

Livsløpsvurdering (LCA) — arbeidskrav

Denne notebooken inneholder oppgaver som dekker følgende i læreboka (Bakshi: Sustainable engineering):

- **Chapter 9: Inventory analysis**
- **Chapter 10.1: Mathematical framework / Process networks**

Grunnleggende koder finner dere i [Python cheat sheet \(files/Python_cheat_1.1.pdf\)](#).

Det er laget treningsoppgaver (TG'er) for alle kapitlene vi vil gå gjennom. Dere finner de her i Jupyterhub under *Assignments*, der følgende er relevante for dette arbeidskravet:

- 0-pybasics-tg (bibliotek av koder dere kan bruke i oppgavene, tips: dere vil finne det dere trenger i Cheat sheet'en)
- 1-lci-tg-v2.1 (flere eksempeloppgaver fra læreboka *kapittel 9*)
- 2-mathframe-tg-v2.1 (flere eksempeloppgaver fra læreboka *kapittel 10.1*)

Bruk gjerne disse om dere synes noen av delene er vanskelige. TG'ene har fortløpende retting slik at dere kan sjekke svaret for hver oppgave underveis. De har også lenker til videoer som forklarer matematikken i de enkelte oppgavene.

Dette arbeidskravet er satt opp med autogenerert retting hver 30 minutt. Merk at noen av tallene er randomisert.

In [1]:



```
1 # Kjør cellen nedenfor før du starter oppgavene
2 import numpy as np
3 import os, imp
4
5 path = '.meta.py'
6 name = path.split('.')[ -2]
7 with open(path, 'rb') as fp:
8     meta = imp.load_module(name, fp, path, ('.py', 'rb', imp.PY_SOURCE))
9
10 import meta
11 assignments = meta.assignments()
```

Problem 1 (40 poeng): matriser i LCA

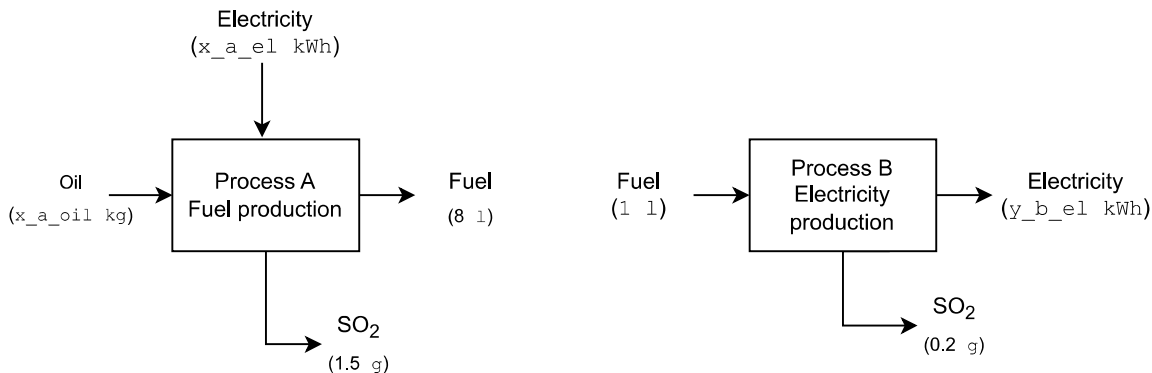
Kjør cella under for å generere oppgaven (klikk cella og velg "Run")

In [2]:

```
1 assignments.getAssignment('assignment_1')
```

Out[2]:

Legg til grunn de to prosessene som er vist under. Prosess A bruker $x_{a_oil} = 5$ kg olje og $x_{a_el} = 7$ kWh elkraft for å produsere 8 liter drivstoff, med et utslipp på 1.5 g SO_2 . Prosess B krever 1 liter drivstoff for å levere $y_{b_el} = 1$ kWh elektrisk kraft, med et utslipp av 0.2 g SO_2 .



Systemet leverer samlet 1140 kWh elkraft til sluttforbruk (altså *final demand*).

a) (10 poeng) Definer teknologimatrisa (*technology matrix*) **A** for systemet, bruk følgende format:

$$A = \begin{bmatrix} a_{processA,fuel} & a_{processB,fuel} \\ a_{processA,el} & a_{processB,el} \end{bmatrix}$$

der vi har satt *fuel* = drivstoff, og *el* = elkraft. Dere må altså plassere rett tall på riktig sted i matrisa

Definer svaret som en Python-variabel **A**. Det må være definert som en `np.array`.

In [3]:

```
1 # GOODGOOD
2 A = np.array([[8, -1], [-7, 1]])
3 print(A)
```

```
[[ 8 -1]
 [-7  1]]
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om det er oppgitt som *np.array* og riktig dimensjon).

In [4]:

```
1 assert type(A) == np.ndarray, 'Variabelen A skal være laget som np.array()'
2 assert np.shape(A) == (2,2), 'Dimensjonene til matrisa A skal være (2,2)'
```

b) (5 poeng) Definer sluttforbruksvektoren (*final demand vector*) **f** for systemet, bruk følgende format:

$$f = \begin{bmatrix} f_{fuel} \\ f_{el} \end{bmatrix}$$

Definer svaret som en Python-variabel `f`. Det må være definert som en `np.array`.

In [5]:

Run

```
1 # GOODGOOD
2 f = np.array([[0],[1140]])
3 print(f)
```

```
[[ 0]
 [1140]]
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om det er oppgitt som `np.array` og riktig dimensjon).

In [6]:

```
1 assert type(f) == np.ndarray, 'Variabelen f skal være laget som np.array()'
2 assert np.shape(f) == (2,1), 'Dimensjonene til vektoren f skal være (2,1)'
```

c) (10 poeng) Definer intervensjonsmatrisa (*intervention matrix*) B for systemet, bruk følgende format.

$$B = \begin{bmatrix} B_{processA,oil} & B_{processB,oil} \\ B_{processA,SO_2} & B_{processB,SO_2} \end{bmatrix}$$

Bruk negative verdier for innstrømmer (*input av olje*, siden dette er uttak av naturressurser), og positive verdier for utslipp av SO_2 . Bruk enheter kg for uttak av råolje og gram for SO_2 -utslipp.

Definer svaret som en Python-variabel `B`. Det må være definert som en `np.array`.

In [7]:

Run

```
1 # GOODGOOD
2 B = np.array([[-5, 0], [1.5, 0.2]])
3 print(B)
```

```
[[ -5.   0. ]
 [ 1.5  0.2]]
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om det er oppgitt som `np.array` og riktig dimensjon).

In [8]:



```
1 assert type(B) == np.ndarray, 'Variabelen B skal være laget som np.array()'
2 assert np.shape(B) == (2,2), 'Dimensjonene til matrisa B skal være (2,2)'
```

d) (10 poeng) Regn ut hvor mye olje (*oil*, kg) og SO₂-utslipp (gram) som behoves for å levere et **sluttforbruk på 1 liter drivstoff** (*final demand of 1 liter fuel*). Bruk følgende format:

$$r = \begin{bmatrix} r_{oil} \\ r_{SO_2} \end{bmatrix}$$

Definer svaret som en Python-variabel `r`. Ved matrisemultiplikasjon med arrays vil det av seg selv være et `np.array`.

In [9]:



Run

```
1 # Skriv svaret ditt her 😊
2 g = np.array([[1],[0]])
3 s = np.linalg.inv(A)@g
4 print(s)
5 r = B@s
6 print(r)
```

```
[[1.]
 [7.]]
[[-5. ]
 [ 2.9]]
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om det er oppgitt som *np.array* og riktig dimensjon).

In [10]:



```
1 assert type(r) == np.ndarray, 'Variabelen r skal være et np.array()'
2 assert np.shape(r) == (2,1), 'Dimensjonene til vektoren r skal være (2,1)'
```

e) (5 poeng) Beregn hvor mye drivstoff som behøves for å levere 1 kWh elkraft. Legg merke til at vi ber om *drivstoff* (fuel), ikke olje.

Definer svaret som en Python-variabel `f_fuel_perkWh`.

In [11]:



Run

```
1 # GOOOOOOD
2 f_fuel_perkWh = 1
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om det er oppgitt som *tall*, ikke array).

In [12]:

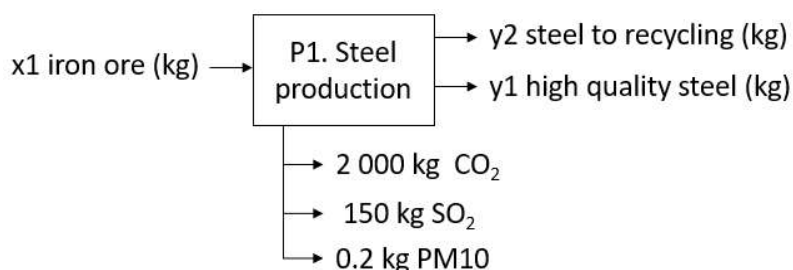


```
1 assert (type(f_fuel_perkWh) in [int, float, np.int64, np.float64, np.ndarray]), 'Sva
```

Problem 2 (20 poeng): allokering for en prosess

En produsent av høykvalitetsstål begynner å sortere eget produksjonsavfall som så sendes til gjenvinning (steel to recycling). De har dermed to produkter som de selger:

- sortert produksjonsavfall (steel to recycling)
- komponenter i høykvalitetsstål (high quality steel).



For å beregne utslipp per kg produsert høykvalitetsstål må vi gjøre en allokering av innsatsmidler og utslipp mellom de to produktene.

Kjør cella under for å generere oppgaven (klikk på cella og velg "Run")

In [13]:



```
1 assignments.getAssignment('assignment_2')
```

Out[13]:

Legg til grunn at produksjonen krever $x1 = 2000$ kg med jernmalm for å produsere $y1 = 1100$ kg høykvalitetsstål og $y2 = 220$ kg utsortert stål til gjenvinning.

Du er i tillegg gitt følgende informasjon om enhetsverdien av de to produktene:

- $y1_{econ} = 1000$ kr per kg høykvalitetsstål
- $y2_{econ} = 110$ kr per kg stål til gjenvinning.

a) (10 poeng) Regn ut **andelen** av utslippene som allokeres til høykvalitetsstål (high quality steel) med bruk av *masseallokering* og *økonomisk allokering*.

Definer svaret som Python-variabler `x_mass_steel` og `x_econ_steel`.

In [16]:



Run

```
1 # Skriv svaret ditt her 😊
2 x1 = 2000
3 y1 = 1100
4 y2 = 220
5 y1e = 1000
6 y2e = 110
7
8 x_mass_steel = y1/(y1+y2)
9 x_econ_steel = y1*y1e/(y1*y1e+y2*y2e)
10 print(x_mass_steel, "\n", x_econ_steel)
```

```
0.8333333333333334
0.9784735812133072
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om svarene er oppgitt som *tall*, ikke *array*).

In [17]:



```
1 assert all (type(i) in [int, float, np.int64, np.float64]
2             for i in [x_mass_steel, x_econ_steel]), 'Svarene skal oppgis som tall'
```

b) (10 poeng) Bruk *økonomisk* allokering og beregn utslippene av CO_2 , og SO_2 og PM_{10} fra produksjon av y_1 kg høykvalitetsstål.

Merk at vi ber om utslippene *per kg stål*, ikke for prosessen samlet.

Definer svaret som Python-variabler `steel_hq_CO2`, `steel_hq_SO2` og `steel_hq_PM10`.

In [18]:



Run

```
1 # Skriv svaret ditt her 😊
2 CO2 = 2000
3 SO2 = 150
4 PM10 = 0.2
5
6 steel_hq_CO2 = CO2 * x_econ_steel / y1
7 steel_hq_SO2 = SO2 * x_econ_steel / y1
8 steel_hq_PM10 = PM10 * x_econ_steel / y1
9 print(steel_hq_CO2, "\n", steel_hq_SO2, "\n", steel_hq_PM10)
```

```
1.7790428749332858
0.13342821561999643
0.0001779042874933286
```

Kjør cella under for å sjekke at formatet er riktig (sjekker om svarene er oppgitt som *tall*, ikke *array*).

In [19]:

```
1 assert all (type(i) in [int, float, np.int64, np.float64]
2             for i in [steel_hq_CO2, steel_hq_SO2, steel_hq_PM10]), 'Svarene skal oppgitt som tall'
```

Problem 3 (40 poeng): Per Olsens bondegård

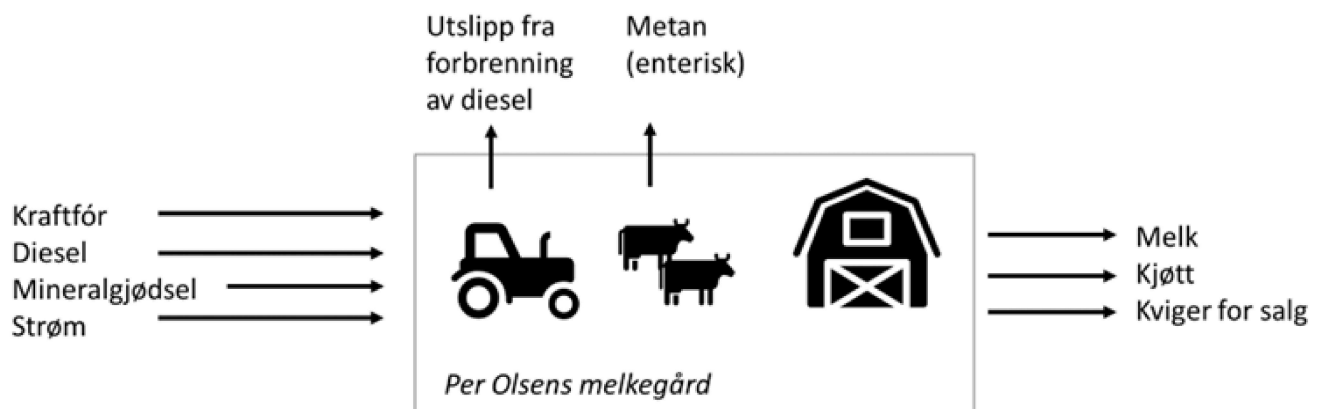
Denne oppgaven vil være tema for flipped classroom om allokering

Oppgaven lages ved å kjøre cella under (klikk cella og trykk "Run")

Et gårdsbruk er et eksempel på et produksjonssystem med mange produkter. I dette tilfellet har Per Olsen et melkebruk, det vil si gården produserer melk, melkekyr og storfe til slakt. Hvis man ønsker å sammenligne ett av disse produktene med et alternativ er man avhengig av å allokere innsatsmidler og utslipp mellom de tre produktene, for eksempel om man vil sammenligne klimautslippene fra storfe-kjøtt med utslipp fra svinekjøtt, eller utslippene per liter melk fra en typisk melkegård som Per Olsens gård med melk fra økologisk landbruk (som kanskje er et integrert bruk med også grønnsaksproduksjon) eller et større melkebruk der kjøttproduksjonen er av mindre betydning. For å sammenligne på tvers av disse må regne klimautslippene per enhet produkt og dermed allokere prosessinnsats og prosessutslipp mellom produktene.

Tallene er hentet fra en norsk LCA-studie fra 2013 der Bioforsk og NTNU vurderte arketyper for norske gårdsbruk. Tabellene under summerer noen av tallene for et midtnorsk melkebruk. Vi gjør mange forenklinger, blant annet har vi ikke med utslipp av N₂O og metan fra kugjødelse og kunstgjødelse.

Her følger et forenklet flytskjema for Per Olsens gård. Gården drives som ett samlet system så det er ikke mulig å skille ut de enkelte delene av produksjonen. Melkeproduksjonen er avhengig av kjøttproduksjon, og motsatt, og hele gården er derfor å betrakte som én prosess som vi må dele i tre siden vi har tre produkter.



Oppgavebeskrivelse

Oppgaven går ut på å regne ut klimautslippene per liter melk. Vi skal bruke økonomisk allokering, altså dele utslippene basert på verdien av de ulike strømmene og deretter regne utslipp per enhet produkt. Matematikken for dette er beskrevet i Bakshi kap 10.1.2, som «partitioning» side 182-184.

Detaljert prosedyre finnes i [Oppgaveveiledningen \(files/PerOlsen.pdf\)](#), kort fortalt skal vi:

- Definere allokeringskoeffisientene w_{milk} , w_{meat} , w_{heif} basert på den økonomiske verdien av de ulike strømmene og bruke denne i teknologimatrise for å dele opp gården i tre prosesser (*heifer* er kviger på engelsk).
- Sette opp teknologimatrise A for systemet med tre ulike produkter fra gården
- Definere intervensjonsmatrisa B og også her dele opp utslippene mellom de tre prosessene for gården

- Definere sluttforbruk (final demand, f), der dette er definert som 1 liter melk
- Beregne utslipp per liter melk, som $R = BA^{-1}f$
- Til slutt sammenligne dette med å drikke noe annet, for eksempel fruktjuice fra Malta som har et utslipp på 0.4-1.2 kg CO₂e per liter avhengig av hva det er laget av .

Kjør cella under for å generere tallene dere skal bruke.

In [20]:



```
1 assignments.getAssignment('assignment_3')
```

Out[20]:

Følgende tabeller beskriver den årlige aktiviteten på Per Olsens bondegård. Disse tallene er også oppgitt i *Oppgaveveiledningen*.

Årlige utslipp og CO₂-ekvivalenter

Utslipp	Årsak	Mengde per år	CO ₂ -ekvivalenter per kg
CO ₂	Dieselforbrenning	9980 kg CO ₂	1 kg CO ₂ e/kg
CH ₄	Enteriske utslipp	5270 kg CH ₄	34 kg CO ₂ e/kg

Årlig forbruk av innsatsmidler, med CO₂-ekvivalenter fra produksjon av hver enkelt

Input	Inneholder	Mengde per år	Produksjonsutslipp
Kraftfor	Produksjon	60000 kg	0.63 kg CO ₂ e/kg
Diesel	Produksjon	3900 liter	0.25 kg CO ₂ e/liter
Kunstgjødsel	Produksjon	3195 kg	1.88 kg CO ₂ e/kg
Strøm	Produksjon	26000 kWh	0.05 kg CO ₂ e/kWh

Årlig produksjon fra Per Olsens gård. Kjøtt er både kviger og okser

Produkt	Mengde per år	Enhetsverdi
Melk	133000 liter	4 kr/liter
Kjøtt, slaktevekt	5430 kg	40 kr/kg
Kviger for salg	1 stk	11250 kr/stk

Når dere regner allokering ut fra massen for hvert produkt kan dere legge til grunn at en kvige har en slaktevekt på rundt 300 kg.

Riktig svar utløses ved at dere regner ut klimautslippene per liter melk og oppgir denne som CO_{2melk} i enhet [kg CO₂e/liter] ut fra en økonomisk allokering.

In [27]:



Run

```

1  # Skriv svaret ditt her 😊
2  milk = 133000 # l
3  meat = 5430 # kg
4  heif = 1 # stk
5
6  price_milk = 4
7  price_meat = 40
8  price_heif = 11250
9
10 flow_milk = milk * price_milk
11 flow_meat = meat * price_meat
12 flow_heif = heif * price_heif
13
14 tot_flow = flow_milk + flow_meat + flow_heif
15
16 w_milk = flow_milk / tot_flow
17 w_meat = flow_meat / tot_flow
18 w_heif = flow_heif / tot_flow
19
20 food = -60000
21 fuel = -3900
22 fert = -3195
23 el = -26000
24
25 CO2e_food = 0.63
26 CO2e_fuel = 0.25
27 CO2e_fert = 1.88
28 CO2e_el = 0.05
29
30 CO2 = 9880 * 1
31 CH4 = 5270 * 34
32 U = CO2 + CH4
33
34 A = np.array([[1, 0, 0, 0, food*w_milk, food*w_meat, food*w_heif],
35               [0, 1, 0, 0, fuel*w_milk, fuel*w_meat, fuel*w_heif],
36               [0, 0, 1, 0, fert*w_milk, fert*w_meat, fert*w_heif],
37               [0, 0, 0, 1, el*w_milk, el*w_meat, el*w_heif],
38               [0, 0, 0, 0, milk, 0, 0],
39               [0, 0, 0, 0, 0, meat, 0],
40               [0, 0, 0, 0, 0, 0, heif]])
41
42 B = np.array([CO2e_food, CO2e_fuel, CO2e_fert, CO2e_el, U*w_milk, U*w_meat, U*w_heif])
43
44 f = np.array([0, 0, 0, 0, 1, 0, 0])
45
46 R = B@np.linalg.inv(A)@f
47 CO2melk = R
48 print(CO2melk)
49

```

1.2368550200539157

Kjør cella under for å sjekke at formatet er riktig

In [28]:



```
1 assert (type(CO2melk) in [int, float, np.int64, np.float64]), 'Svaret skal være et ei
```

Bonusoppgave: Når vi gjør slike analyser for bedrifter er det ikke alltid lett å få tak i priser for alle produkter, slik at vi kanskje må bruke masseallokering. For å bli bedre kjent med allokering og hva det kan bety for resultatene forsøk å svare på følgende spørsmål:

- Gjør samme beregning med masse-basert allokering. Har dette noe å si for klimautslippene per liter melk? (Anta for enkelthets skyld at en kvige for salg veier 300 kg.)
- Forsøk å regne med en annen melkepris, for eksempel 4.50 kr per liter. Hvordan påvirker dette utslippene til melk? Hva skjer med de beregnede utslippene per kg kjøtt? Kjør modellen deres med ulike priser og sluttforbruk.
- Kan dere tenke dere andre «vitenskapelige» enheter som kan brukes til å fordele utslippene? Her kan man for eksempel tenke hva som er *funksjonen* til melk, eller mat generelt.
- I hvilken grad påvirkes sammenligningen med juice av valg av allokeringsprinsipp?

[Oppgaveveiledningen \(files/PerOlsen.pdf\)](files/PerOlsen.pdf) har også noen spørsmål til refleksjon for prosjektoppgaven dere skal gjøre.