Matte 2 Oblig

Luis Patel Berglund

10.04.2025

Så var det på tide å finne på noe kult å levere som oblig, der jeg strakk hjernen min for å finne på noe som ikke var standardprosjekter. Det er jo ikke bare bare da det finnes så utrolig mye rart her i verden man kan bruke matte til. Enten det er å påvirke aksjemarkedet med halvhjertede ligninger for å sette tariffer, eller å sende en rakett til månen. Men her sitter jeg,på bursdagen min, med en kopp dobbel espresso og en halvtørr donut fra Bunnpris , og da var det lite annet jeg kunne gå for...

1 Intro

Som en ny student i Trondheim trengte jeg en hobby som jeg kunne gjøre til min identitet. Enkelt: det ble kaffe. Jeg kjøpte meg en espressomaskin, og ikke minst en overpriced kvern.

Det er mange faktorer som spiller inn for å få en skikkelig god espresso. Maskinen må bygge opp nok trykk til å presse vannet gjennom kaffen. Så har vi hvor fint eller grovt kaffen er malt, og ikke minst temperaturen på vannet. Jeg ønsker å undersøke sammenhengen mellom disse faktorene ved hjelp av Python og matematisk modell.

2 Metode Espresso delen

Det er igjen mange valg å ta da man skal velge på hva man skal se, men jeg har nå heldigvis lært meg noen formler gjennom min tid på gløs haugen (Noen av formlene som kommer nå er vel fra fluidmekanikk, så om dette er for mye vann over hodet ser vi fort.)

1. Arrhenius' lov (temperaturavhengig ekstraksjonshastighet)

$$k = A \cdot e^{-\frac{E_a}{RT}} \tag{1}$$

Forklaring: Arrhenius' lov beskriver hvordan reaksjonshastigheter øker med temperaturen. Dette brukes i kaffemodellen for å estimere hvor raskt smakskomponenter ekstraheres ved ulike bryggetemperaturer.

- k: reaksjonshastighetskonstant
- A: pre-eksponentiell faktor (konstant)
- E_a: aktiveringsenergi (J/mol)
- R: gasskonstant ($8.314\,\mathrm{J/mol \cdot K}$)
- T: temperatur (K)

2. Darcy's lov (væskestrøm gjennom porøst materiale)

$$Q = \frac{\kappa \cdot A \cdot \Delta P}{\mu \cdot L} \tag{2}$$

Forklaring: Darcy's lov brukes for å beregne vannets gjennomstrømning gjennom kaffen, som oppfører seg som et porøst materiale.

- Q: volumstrøm (m^3/s)
- κ: permeabilitet (m^2), typisk $\kappa \propto \frac{d^2}{150}$
- A: tverrsnittsareal av filteret (m^2)
- ΔP: trykkforskjell (Pa)
- μ: væskeviskositet (Pa $\cdot$ s)
- L: høyde på kaffebedet (m)

3. Overflateareal for kaffepartikler

$$A_{\text{total}} = \frac{3 \cdot m}{\rho \cdot r} \tag{3}$$

Forklaring: Dette uttrykket gir total overflateareal for en mengde sfæriske partikler, som vannet kan komme i kontakt med under ekstraksjonen. Dette er da en antaglese om at det er sfæriske partikler, noe som i virkeligheten ikke vil være helt sant.

- A_{total}: total overflateareal (m^2)
- m: masse kaffe (kg)
- ρ: partikkeltetthet (kg/m^3)
- r: partikkelradius (m), vanligvis $r = \frac{d}{2}$

4. Massetransport / ekstraksjonshastighet

$$\text{Rate} = k \cdot A \cdot (C_s - C) \cdot Q \tag{4}$$

Forklaring: Denne ligningen blir da en kombinasjons-ligning som uttrykker hvor raskt løselige forbindelser blir overført fra kaffegrut til vann. Den kombinerer Arrhenius' lov, overflateareal av sfæriske partikler, og Darcy's lov.

- k: ekstraksjonskonstant (fra Arrhenius)
- A: overflateareal av partiklene
- C_s: metningskonsentrasjon (maks løselighet)
- C: aktuell konsentrasjon i væsken
- Q: væskestrøm (fra Darcy's lov)

2.1 Pyhton kode for plot

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from mpl_toolkits.mplot3d import Axes3D #Viktig    ha 3D, engang i tiden var det skikkelig kult. Synd jeg ikke fikk sett Minecraft movie i 3D
from matplotlib import cm

# Fysiske konstanter
gasskonstant = 8.314  # J/mol K
optimal_temp = 93 + 273.15  # K (93 C )
maks_l selighet = 0.3  #S kte p   google, hva er max l selighet p   vann, fikk ish 0.3 ergo ish 30%
partikkel_tetthet = 500  # kg/ m

# Kaffe-komponenter og deres egenskaper
sammensetning = {
    'Syrer': {'maks_l selighet': 0.08, 'aktiveringsenergi': 38000, 'farge': 'red'},
    'Sukkerarter': {'maks_l selighet': 0.12, 'aktiveringsenergi': 45000, 'farge': 'green'},
    'Bitterstoffer': {'maks_l selighet': 0.1, 'aktiveringsenergi': 52000, 'farge': 'blue'}
} #Ett rask google s k , og dette var det jeg fant ut var de mest avgj rdene komponente som bidrar til smak.

def viskositet(T):

    return 0.001 * np.exp(2000 * (1/T - 1/373.15))

def ekstraksjonsrate(konsentrasjon, kverningsgrad, temp, trykk, dose=18e-3,
                     maks_l s=0.3, Ea=45000):

    k = 5e-3 * np.exp(-Ea / (gasskonstant * temp))

    h yde_bed = 0.03  # meter, alts   da h yden p   "Portafilteret(nerdeemoji)"
    permeabilitet = (kverningsgrad ** 2) / 150  # Kozeny-Carman-relasjon
    mu = viskositet(temp)

    gjennomstr mning = (np.pi * 0.03**2 * permeabilitet * trykk) / (mu * h yde_bed)

    overflate = 3 * dose / (partikkel_tetthet * (kverningsgrad / 2))

    return k * overflate * (maks_l s - konsentrasjon) * gjennomstr mning

def ekstraksjons_ode(y, t, d, T, P, dose=18e-3, maks_l s=0.3, Ea=45000):

    C, S = y
    rate = ekstraksjonsrate(C, d, T, P, dose, maks_l s, Ea)
    if rate * t > S and S > 0:
        rate = S / t

    return [rate, -rate]

def ekstraksjonsmodell(kverningsgrad, temp=optimal_temp, trykk=9e5, dose=18e-3,
                       kontakttid=25, maks_l s=0.3, Ea=45000):

```

```
    t = np.linspace(0.01, kontakttid, 100)
    starttilstand = [0, maks_l s]

    l sning = odeint(ekstraksjons_ode, starttilstand, t, args=(kverningsgrad, temp, trykk, dose, maks_l s, Ea))

    return t, l sning[:, 0], l sning[:, 1]

def lag_espresso_analyse():
    fig = plt.figure(figsize=(15, 15))

    # Plot 1: Effekt av kverningsgrad p  ekstraksjon
    ax1 = fig.add_subplot(221)
    kverningsgrader = [150e-6, 250e-6, 350e-6, 450e-6]
    for d in kverningsgrader:
        t, C, _ = ekstraksjonsmodell(d)
        ax1.plot(t, C, label=f"{d*1e6:.0f}  m ")
    ax1.set_xlabel('Tid (s)')
    ax1.set_ylabel('Konsentrasjon')
    ax1.set_title('Effekt av kverningsgrad p  ekstraksjon')
    ax1.legend()
    ax1.grid(True)

    # Plot 2: Optimal kverningsgrad og f lsomhet
    ax2 = fig.add_subplot(222)
    d_range = np.linspace(100e-6, 500e-6, 40)
    utbytter = [ekstraksjonsmodell(d)[1][-1] / maks_l selighet * 100 for d in d_range]
    f lsomhet = np.gradient(utbytter, d_range * 1e6)

    ax2.plot(d_range*1e6, utbytter, 'b-', label='Utbytte (%)')
    ax2_twin = ax2.twinx()
    ax2_twin.plot(d_range*1e6, f lsomhet, 'r--', label='F lsomhet')

    ax2.set_xlabel('Kverningsgrad ( m )')
    ax2.set_ylabel('Ekstraksjonsutbytte (%)', color='b')
    ax2_twin.set_ylabel('F lsomhet', color='r')
    ax2.set_title('Optimal kverningsgrad')
    ax2.axvspan(180, 380, alpha=0.2, color='green', label='Optimalt omr de')

    linjer1, etiketter1 = ax2.get_legend_handles_labels()
    linjer2, etiketter2 = ax2_twin.get_legend_handles_labels()
    ax2.legend(linjer1 + linjer2, etiketter1 + etiketter2)
    ax2.grid(True)

    # Plot 3: Ekstraksjon av ulike smaksstoffer
    ax3 = fig.add_subplot(223)
    for navn, egenskaper in sammensetning.items():
        t, C, _ = ekstraksjonsmodell(
            250e-6, maks_l s=egenskaper['maks_l selighet'], Ea=egenskaper['aktiveringsenergi']
        )
        ax3.plot(t, C, color=egenskaper['farge'], label=navn)

    t = np.linspace(0.01, 25, 100)
    total = np.zeros_like(t)
    for egenskaper in sammensetning.values():
        _, C, _ = ekstraksjonsmodell(
```

```
            250e-6, maks_l s=egenskaper['maks_l selighet'], Ea=egenskaper['aktiveringsenergi']
        )
        total += C

    ax3.plot(t, total, 'k--', label='Total')
    ax3.set_xlabel('Tid (s)')
    ax3.set_ylabel('Konsentrasjon')
    ax3.set_title('Ekstraksjon av smaksprofiler')
    ax3.legend()
    ax3.grid(True)

    # Plot 4: 3D     Temperatur og kverningsgrad vs utbytte
    ax4 = fig.add_subplot(224, projection='3d')
    d_vals = np.linspace(150e-6, 400e-6, 10)
    T_vals = np.linspace(88, 96, 10) + 273.15
    D, T = np.meshgrid(d_vals, T_vals)
    Z = np.zeros_like(D)

    for i in range(len(T_vals)):
        for j in range(len(d_vals)):
            _, C, _ = ekstraksjonsmodell(D[i, j], temp=T[i, j])
            Z[i, j] = C[-1] / maks_l selighet * 100

    flate = ax4.plot_surface(D*1e6, T - 273.15, Z, cmap=cm.viridis, linewidth=0, antialiased=True)
    ax4.set_xlabel('Kverningsgrad ( m )')
    ax4.set_ylabel('Temperatur ( C )')
    ax4.set_zlabel('Utbytte (%)')
    ax4.set_title('Sammenheng mellom temperatur, kverningsgrad og utbytte')

    fargebar = fig.colorbar(flate, ax=ax4, shrink=0.5, aspect=5)
    fargebar.set_label('Ekstraksjonsutbytte (%)')

    plt.tight_layout()
    plt.show()
    plt.savefig()
    return fig

lag_espresso_analyse()
```

3 Plots og tolkning

3.1 Plot 1

Jeg starter med å se på det første plotet(1). Dette plottet viser en sammenligning konsentrasjonen av løste stoffer i kaffen over tid for ulike kverningsgrader (150–450 µm).

Den viser oss en finerekveringsgrad, altså laverer µm, vil gi en raskere og høyeree ekstraksjon. Det er grunnet mindre partikler gir større overflate og dermed raskere ekstraksjon. Som vil for koppen med kaffe bety at hvis den blir for fin kan det lede til en veldig bitter smak, det vil da være en overekstraksjon. Og da motsatt vil en for grov kverna kaffe gi oss en underekstrahert kaffe som da vil være svak og sur.

3.2 Plot 2

Det andre plottet (2) vil da vise oss "Optimal" kverningsgrad. Vi ser på grafen:

- Blå kurve: Ekstraksjonsutbytte i prosent som funksjon av kverningsgrad.
- Rød strek: Følsomhet, dvs. hvor raskt utbyttet endrer seg med kverningsgraden.
- Grønt område: Anbefalt optimalt område for kverningsgrad (180–380 µm)

Dette gir da en antydning til hva den ideelle kveringsgraden vil være basert på veridene man putter inn i koden. Jeg vil si at denne var litt optimistisk å prøve å lage da denne er en veldig teoretisk, og veldig enkel modell. For det er ikke alltid du har lyst på høyest utbytte, for det kan fort gjøre at det blir dårlig smak. Men man kan jo teste, sjekke om det funker bra med den gitte kveringsgraden f.eks i krysningen mellom følsomheten og ekstraksjonsutbyttet.

3.3 Plot 3

Plot 3 (3) skal vise hvordan ulike smaks komponentene i kaffen ekstraheres over tid under brygging, teoretisk. Den skal gi en innsikt i hvordan smaken i en kopp espresso utvikler seg, avhengig av hvor lenge bryggingen varer. I virkeligheten vil dette varier da mye med typen kaffe, og ikke minst hvor lenge bønnene har stått eksponert for oksygen, men det er et eget tema i seg selv.

3.4 Plot 4

Plot 4(4) viser da en fancy 3D graf som skal da vise oss den totale sammenhengen mellom kverningsgrad, tempratur og tid. Det man ser her er at en høy tempratur og finere kverningsgrad skal gi oss ett høyere utbytte.

4 Volum av en Donut

Da var det tid for del 2 av oppgaven, volumet av en donut

Vi lærte jo om omdreiningslegemer i R2 for ett par år siden da jeg gikk der, så for å tørke av litt støv av den tankegangen prøver jeg meg nå på å utlede volumet av en donut, eller da en Torus som det da heter. Men jeg vil prøve på en måte som implementerer litt tankegang fra det jeg har lært på Gløshaugen (Kanskje mest fra matte 1).

1. Parametrisering av en Torus

En torus med stor radius R (avstanden fra sentrum av røret til torusens sentrum) og liten radius r (radiusen på selve røret) kan parametriseres slik:

$$\vec{X}(\theta,\phi) = \begin{bmatrix} (R + r\cos\theta)\cos\phi \\ (R + r\cos\theta)\sin\phi \\ r\sin\theta \end{bmatrix}, \quad \theta,\phi \in [0, 2\pi]$$

Dette representerer en overflate som roteres rundt z-aksen.

2. Overflateelement ved hjelp av Jacobimatrisen

Vi finner partiellderiverte med hensyn på θ og ϕ:

$$\frac{\partial \vec{X}}{\partial \theta} = \begin{bmatrix} -r\sin\theta\cos\phi \\ -r\sin\theta\sin\phi \\ r\cos\theta \end{bmatrix}, \quad \frac{\partial \vec{X}}{\partial \phi} = \begin{bmatrix} -(R+r\cos\theta)\sin\phi \\ (R+r\cos\theta)\cos\phi \\ 0 \end{bmatrix}$$

Vi bruker kryssproduktet for å finne arealelementet:

$$\vec{A} = \left\| \frac{\partial \vec{X}}{\partial \theta} \times \frac{\partial \vec{X}}{\partial \phi} \right\| d\theta d\phi$$

Beregner kryssproduktet:

$$\left\| \frac{\partial \vec{X}}{\partial \theta} \times \frac{\partial \vec{X}}{\partial \phi} \right\| = r(R + r\cos\theta)$$

3. Volumet via overflateintegrasjon

Torusen kan tenkes som summen av små sylindere, og vi bruker:

$$V = \iint_{\theta=0}^{2\pi} \iint_{\phi=0}^{2\pi} \left\| \frac{\partial \vec{X}}{\partial \theta} \times \frac{\partial \vec{X}}{\partial \phi} \right\| d\theta d\phi$$

Sett inn verdien:

$$V = \int_0^{2\pi} \int_0^{2\pi} r(R + r\cos\theta)\, d\theta\, d\phi$$

4. Utregning

Først integrerer vi over θ:

$$\int_0^{2\pi} r(R + r\cos\theta)\, d\theta = rR \int_0^{2\pi} d\theta + r^2 \int_0^{2\pi} \cos\theta\, d\theta = 2\pi rR$$

Så integrerer vi over ϕ:

$$\int_0^{2\pi} d\phi = 2\pi$$

Dermed:

$$V = 2\pi rR \cdot 2\pi = 4\pi^2 Rr$$

Volumet av en Donut med radius R og rørradius r er:

$$\boxed{V = 4\pi^2 Rr^2}$$

Dette så da nesten penere ut enn min donut, hvis alt har gått for seg rikitg...

5 Konklusjon

Yes, da var man her da. Da det gjelder delen om espresso fikk jeg vist fram en viss sammenheng, hvertfall forkalrt litt av tanken og fårr illistret plottene. Men dette var en veldig forenklet modell med en del antagleser, som nevnt i rapporten er det mye som vil være mer teoretisk enn praktisk. Men jeg fikk alikavel brukt flere ting jeg har lært fra ulike emner for å lage en enkel modell, selv om matten kanskje kunne vært litt mer komplisert.

Jeg tror alt skal være riktig med donut delen, hvis det er feil så kan man jo selvfølgelig bruke samme metoden som fra R2...

Men jeg føler jeg lærte mye av dette, både om kaffe, og ikke minst matte.

5.1 Folk som virkelig vet hva de holder på med

https://sites.dmi.uns.ac.rs/ecmimw2018/reports/Report$_{G}roup_{1}0_{E}spresso_{C}offee_{P}roblem.pdf$

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0219906sec018

Figure 1:

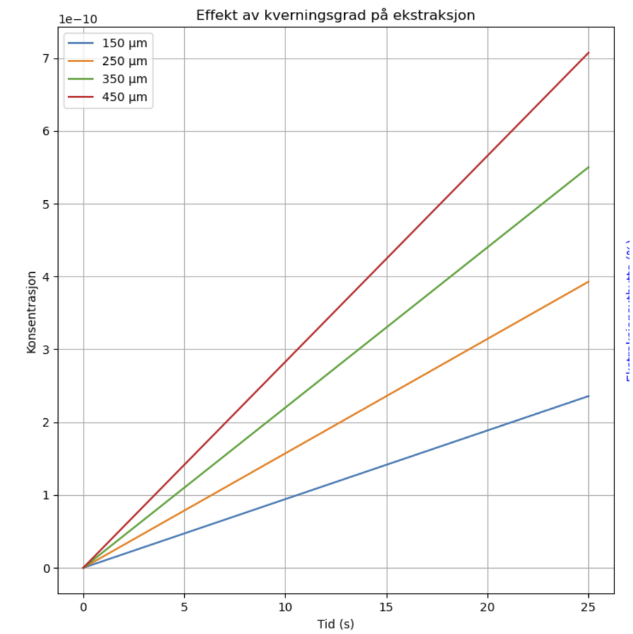

Figure 2:

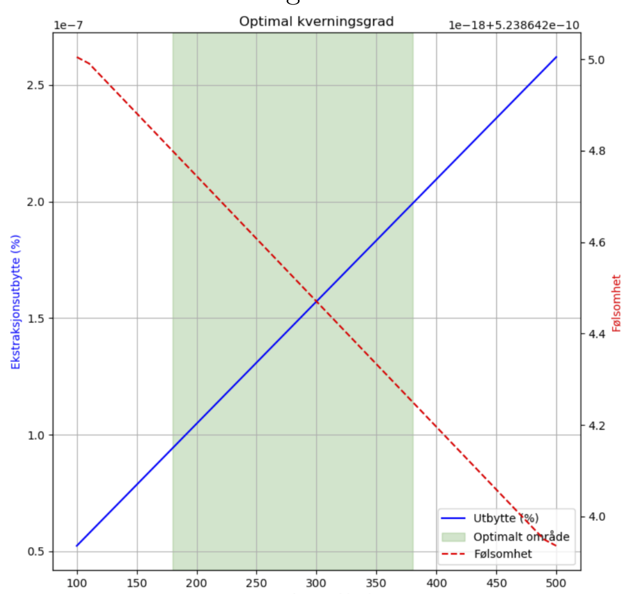

Figure 3:

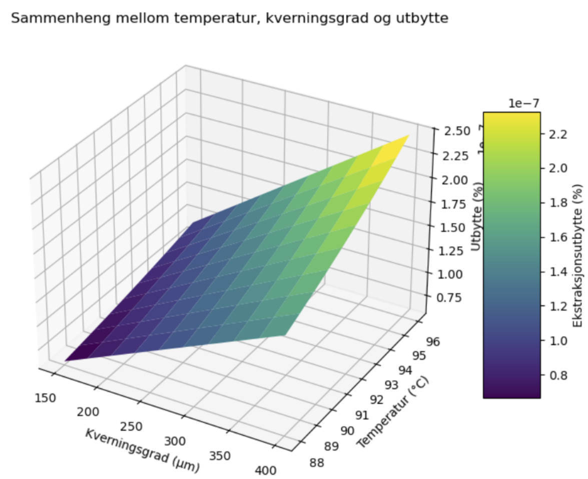

Figure 4:

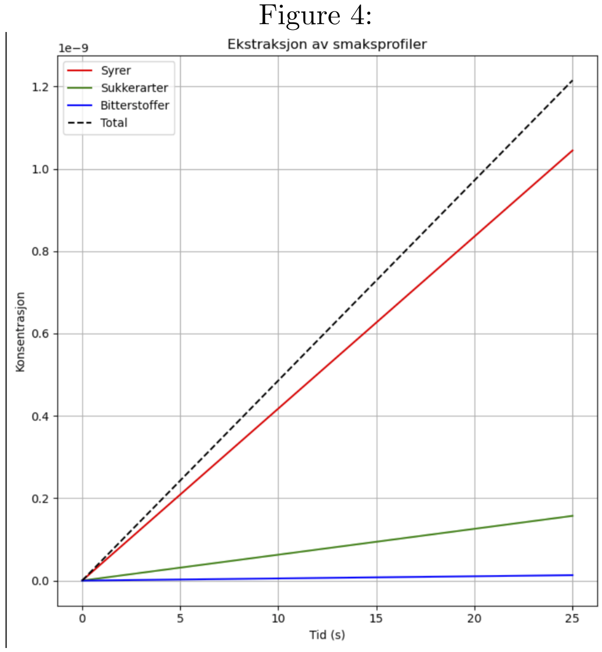

Figure 5: Her er en latte(Laget av meg) som demonstrerer min kjærlighet til matte