

Eksamen Jupyter notebook

Denne delen representerer 24% av eksamen.

Pass på å levere eksamensbesvarelsen ved å trykke "submit", på samme måte som du har gjort i de andre innleveringene i emnet. Husk også på å svare ut spørsmålet i Inspira hvorvidt du har levert eller ikke.

Vi har ikke mulighet å gi poeng for oppgavesett som ikke er levert ("submit") innenfor eksamenstiden, inkludert utvidet eksamenstid for de det gjelder.

Opplever du problemer med å submit, vær vennlig og vent 5 minutt og prøv igjen. Vi vil følge overbelastning på server og utvide innleveringstiden om nødvendig.

Hvis dere gjør beregninger utenom JupyterHub må dere legge merke til at rettingen legger til grunn rett svar innenfor minst 2 siffer etter komma. Det vil for eksempel si at dere må skrive svaret som `num_a = 1.11` og ikke `num_a = 1.1`.

Kjør cellen nedenfor før du starter eksamen

```
In [1]: ▶ import numpy as np
import os, imp

def import_meta(path):
    name = path.split(os.path.sep)[-1].split('.')[ -2]
    with open(path, 'rb') as fp:
        return imp.load_module(name, fp, path, ('.py', 'rb', imp.PY_SOURCE))

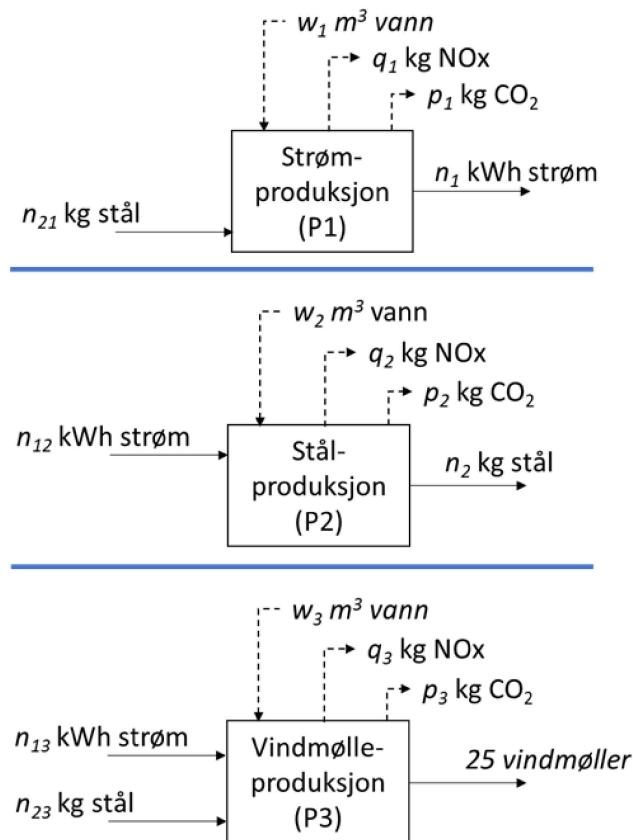
meta = import_meta('.meta.py')
from meta import assignments
```

Oppgave 1: LCA for produksjon av vindmølle

Kjør cellen nedenfor for å generere oppgaven

```
In [2]: from meta import assignments
assignments().getAssignment(assignment_num=1)
```

Out[2]: Ta utgangspunkt i følgende tre prosesser, henholdsvis produksjon av strøm, stål og vindmøller. Mer informasjon om de enkelte prosessene følger under flytskjema.



Informasjon til oppgaven:

- Prosess 1 krever $n_{21} = 3000$ kg stål for å produsere $n_1 = 98000$ kWh strøm, og dette innebærer et vannforbruk på $w_1 = 14700$ m³ vann og et direkteutslipp fra P1 på $q_1 = 93$ kg NO_x og $p_1 = 4900$ kg CO₂.
- Prosess 2 (P2) krever $n_{12} = 98000$ kWh strøm fra P1 for å produsere $n_2 = 140000$ kg stål. I produksjon av dette stålet er det et vannforbruk på $w_2 = 0$ m³ vann og et utslipp på $q_2 = 255$ kg NO_x og $p_2 = 13400$ kg CO₂.
- For at P3 skal levere 25 vindmøller, krever prosessen innsats av $n_{13} = 14000$ kWh strøm og $n_{23} = 140000$ kg stål, og et forbruk av $w_3 = 2800$ m³ vann, og et utslipp på $q_3 = 285$ kg NO_x og $p_3 = 15000$ kg CO₂.

a) (5 poeng) Ferdigstill teknologimatrisen for prosessnettverket ved å fylle inn de tomme cellene i matrisa under (technology matrix, kalt A) ut fra den informasjonen du er gitt. Behold rekkefølge i rader og koloner. Angi svaret som matrisen A .

	P1	P2	P3
Strøm (kWh)	n_1	?	?
Stål (kg)	$-n_{21}$?	?
Vindmøller (stk)	0	?	25

Husk at for å lage en 3x3 matrise bruker du `A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])` . Du kan se på matrisen ved å skrive `print(A)` .

```
In [3]: ▶ n21 = 3000 # randomisert
n12 = 98000 # randomisert
n23 = 140000 # randomisert
n13 = 14000 # randomisert
n1 = n12
n2 = n23
A = np.array([[n1, -n12, -n13], # strom
              [-n21, n2, -n23], # stal
              [0, 0, 25]]) # vindmoller
```

Kjør cellen nedenfor for å sjekke formatering

```
In [4]: ▶ assert type(A) == np.ndarray, 'Svaret må være en np.array() matrise'
assert np.shape(A) == (3,3), 'Matrisen må ha formen (3,3)'
```

b) (5 poeng) Ferdigstill intervensjonsmatrisen for prosessnettverket ved å fylle inn de tomme cellene i matrisa under (intervention matrix, kalt B). Merk at ressursuttak noteres som negative verdier. Behold rekkefølge i rader og koloner. Angi svaret som matrisen B .

	P1	P2	P3
Vann (m ³)	$-w_1$?	?
NO _x (kg)	?	q_2	?
CO ₂ (kg)	?	?	15000

```
In [5]: w1 = 14700 # randomisert
q1 = 93 # randomisert
p1 = 4900 # randomisert
w2 = 0 # randomisert
q2 = 255 # randomisert
p2 = 13400 # randomisert
w3 = 2800 # randomisert
q3 = 285 # randomisert
p3 = 15000 # randomisert
B = np.array([[-w1, -w2, -w3], #vann
              [q1, q2, q3], #nox
              [p1, p2, p3]]) #co2
```

Kjør cellen nedenfor for å sjekke formatering

```
In [6]: assert type(B) == np.ndarray, 'Svaret må være en np.array() matrise'
assert np.shape(B) == (3,3), 'Matrisen må ha formen (3,3)'
```

c) (5 poeng) Bruk teknologimatrissa A og intervensjonsmatrissa B til å regne ut livsløpsutslippene av NOx forbundet med produksjon av 1 vindmølle. Angi svaret som variabelen `nox_vindmølle`.

Merk at vi spør etter en variabel og ikke vektoren `r`. Nyttige funksjoner for denne oppgaven er:

- Opprette en kolonnevektor: `f = np.array([[1], [2], [3]])`
- Matrisemultiplikasjon: `A@B`
- Inverstransformasjon (A^{-1}): `np.linalg.inv(A)`

Hver plass i en matrise har et tilegnet nummer. For en 2x2 matrise definert som `L = np.array([[11, 12], [13, 14]])` er for eksempel `L[0,0] = 11`, `L[0,1] = 12` osv.

```
In [7]: f = np.array([[0], [0], [1]])
r = B@np.linalg.inv(A)@f
nox_vindmølle = r[1,0]
```

Kjør cellen nedenfor for å sjekke formatering

```
In [8]: assert (type(nox_vindmølle) in [int, float, np.int64, np.float64]), 'Svaret m
```

Oppgave 2: Materialstrømsanalyse for bruer

Kjør cellen nedenfor for å generere oppgaven

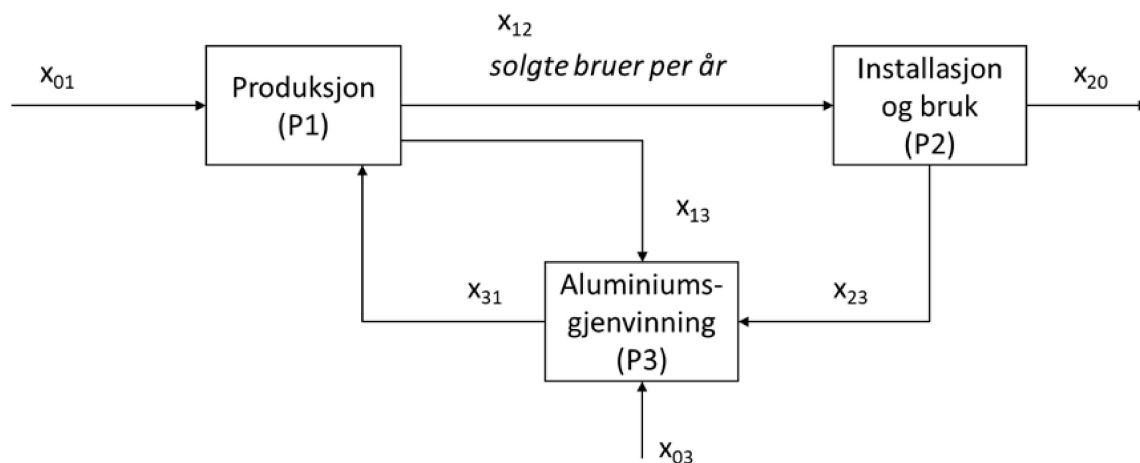
```
In [9]: ▶ from meta import assignments
assignments().getAssignment(assignment_num=2)
```

Out[9]: Vi skal i denne oppgaven se på materialstrømmene for produksjonen av bruer. Det er et tenkt system for en norsk bruprodusent som lager kassebruer i aluminium. Bruene er designet for å fungere som midlertidige installasjon i påvente av oppgradering eller i forbindelse med veiutbygging. Bruene har ulik dimensjon, men vi antar her at de er i en «standard» brystørrelse for tofelts-veg (8 meter bred) og samlet spenn på 30 meter.

Vi begynner med prosessen for produksjon (P1), som kan antas å levere 68 slike standard-bruer hvert år fra produksjon (P1) til installasjon (P2). I hver bru er det 29 tonn aluminium. Videre er det gitt at:

- 81% av aluminium inn i P1 ender som material i en bru som blir solgt (til P2). Resten av aluminium i P1 sorteres og sendes til gjenvinning (i P3).
- I produksjonen av bruer benyttes det 50% gjenvunnet material (fra P3), og resten er import til systemet av nyprodusert aluminium.

Vi legger til grunn at det ikke er akkumulering i systemet og du kan anta at materialsammensetningen er lik for alle utstrømmer fra P1. Et flytskjema for aluminiumsstrømmene er vist under.



a) (5 poeng) Hvor mye nytt aluminium trenger produsenten i P1 årlig? Dette er i flytskjema identifisert som x_{01} .

Angi svaret med variabelen m_{ny} , i enheter tonn/år.

```
In [10]: ▶ M = 29      # randomisert
n = 68      # randomisert
alu = 0.81 # randomisert
m_ny = M*n/alu*0.5
```

Kjør cellen nedenfor for å sjekke formatering

```
In [11]: ▶ assert (type(m_ny) in [int, float, np.int64, np.float64]), 'Svaret må være et
assert (m_ny > 0), 'Svaret må være en positiv verdi'
```

b) (4 poeng) Ferdigstill balanseligningene for de tre prosessene ved å fylle inn de tomme cellene i matrisa under.

Uttrykk (definer) svaret gjennom matrisen `A_bal` . Bruk rekkefølger i kolonner og rader som gitt i følgende matrise, og merk at noen tallverdier er ført inn allerede.

	x01	x12	x13	x31	x20	x23	x03
P1	1	?	?	?	?	?	?
P2	?	1	?	?	?	?	?
P3	?	?	1	?	?	?	?

```
In [12]: ▶ A_bal = np.array([[1, -1, -1, 1, 0, 0, 0], #P1
                             [0, 1, 0, 0, -1, -1, 0], #P2
                             [0, 0, 1, -1, 0, 1, 1]]) #P3
```

Kjør cellen nedenfor for å sjekke formatering

```
In [13]: ▶ assert type(A_bal) == np.ndarray, 'Svaret må være en np.array() matrise'
assert np.shape(A_bal) == (3,7), 'Matrisen må ha formen (3,7)'
```

```
In [ ]: ▶
```