2(t) = \cos(\pi t). \quad (3.12)$$

Vi merker oss at tolker vi  $\vec{y}(t)$  som et punkt i planet ligger løsningen alltid på sirkelen  $\|\vec{y}\|^2 = (y^1)^2 + (y^2)^2 = 1$ .<sup>3</sup>

**Oppgave 3a** Benytt approksimasjonen for den deriverte gitt i uttrykk (3.4) for å skrive om ligningene i (3.9)–(3.10) til et eksplisitt oppdateringsskjema. Forklar hvordan du går frem.

<sup>1</sup>Notasjonen  $y^i$  for komponent nummer  $i$  av vektoren  $\vec{y}$  må ikke forveksles med tallet  $y$  opphøyd i  $i$ -te potens.

<sup>2</sup>Systemet kan skrives som  $dy(t)/dt = f(y(t))$  med  $f^1(y) = \pi y^2$  og  $f^2(y) = -\pi y^1$ .

<sup>3</sup>Her bruker vi potenser siden  $y^j$  er et tall!

```

1 import matplotlib.pyplot as plt
2
3 y1_approx, y2_approx = euler_solve(...)
4 y1_exact, y2_exact = exact_solve(...)
5
6 plt.figure(1)
7 plt.plot(y1_approx, y2_approx, "-", label = "$y_{approx}$")
8 plt.plot(y1_exact, y2_exact, "--", label = "$y_{exact}$")
9 plt.show()

```

Figur 3.2: Et eksempel på hvordan  $y^1$  og  $y^2$  plottes mot hverandre.

**Oppgave 3b** Skriv et program som anvender Eulers metode på systemet i intervallet  $t \in [0, 10]$  med tidssteg  $\delta t \in \{1, 0.1, 0.01\}$ . For hvert valg av tidssteg skal du plote komponentene mot hverandre (se eksempel i figur 3.3). Husk tittel og tekst på aksene. Hvordan påvirket valget av tidssteg den approksimative løsningen? Kommenter resultatet.

# Kapittel 4

## Delprosjekt II

### 4.1 Oppgave 4: Transport av vann

#### Om oppgaven

I denne oppgaven skal du skrive et Python-skript som løser den hydrologiske delmodellen beskrevet i seksjon 2.1.<sup>1</sup> Det kan lønne seg å dele skriptet i hovedseksjoner, for eksempel:

1. Importering av ulike moduler, slik som verktøymodulen fra delprosjekt I.
2. Definerings av diverse globale variable, som parameterverdier som går igjen i hele prosjektet. Eksempler er  $A_{\min}$ ,  $B_{\min}$ ,  $K_{AIH}$ ,  $Q_{\text{over}}$ , osv.
3. Definerings av hjelpefunksjoner for å gjøre beregninger man ofte gjør men med kompliserte definisjoner. For eksempel kan det lønne seg å ha en egen funksjon for å rekne ut  $A_{\text{sig}}$  som funksjon av  $B$ .
4. Definisjon av hovedsimuleringsrutiner.
5. Kjøring av hovedsimuleringsrutine og visualisering av resultat.

Du står selvstendig fritt til å organisere programmet ditt slik du ønsker. De øvrige oppgavene kan med fordel implementeres i det samme skriptet som i denne oppgaven, fortløpende, ettersom du gjenbruker resultatene.

Den hydrologiske delmodellen (vanntransporten) beskrevet i seksjon 2.1 består av to koblede differensiallikninger (2.14a) og (2.14b) med tilhørende initialbetingelser:

$$\frac{dA}{dt} = P - E_A - Q_A, \quad (4.1a)$$

$$\frac{dB}{dt} = Q_A \cdot A_{\text{sig}} - E_B - Q_B - Q_{\text{over}}, \quad (4.1b)$$

---

<sup>1</sup>Merk at difflikningene er uavhengige av difflikningene for sulfattransport. De kan derfor løses separat, men ikke omvendt: for sulfattransport trenger vi løsningen fra den hydrologiske modellen.

$$A(0) = A_{\min}, \quad B(0) = B_{\min}.$$

Her er  $A(t)$  og  $B(t)$  vannstandene i henholdsvis  $A$ - og  $B$ -reservoaret målt i mm som funksjon av tiden  $t$ . Størrelsene  $Q_A$  og  $Q_B$  er definert i likning (2.9), og  $A_{\text{sig}}$  i likning (2.10). Evapotranspirasjonen  $E_A$  og  $E_B$  er gitt i likning (2.11).

Vi lar nå  $h$  være steglengden i Eulers metode.<sup>2</sup> Bruker vi vannbudsjettuttrykkene (2.2) og (2.5b) for å definere Eulers metode får vi

$$A_{j+1} = A_j + h[P(t_j) - E_A(t_j) - Q_A(t_j)] \quad (4.2a)$$

$$B_{j+1} = B_j + h[A_{\text{sig}}(t_j)Q_A(t_j) - E_B(t_j) - Q_B(t_j) - Q_{\text{over}}(t_j)] \quad (4.2b)$$

Her har vi spesifisert at leddene i høyresiden skal evalueres for  $t = t_j$ . Initialbetingelsene er gitt for  $t = t_0 = 0$ , det vil si  $A_0 = A_{\min}$  og  $B_0 = B_{\min}$ . Her er  $Q_{\text{over}}$  gitt som

$$Q_{\text{over}} = \max[(B - B_{\max})/h + A_{\text{sig}}Q_A - E_B - Q_B, 0]. \quad (4.3)$$

Dette uttrykket er slik at Eulers metode vil tømme alt det overskytende vannet i løpet av ett tidssteg over i bekken.

Tidsgitteret er som følger:  $t_j = hj$ , med  $j = 0, 1, \dots, n$ , og  $t_n = t_{\text{final}}$  er den siste dagen vi har data for temperatur og nedbør. Merk at du har  $n + 1$  punkter totalt. Vi setter  $h = 1/d$  dag, med  $d$  et heltall, slik at det blir eksakt  $d$  beregningspunkter per dag. Da har vi  $n = t_{\text{final}}d$ .

**Interpolering av data** Vi ser at delmodellen (4.1) “drives” av temperaturen  $T$  og nedbøren  $P$ . For disse størrelsene skal vi bruke måledata<sup>3</sup> som er lagret i `birkenes.data`. Disse dataene har for grov oppløsning til å løse difflikningene tilfredsstillende. Derfor skal de *interpoleres*. Temperaturen er en jevnt svingende funksjon og interpoleres med en glatt funksjon, en såkalt *kubisk spline*. Nedbør og sulfatkonsentrasjon i nedbør svinger ikke jevnt, og vi bruker her en *stykkevis konstant tilnærming*. Her er oppskriften på dette:

<sup>2</sup>Denne notasjonen er ofte brukt for maskevidden på et gitter, og det er mange som synes variabelnavn som `delta_t` er uoversiktlige i programmer.

<sup>3</sup>Verdiene er gjennomsnitt pr. døgn.

```

1 # Vi antar at NumPy er importert via kommandoen:
2
3 import numpy as np
4
5 # temperatur0 = NumPy array med data fra 'birkenes.data'
6 # nedboer0 = NumPy array med data fra 'birkenes.data'
7 # sulfat0 = NumPy array med data fra 'birkenes.data'
8
9 # Lag en array med tidsaksen dataene er lagret p{\aa}
10 t0 = np.array(range(len(temperatur0)))
11
12 temperatur = interp1d(t0, temperatur0, kind = 'cubic')
13 nedboer = interp1d(t0, nedboer0, kind = 'zero')
14 sulfat = interp1d(t0, sulfat0, kind = 'zero')
15
16 # Naa kan vi finne temperatur, nedb{\o}r og sulfat for _alle_
17 # tidspunkt mellom dagene, feks:
18
19 test = temperatur(0.5) # beregner T midtveis mellom dag 0 og 1.

```

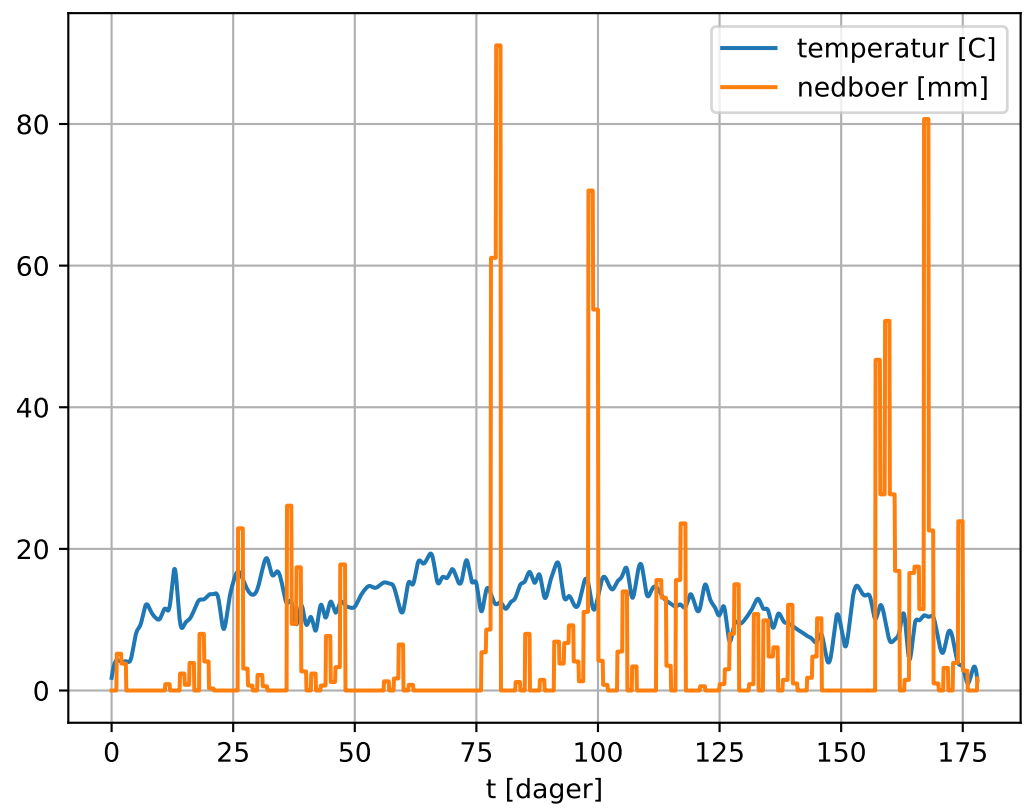
Figur 4.1 viser et plott av de resulterende funksjonene for temperatur og nedbør, produsert med følgende kode:

```

1 import matplotlib.pyplot as plt
2
3 t_slutt = t0[-1]
4 t = np.linspace(0, t_slutt, 2000) # 2000 plottepunkter totalt
5
6 plt.figure()
7 plt.plot(t,temperatur(t), t, nedboer(t), t)
8 plt.grid()
9 plt.legend(['temperatur [C]', 'nedboer [mm]'])
10 plt.xlabel('t [dager]')
11
12 plt.show()

```





Figur 4.1: Interpolert nedbør og temperatur

#### Oppgave 4a: Implementering

Skriptet skal løse (4.2) som beskrevet over. Når løsningen  $A$  og  $B$  er beregnet, bestemmer du avrenningen til bekken

$$Q = (1 - A_{\text{sig}}) \cdot Q_A + Q_B + Q_{\text{over}}.$$

Steglengden skal være  $h = 1.0/d$  dag med  $d$  et heltall, men du står ellers fritt til å eksperimentere. Passende verdier kan være  $d = 2$  og  $d = 10$ . Det kan være lurt å la  $d$  være en parameter til løseren. Beregn for alle  $t_j = hj \leq t_{\text{final}}$ , der  $t_{\text{final}}$  er den siste dagen du har inputdata for.

**Massebalanse.** En viktig sjekk på om beregningene er fornuftige er å undersøke om du har *massebalanse*, om vannbudsjettet stemmer. Det vil si at du undersøker om

$$\underbrace{\text{“Endring i } A \text{ og } B\text{”}}_{\text{VS}} \stackrel{?}{=} \underbrace{\text{“Akkumulert strøm inn”} - \text{“Akkumulert strøm ut”}}_{\text{HS}}. \quad (4.4)$$

Vi kan uttrykke (4.4) litt mer formelt. Du skal sammenligne

$$\text{VS} = A(t_n) - A(0) + B(t_n) - B(0) \quad (4.5)$$

med

$$\text{HS} = \int_0^{t_n} (P - Q - E) ds \approx \sum_{j=0}^{n-1} (P_j - Q_j - E_j)h, \quad (4.6)$$

der  $E = E_A + E_B$  og  $t_n$  er slutt-tidspunktet for simuleringen. Videre er  $A_k, B_k, P_k, Q_k$  og  $E_k$  er de numeriske verdiene for henholdsvis  $A, B, P, Q$  og  $E$  ved tidspunktet  $t = t_k$  for  $k = 0, 1, \dots, n-1$ . Faktoren  $h$  er som tidligere steglengden som benyttes i Eulers metode. Forskjellen mellom verdiene VS og HS skal være liten, nær null i til maskinpresisjon.

Implementer sjekken for massebalanse i simuleringrutinen.

#### Oppgave 4b: Simuleringer

Eksperimenter med ulike verdier av  $d$  for å finne en verdi du synes er rimelig ved å undersøke plott av  $A$  og  $B$  som funksjon av  $t$  i samme koordinatsystem. Ett representativt tilfelle skal være med i besvarelsen.

Lag så et plott av  $Q$  i samme koordinatsystem som de målte døgnerverdiene i `birkenes.data`, i kolonnen `'avrenn'`. (Disse skal ikke interpoleres.) Diskuter. Ta gjerne Figur 4.1 med i betraktning.

## 4.2 Oppgave 5: Transport av sulfat

### Om oppgaven

I denne oppgaven skal du skrive et selvstendig program. Om du ønsker, kan du bygge ut programmet fra oppgave 4.

I den kjemiske delmodellen fra seksjon 2.2 inngår et system av differensiallikninger som beskriver transport av sulfat. Dette systemet er ganske likt det vi betraktet i den hydrologiske delmodellen,

$$\frac{d}{dt}M_A = P \cdot C_P - Q_A \cdot C_A, \quad (4.7a)$$

$$\frac{d}{dt}M_B = A_{\text{sig}} \cdot Q_A \cdot C_A - (Q_B + Q_{\text{over}}) \cdot C_B, \quad (4.7b)$$

og initialbetingelsene

$$M_A(0) = 4 \cdot 10^{-5} \text{ mol/L} \cdot A(0), \quad M_B(0) = 4 \cdot 10^{-5} \text{ mol/L} \cdot B(0).$$

Her betegner  $M_A$  og  $M_B$  sulfatmengdene i henholdsvis  $A$ - og  $B$ -reservoaret målt i  $\text{mol/m}^2$ . Videre er sulfatkonsentrasjonene i de to reservoarene gitt som

$$C_A = \frac{M_A}{A}, \quad C_B = \frac{M_B}{B},$$

der  $A$ ,  $B$  er løsningen av (4.1). Konsentrasjonen av sulfat i nedbøren,  $C_P$ , er gitt ved observasjonsverdier som leses fra filen `birkenes.data` (kolonne 1). Vi betrakter  $C_P$  som en stykkevis konstant funksjon, akkurat som nedbøren  $P$ . Vær oppmerksom på at  $C_P$  har negative verdier for de dagene det ikke regner (når  $P = 0$ ).

Merk at  $A$  og  $B$ , og dermed  $A_{\text{sig}}$ ,  $Q_A$ ,  $Q_B$  og  $Q_{\text{over}}$  er fullstendig bestemt av den hydrologiske delmodellen (4.1). De eneste ukjente er  $M_A$  og  $M_B$  som funksjon av tid.

### Oppgave 5a: Implementering.

Skriv ned Eulers metode for systemet (4.7). Implementer denne i et skript. Du kan med fordel bygge ut programmet fra oppgave 4, enten ved å skrive en ny simuleringsfunksjon eller utvide den gamle, til å løse systemet (4.7). Merk at  $A$  og  $B$  fra den hydrologiske delmodellen inngår, slik at denne må løses først, evt. samtidig. Bruk løsningen av (4.7) til å beregne sulfatkonsentrasjonen i avrenningen,

$$C_Q = \frac{1}{Q} \left( (1 - A_{\text{sig}}) \cdot Q_A \cdot C_A + (Q_B + Q_{\text{over}}) \cdot C_B \right). \quad (4.8)$$

**Massebalanse.** Som for de likningene for  $A$  og  $B$  skal programmet testes ved å undersøke om rutinen gir massebalanse for sulfat, og om differansen  $VS - HS$  går mot null når  $h$  avtar. For sulfattransport har vi

$$VS = M_A(t_{\text{final}}) - M_A(0) + M_B(t_{\text{final}}) - M_B(0) \quad (4.9)$$

$$HS = \int_0^{t_{\text{final}}} g(t) dt \approx h \sum_{j=1}^{n-1} g(t_j) \quad (4.10)$$

med

$$g(t) = P(t) \cdot C_P(t) - [1 - A_{\text{sig}}(t)] \cdot Q_A(t) \cdot C_A(t) - [Q_B(t) + Q_{\text{over}}(t)] \cdot C_B(t) \quad (4.11)$$

Testen skal inngå som integrert del av simuleringsrutinen.

### Oppgave 5b: Simuleringer

For verdien av  $d$  du brukte i oppgave 4b, regn ut konsentrasjonene  $C_A$ ,  $C_B$ , og  $C_Q$  som funksjon av tid. Plott  $C_A$  og  $C_B$  i samme aksesystem, og lag et plott av  $C_Q$  i et nytt system. Diskuter.

Det kan her være lurt å eksperimentere med  $d$  for å se om du ser noen endringer i resultat-plottene. Du må da passe på at den hydrologiske modellen kjøres med samme steglengde.

## 4.3 Oppgave 6: Den fullstendige forsuringsmodellen

### Om oppgaven

I denne oppgaven skal du finne konsentrasjonene av kationene  $H^+$ ,  $Ca^{2+}$  og  $Al^{3+}$  samt bikarbonat,  $HCO_3^-$ , i avrenningen  $Q$ . Verktøyene vi har utviklet hittil vil være nyttige. Du kan fikse steglengden  $h = 1.0/d$  dag til en passende verdi du mener gir et godt resultat. Du kan gjerne bygge ut tidligere program for å løse oppgaven.

Fra kapittel 2 husker vi at konsentrasjonene av kationene og bikarbonat er i begge reservoarene gitt ved likevektslikningene

$$K_{HCa} = \frac{[H^+]^2}{[Ca^{2+}]} \quad (4.12)$$

$$K_{AlH} = \frac{[Al^{3+}]}{[H^+]^3} \quad (4.13)$$

$$K_H = [H^+][HCO_3^-] \quad (4.14)$$

der  $[\cdot]$  står for konsentrasjoner målt i mol/L. Her er  $K_{HCa}$  lik  $10^{-2.2}$  for **A**-reservoaret og  $10^{-3.2}$  for **B**-reservoaret. I begge reservoarene er  $K_{AlH} = 10^9$  og  $K_H = 2.5 \cdot 10^{-10}$ . I begge reservoarene gjelder også elektronøytralitetsprinsippet:

$$[H^+] + 2[Ca^{2+}] + 3[Al^{3+}] = 2[SO_4^{2-}] + [HCO_3^-] \quad (4.15)$$

### Oppgave 6a: Beregning av ionekonsentrasjoner i avrenning.

Gitt sulfatkonsentrasjonene i hvert reservoar ( $C_A = [H^+]_A$  og  $C_B = [H^+]_B$ ) fra oppgave 5, kan du beregne konsentrasjonene av kationer og bikarbonat i reservoarene på samme måte som i oppgave 1b,

$$[Ca^{2+}]_A, [Ca^{2+}]_B, \text{ osv.}$$

Gjør dette for alle beregningspunktene  $t_j$  hvor vi kjenner sulfatkonsentrasjonene, slik at du finner konsentrasjonene av kationer og bikarbonat i hvert reservoar som funksjoner av tiden.

Regn ut konsentrasjonene i avrenningen også. I første omgang gjør du dette for alle fire komponenter på samme måte som for sulfat (se likning (4.8)). For eksempel er  $[H^+]_Q$ , konsentrasjonen av hydronium i avrenningen, gitt ved

$$[H^+]_Q = \frac{1}{Q} \left( (1 - A_{\text{sig}}) \cdot Q_A \cdot [H^+]_A + (Q_B + Q_{\text{over}}) \cdot [H^+]_B \right). \quad (4.16)$$

For de tre andre ionekonsentrasjonene blir det tilsvarende uttrykk. Som i forrige kapittel setter vi konsentrasjonene til null ved tidspunkt der  $Q = 0$ .

### Oppgave 6b: Likevektsjustering.

Når vi tidligere beregnet konsentrasjonene i reservoarene **A** og **B**, brukte vi relasjonene (4.12)–(4.14) sammen med elektronøytralitetsbetingelsen (4.15). Denne framgangsmåten medfører at i hvert reservoar er alle likningene (4.12)–(4.15) oppfylte. Når vi deretter bruker uttrykk på formen (4.16) til å utlede konsentrasjoner i avrenningen, vil elektronøytralitetsprinsippet fortsatt gjelde, mens likevektslikningene (4.12)–(4.14) ikke er oppfylte for kjemien i avrenningen. Kan du forklare hvorfor dette skjer?

I beskrivelsen av den kjemiske delmodellen i kapittel 2 nevnte vi at  $\text{CO}_2$ -innholdet i jordvæsken er større enn i overflatevann. Dette kommer blant annet av respirasjon fra mikroorganismer i jorda. Når jordvæsken (her: vann i reservoarene) kommer ut i atmosfæren (her: avrenning), vil  $\text{CO}_2$  fordampe. Vi må ta hensyn til denne prosessen ved å justere konsentrasjonene  $[\text{H}^+]_Q$ ,  $[\text{Al}^{3+}]_Q$  og  $[\text{HCO}_3^-]_Q$ . Rent praktisk gjøres dette ved å beholde sulfat- og kalsiumkonsentrasjonene ( $[\text{SO}_4^{2-}]_Q$  og  $[\text{Ca}^{2+}]_Q$ ) uendret, mens de resterende komponentene beregnes slik at likningene (4.13)–(4.15) gjelder<sup>4</sup> for avrenningen. Målinger fra Birkenes-feltet har vist at konstantene i (4.13) og (4.14) i dette tilfellet bør velges som  $K_{AlH} = 10^9$  og  $K_H = 1.2 \cdot 10^{-11}$ .

Fra elektronøytralitetsbetingelsen og likevektslikningene i avrenningen, skriv ned en algebraisk likning for  $[\text{H}^+]_Q$ . Husk at kalsium og sulfat har kjente konsentrasjoner.

Finn  $[\text{H}^+]_Q$  ved hjelp av Newtons metode, og beregn deretter de justerte  $[\text{Al}^{3+}]_Q$ ,  $[\text{HCO}_3^-]_Q$ , for alle tidspunkter  $t$  i gitteret.

### Oppgave 6c: Grafiske framstillinger.

Lag diagrammer som viser konsentrasjonene av hydronium, aluminium og bikarbonat i avrenningen før og etter justering, som funksjon av tiden. Lag et diagram som viser sulfatkonsentrasjonen og ett som viser bikarbonatkonsentrasjonen som funksjon av tiden.

Med tanke på giftigheten av aluminium og surheten i vannet for fisk som ørret (se seksjon 2.2), hva tenker du om langtidsperspektivet for et eventuelt vann der bekken renner ut?

Lag også et plott som viser at de justerte konsentrasjonene i avrenningen tilfredsstiller elektronøytralitetsprinsippet (4.15) i hele simuleringsperioden. Dette gjør du enklest ved å plote størrelsen

$$e = [\text{H}^+]_Q + 2[\text{Ca}^{2+}]_Q + 3 \cdot [\text{Al}^{3+}]_Q - 2 \cdot [\text{SO}_4^{2-}]_Q - [\text{HCO}_3^-]_Q$$

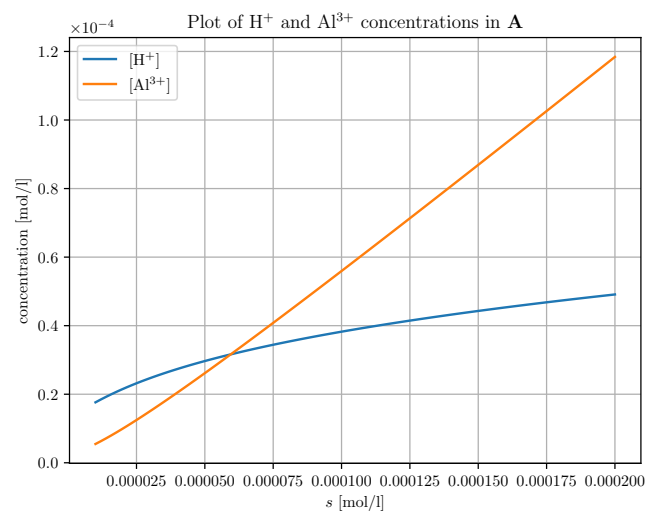
som funksjoner av tiden. Plott  $e$  for både før og etter justering og sammenlikn. Tallene skal være veldig små etter justering.

---

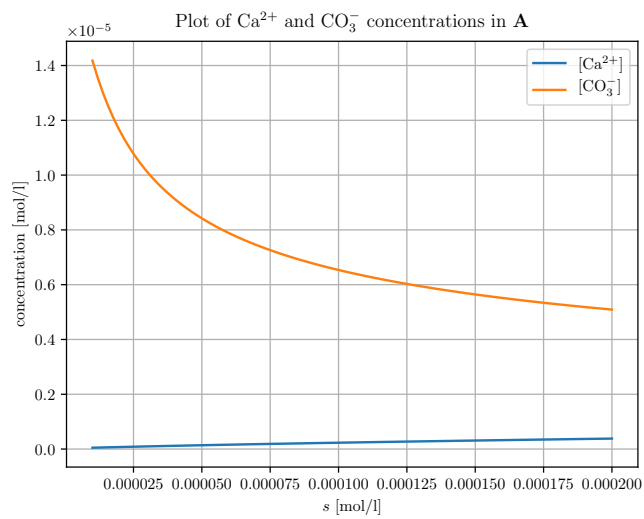
<sup>4</sup>Siden  $[\text{Ca}^{2+}]_Q$  ikke endres, er det ikke aktuelt å bruke likning (4.12) under justeringen.

## Tillegg A

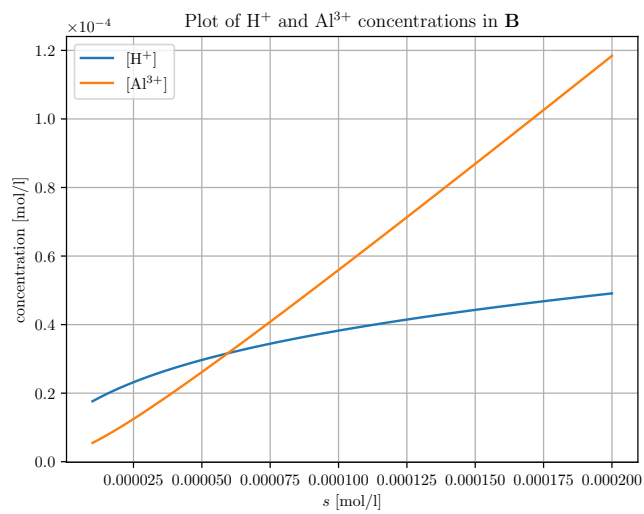
# Eksempelsimuleringer til kontroll



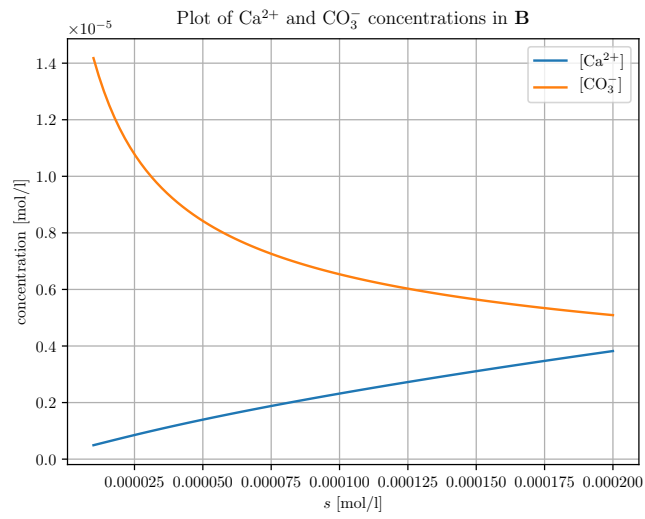
Figur A.1: Konsentrasjoner i oppgave 1.



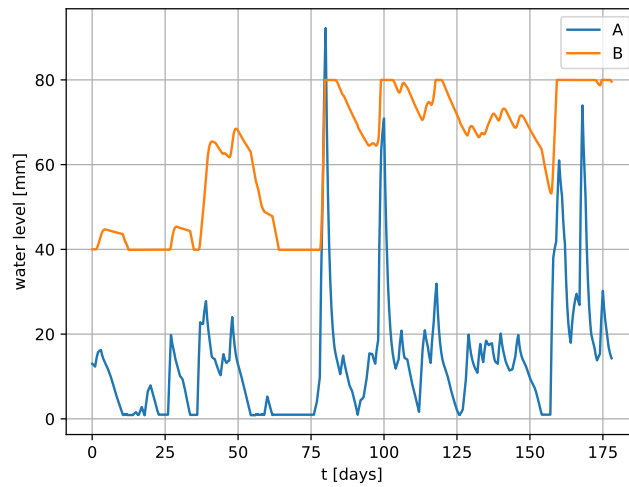
Figur A.2: Konsentrasjoner i oppgave 1.



Figur A.3: Konsentrasjoner i oppgave 1.

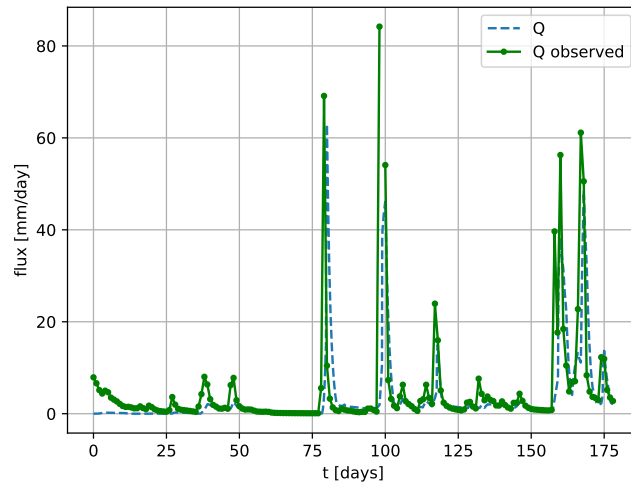


Figur A.4: Konsentrasjoner i oppgave 1.

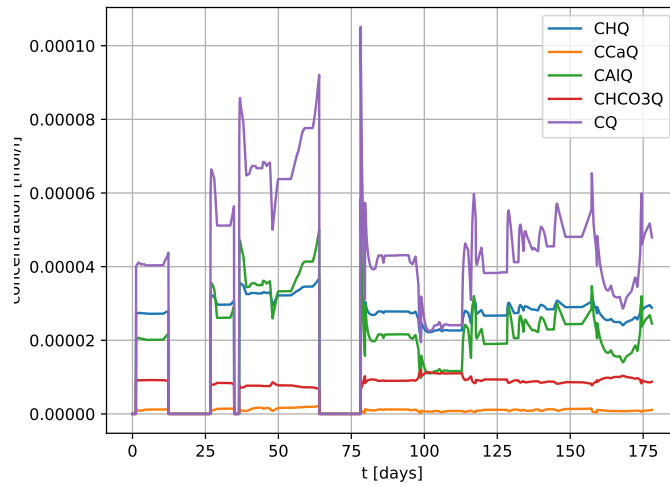


Figur A.5: Beregning med  $d = 20$ .

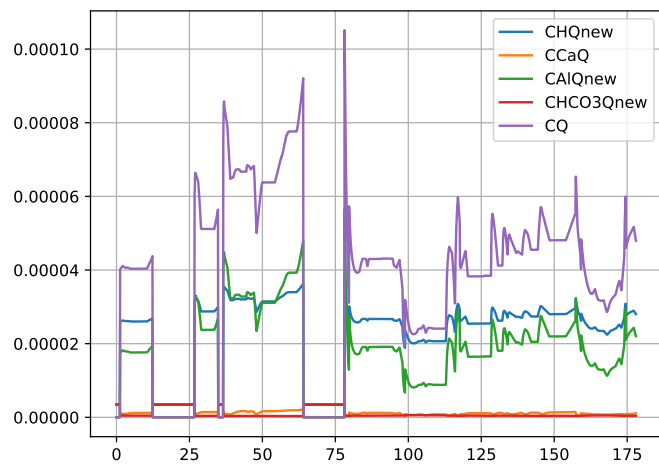




Figur A.6: Observert avrenning vs. beregnet avrenning.  $d = 50$ .



Figur A.7: Beregning med  $d = 50$ .



Figur A.8: Beregning med  $d = 50$ .