

Materialstrømsanalyse (MFA) — arbeidskrav

Denne notebooken inneholder oppgaver som dekker *materialstrømsanalyse* i pensum:

- **Chapter 12.4 Material flow analysis** (læreboka til Bakshi: Sustainable engineering)
- **Notat om materialstrømsanalyse** (dekker spesielt matematikken som vi bruker her)

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

Det er også laget treningsoppgaver (TG'er) for MFA. Dere finner denne her i Jupyterhub under *Assignments*:

- 0-pybasics-tg (bibliotek av koder dere kan bruke i oppgavene, tips: dere vil finne det dere trenger i Cheat sheet'en)
- 3-mfa-tg-v2.1 (eksempeloppgaver med utgangspunkt i *Notat om materualstrømsanalyse*)

Bruk gjerne treningsoppgavene om dere synes noen av delene er vanskelige. TG'ene har fortløpende retting slik at dere kan sjekke svaret for hver oppgave underveis.

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()
```

Problemstilling (til Problem 1-6)

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

In [2]:



```
1 assignments.getAssignment("problem_definition")
```

Out[2]:

Oppgaven gjelder aluminiumsstrømmer i drikkevareemballasje, det vil si aluminiumsbokser (aluminium cans). De følgende punktene beskriver forskjellige deler av systemet, hvor dere skal sette informasjonen sammen til en materialstrømsmodell.

Vi ser på et system med følgende elementer:

- Process 1 ($P1$), som produserer aluminiumsplater. I prosessen $P1$ blir 72% av aluminium inn til prosessen (primær og resirkulert) omdannet til aluminiumsplater som så sendes til Process 2, $P2$. Innstrøm av aluminium er en blanding av primæraluminium (ny aluminium) og gjenvunnet aluminium fra Process 5, $P5$. Ny aluminium kommer fra utenfor systemet og er derfor en "import". Tap fra plateproduksjonen i $P1$ sendes til gjenvinning i $P5$.
- Process 2 ($P2$), som produserer drikkebokser. I prosessen $P2$ blir 79% av aluminium i aluminiumsplater omdannet til aluminiumsbokser. Boksene sendes så til tapping i $P3$. Tap fra boksproduksjonen sendes til gjenvinning i $P5$.
- Process 3 ($P3$), som tapper i drikke og selger drikke i bokser. I prosessen $P3$ er det et svinn (tap) på 1 av hver 200 bokser. Tap fra $P3$ sendes til gjenvinning i $P5$, resten selges til forbruk i $P4$.
- Process 4 ($P4$), som er drikkens forbruksfase. I prosessen $P4$ blir en andel på 85% av boksene sortert til gjenvinning i $P5$, mens resten går usortert ut av systemet til generell avfallsbehandling. Den andelen som ikke går til gjenvinning går altså ut av systemet som "eksport".
- Process 5 ($P5$), som gjenvinner aluminium. I prosessen $P5$ blir bokser fra forbruksfase ($P4$), aluminiumsplateproduksjon ($P1$), boksproduksjon ($P2$), og tapping av bokser ($P3$) smeltet om til gjenvunnet aluminium. Samlet for hele prosessen i $P5$ er det et tap på 2.5% som sendes til annen avfallshåndtering og dermed går ut av systemet (som eksport). Gjenvunnet aluminium fra $P5$ sendes til plateproduksjon i $P1$.

Dte mest eav tallgrunlaget her er faktiske tall for Norge, hentet fra masteroppgaven til Andres Tominaga Terukina (2013) og rapporten fra Raadal, Iversen & Modahl (2017).

- Tominaga Terukina, A. (2013). Barriers and solutions for closed-loop aluminium beverage can recycling [Master's thesis, NTNU]. <http://hdl.handle.net/11250/242389> (<http://hdl.handle.net/11250/242389>)
- Raadal, H. L., Modahl, I. S., & Iversen, O. M. K. (2017). Comparison of recycling and incineration of aluminium cans (p. 45). Østfoldforskning. https://norsus.no/wp-content/uploads/or-07-17-alu_final_201017.pdf (https://norsus.no/wp-content/uploads/or-07-17-alu_final_201017.pdf)

Problem 1 – tegn flytskjema

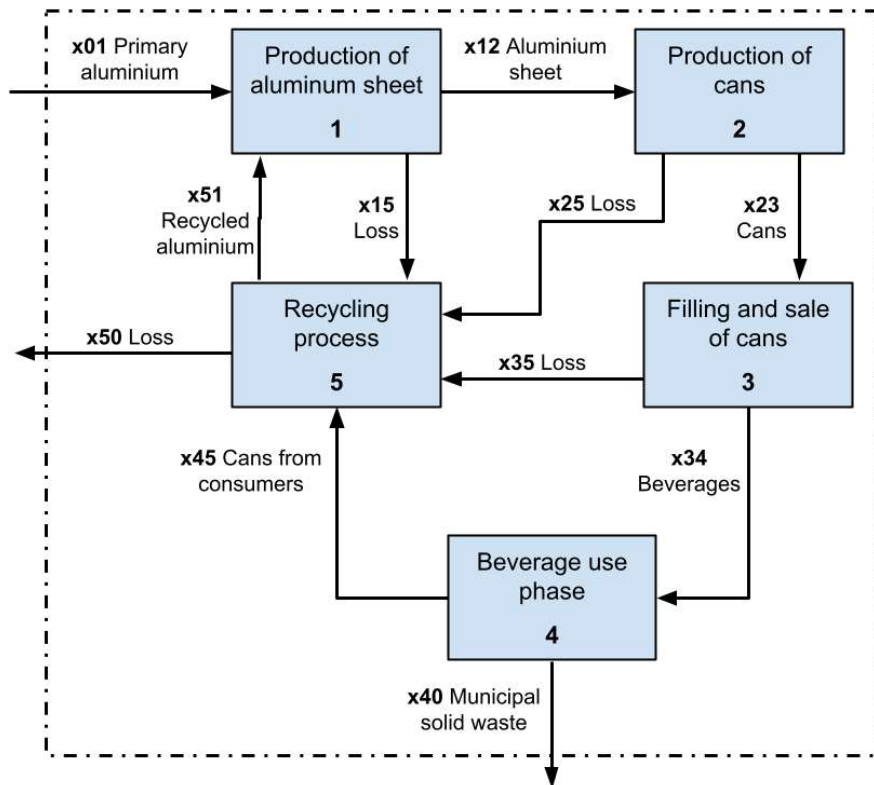
Tegn et flytskjema for systemet. Dette flytskjema skal inneholde alle de fem ulike prosessene, med materialstrømmer mellom prosessene og standard navn (benevnelse) på strømmene. For navngivningen, merk for eksempel at $x12$ betegner en materialstrøm fra prosess 1 til prosess 2, altså mengde aluminiumsplater sent til boksproduksjon.

Dette flytskjema er svært nyttig for å se systemet og hvordan prosessene er koblet med hverandre. Denne oppgaven er ment som trening og har ikke innlevering.

Når du har flytskjema ferdig kan du kjøre cella under og se hva systemet skal se ut som

In [3]:

```
1 from IPython.display import display
2 from IPython.display import Image
3
4 display(Image(filename='files/MFA_Fig.png', width = 500))
```



Problem 2 – massebalanse for en prosess (5 poeng)

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

In [4]:

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



Out[4]:

Sett opp massebalansen for prosess 3 (P_3), altså slik at inn = ut. Plasser innstrømmene til P_3 til venstre for likhetstegnet og utstrømmene til høyre. Bruk x_{ij} (liten x) som navn på strømmen fra prosess i til prosess j , og 0 for å henvise til prosesser som er utenfor systemgrensene.

Angi svaret som variabelen `answer`, som en streng (string) med bruk av apostrofer og uten mellomrom. Se eksempelet under for prosess 1. Du skal altså gjøre det samme for prosess 3.

In [5]:



```
1 # Example answer
2 answer_example = 'x01+x51=x12+x15'
```

In [6]:



Run

```
1 # Skriv svaret ditt her 😊
2 answer = 'x23=x35+x34'
```

Kjør cella under for å sjekke formattering i svaret ditt

In [7]:



```
1 assert type(answer) == str, 'Svaret må oppgis som en streng (string)'
2 assert all (i!=' ' for i in answer), 'Svaret kan ikke inneholde mellomrom'
3
```

Problem 3 – modell-ligninger (10 poeng)

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

In [8]:



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

Out[8]:

Modell-ligninger beskriver matematiske sammenhenger mellom ulike strømmer. Vi skal i denne oppgaven se på en slik sammenheng. Du er gitt informasjon at i prosess 3 (tappingen) så er det slik at "1 av 200 bokser går ut som svinn (tap). Aluminium i dette tapet sendes til gjenvinning i prosess 5 (P_5). Resten av boksene går til forbruk. Bruk denne informasjonen til å regne ut x_{35} som en funksjon av x_{23} .

Oppgi svaret som en Python-funksjon med navn `CalculateFlowX35`. Husk å bruke riktig syntax for å definere funksjoner, det vil si `def`, `return` etc. (se et eksempel i MFA treningsoppgaven (TG), Eksempel 1).

Du skal altså skrive noe slikt som at:

```
def CalculateFlowX35(x23): som betyr at du definerer CalculateFlowX35 som en funksjon av x23
    x35 = sett inn ligning her
    return x35 der du uttrykker svaret ditt, slik at det sjekkes mot løsningsforslag
```

In [9]:



Run

```
1 # Skriv svaret ditt her 😊
2 def CalculateFlowX35(x23):
3     x35 = x23/200
4     return x35
```

Kjør cella under for å sjekke formattering i svaret ditt

In [10]:



```
1 assert (type(CalculateFlowX35(0)) in [int, float, np.int64, np.float64]
2         ), 'Funksjonen må returnere en numerisk variabel'
3
```

Problem 4 – størrelse på strømmene (15 poeng)

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

In [11]:



```
1 assignments.getAssignment("assignment_4")
```

Out[11]:

I Norge drikker vi i gjennomsnitt 50 liter øl, 93 liter brus og 7 liter flaskevann per person per år.

Vi antar her at 100% av øl, 10% av brus og 10% av flaskevann selges i 0.5 L aluminiumsbokser. Hver slik halvlitersboks veier 11 gram.

a) (10 poeng) Regn ut den årlige strømmen av aluminium i aluminiumsbokser som selges. Oppgi svaret som Python-variabelen `m_alucans`, i enhet **kg**/person/år.

In [12]:



Run

```
1 # Skriv svaret ditt her 😊
2 m_alucans = 11e-3*(50+93*0.1+7*0.1)*2
3 print(m_alucans)
```

1.3199999999999998

Kjør cella under for å sjekke formattering i svaret ditt

In [13]:



```
1 assert type(m_alucans) in [int, float, np.int64, np.float64], 'Svaret/svarene må oppgi  
2
```

b) (5 poeng) Den verdien du har regnet som `m_alucans` tilsvarer en av strømmene i Problem 1. Hvilken er det?

Oppgi navnet på denne strømmen med notasjon x_{ij} for en materialstrøm fra prosess i til prosess j , med python-variabelen `flow`. Variabelen skal være en streng (string) uten mellomrom. Et eksempel på et slikt svar er `flow = 'x12'`.

In [31]:



Run

```
1 # Skriv svaret ditt her 😊  
2 # flow = 'x51' #Vet ikke om er riktig  
3 flow = 'x34'
```

Kjør cella under for å sjekke formattering i svaret ditt

In [32]:



```
1 assert type(flow) == str, 'Svaret må være en streng (string)'  
2 assert all (i!=' ' for i in flow), 'Svaret må oppgis uten mellomrom'  
3
```

Problem 5 – løse systemet (10 poeng)

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

In [16]:



```
1 assignments.getAssignment('assignment_5')
```

Out[16]:

Systemet ditt har 5 prosesser, som betyr at du kan sette opp 5 balanseligninger. Til sammen har du 11 strømmer (som er 11 ukjente størrelser). Du trenger 11 ligninger for å løse systemet, det vil si 6 modell-ligninger utover balanseligningene. 5 av disse modell-ligningene kan du sette opp fra den informasjonen du ble gitt i introduksjonen. Den siste ligningen kan du lage med utgangspunkt i svaret ditt fra Problem 4.

Løs systemet og regn ut strømmene som kg/person/år, som du finner som x -vektoren. Du gjør dette ved å definere matriser A og vektor y , og så regne ut x (Se Notat om materialstrømsanalyse for den ligningen du trenger, eller TG'en for MFA). Bruk Python-variablene A , y , og x for å angi de ulike matrisene.

Tips: Matrisene A og y må defineres med `np.array()`. Ligningen for å beregne x vector i Python er vist i TG'en for MFA. Dette skjer ved å først invertere A med bruk av `np.linalg.inv()` og deretter multiplisere inversen med y ved bruk av operatoren `@`.

Merk at rekkefølgen av variablene i A -matrisa må være som følger for at autorettinga skal fungere: x_{01} , x_{12} , x_{15} , x_{23} , x_{25} , x_{34} , x_{35} , x_{40} , x_{45} , x_{50} , x_{51} .

In [17]:



Run

```
1 # Skriv svaret ditt her 😊
2 ''' likninger
3  $x_{01} + x_{51} = x_{15} + x_{12}$ 
4  $x_{12} = x_{25} + x_{23}$ 
5  $x_{23} = x_{34} + x_{35}$ 
6  $x_{34} = x_{40} + x_{45}$ 
7  $x_{15} + x_{25} + x_{35} + x_{45} = x_{50} + x_{51}$ 
8
9  $x_{01} = x_{15} + x_{12} - x_{51}$ 
10  $x_{12} = x_{25} + x_{23}$ 
11  $x_{15} = x_{01} + x_{51} - x_{12}$ 
12  $x_{23} = x_{34} + x_{35}$ 
13  $x_{25} = x_{12} - x_{23}$ 
14  $x_{34} = x_{40} + x_{45}$ 
15  $x_{35} = x_{23} - x_{34}$ 
16  $x_{40} = x_{34} - x_{45}$ 
17  $x_{45} = x_{34} - x_{40}$ 
18  $x_{50} = x_{15} + x_{25} + x_{35} + x_{45} - x_{51}$ 
19  $x_{51} = x_{15} + x_{25} + x_{35} + x_{45} - x_{50}$ 
20
21  $x_{12} = (x_{01} + x_{51}) * 0.72$ 
22  $x_{15} = (x_{01} + x_{51}) * 0.28$ 
23
24  $x_{23} = x_{12} * 0.79$ 
25  $x_{25} = x_{12} * 0.21$ 
26
27  $x_{35} = 0.005 * x_{23}$ 
28  $x_{34} = (1 - 0.005) * x_{23}$ 
29
30  $x_{45} = x_{34} * 0.85$ 
31  $x_{40} = x_{34} * 0.15$ 
32
33  $x_{50} = (x_{15} + x_{25} + x_{35} + x_{45}) * 0.025$ 
34  $x_{51} = (x_{15} + x_{25} + x_{35} + x_{45}) * 0.975$ 
35
36
37
38  $x_{01} + x_{51} = x_{15} + x_{12}$ 
39  $x_{12} = x_{25} + x_{23}$ 
40  $x_{15} = (x_{01} + x_{51}) * 0.28$ 
41  $x_{23} = x_{34} + x_{35}$ 
42  $x_{25} = x_{12} * 0.21$ 
43  $x_{34} = x_{40} + x_{45}$ 
44  $x_{35} = 0.005 * x_{23}$ 
45  $x_{15} + x_{25} + x_{35} + x_{45} = x_{50} + x_{51}$ 
46
47 '''
48 '''
49  $x_{01} = x_{15} + x_{12} - x_{51}$ 
50  $x_{12} = x_{25} + x_{23}$ 
51  $x_{15} = x_{01} + x_{51} - x_{12}$ 
52  $x_{23} = x_{34} + x_{35}$ 
53  $x_{25} = x_{12} - x_{23}$ 
54  $x_{34} = x_{40} + x_{45}$ 
55  $x_{35} = x_{23} - x_{34}$ 
56  $x_{40} = x_{34} - x_{45}$ 
57  $x_{45} = x_{34} - x_{40}$ 
```

```

58 x50 = x15 + x25 + x35 + x45 - x51
59 x51 = x15 + x25 + x35 + x45 - x50
60 ...
61
62
63
64
65
66
67
68
69 A = np.array([[1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 1],
70               [0, 1, 0, -1, -1, 0, 0, 0, 0, 0, 0],
71               [0, 0, 0, 1, 0, -1, -1, 0, 0, 0, 0],
72               [0, 0, 0, 0, 0, 1, 0, -1, -1, 0, 0],
73               [0, 0, 1, 0, 1, 0, 1, 0, 1, -1, -1],
74               [0.72, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0.72],
75               [0, 0.79, 0, -1, 0, 0, 0, 0, 0, 0, 0],
76               [0, 0, 0, 1/200, 0, 0, -1, 0, 0, 0, 0],
77               [0, 0, 0, 0, 0, 0.85, 0, 0, -1, 0, 0],
78               [0, 0, 0.025, 0, 0.025, 0, 0.025, 0, 0.025, -1, 0],
79               [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]])
80 y = np.array([[0], [0], [0], [0], [0], [0], [0], [0], [0], [0], [1.32]])
81 #y = np.transpose(y)
82 x = np.linalg.inv(A)@y
83 print(x)
84 #A = np.array([x01, x12, x15, x23, x25, x34, x35, x40, x45, x50, x51])
[[0.25135842]
 [1.67928249]
 [0.6530543 ]
 [1.32663317]
 [0.35264932]
 [1.32      ]
 [0.00663317]
 [0.198     ]
 [1.122     ]
 [0.05335842]
 [2.08097837]]

```

Kjør cella under for å sjekke formattering i svaret ditt

In [18]:



```

1 assert all (type(i)==np.ndarray
2              for i in [A,y]), 'Variablene A og y skal defineres med np.array()'
3 assert np.shape(A)==(11, 11), 'Matrisen A skal ha formen (11,11)'
4 assert np.shape(y) == (11, 1), 'Vektoren y skal ha formen (11,1)'
5 assert np.shape(x) == (11, 1), 'Vektoren x skal ha formen (11,1)'
6

```

Problem 6 – analysere resultatene (30 poeng)

a) (5 poeng) Bruk resultatene fra forrige oppgave til å bestemme den årlige materialstrømmen av gjenvunnet (resirkulert) aluminium fra *Process 5* (P5) til *Process 1* (P1) per person, det vil si med enhet kg/person/år.

Angi svaret som Python-variabelen `x51` .

Tips: Du kan benytte indeksering for å finne svaret. De ulike enkeltverdiene (elementene) i en vektor angis med en indeks der den første verdien har indeks null. Altså, det første elementet i vektoren x finnes som

In [19]:

Run

```
1 # Skriv svaret ditt her 😊
2 x51 = x[10]
3 print(x51)
```

[2.08097837]

Kjør cella under for å sjekke formattering i svaret ditt

In [20]:

▶

```
1 assert type(x51) in [int, float, np.int64, np.float64] or (type(x51)==np.ndarray
2                                     and np.shape(x51)==(1,)),
3
```

b) (10 poeng) Hvert år tapes det aluminium ut fra systemet. Hvor stor andel av det samlede tapet er på grunn av dårlig sortering hos forbrukere? Oppgi svaret med Python-variabelen `loss_share_consumers`.

Merk at vi *ikke* ber om prosentandel, men andel - som er *et tall mellom 0 og 1*.

In [21]:

▶

Run

```
1 # Skriv svaret ditt her 😊
2 consumer_loss = x[7]
3 total_loss = consumer_loss + x[9]
4 loss_share_consumers = consumer_loss/total_loss
```

Kjør cella under for å sjekke formattering i svaret ditt

In [22]:

▶

```
1 assert type(loss_share_consumers) in [int, float, np.int64, np.float64] or (type(loss_share_consumers)
2                                     and np.shape(loss_share_consumers)==(1,)), 'Svaret må være ett tall'
3
```

c) (5 poeng) Vi antar at systemet er stabilt, det vil si i *steady-state*, dermed slik at det ikke er akkumulering i systemet og at importstrømmer balanseres av eksportstrømmer.

Hva er årlig strøm av primæraluminium til *Process 1* (x_{01}), i enhet kg/person/år? Angi svaret som Python-variabelen `x01`.

In [23]:



Run

```

1 # Skriv svaret ditt her 😊
2 x01 = x[0]
3 print(x01)

```

[0.25135842]

Kjør cella under for å sjekke formattering i svaret ditt

In [24]:



```

1 assert type(x01) in [int, float, np.int64, np.float64] or (type(x01)==np.ndarray
2     and np.shape(x01)==(1,)), 'Svaret nmå være ett tall'
3

```

d) (10 poeng) Det er et betydelig tap av aluminium i produksjonsprosessene, der materialeffektiviteten i *Process 2* i dag er \$79\%\$. Vi ønsker å undersøke potensialet i bedre dimensjonering av aluminiumsplatene som leveres fra *Process 1*, der vi tenker at dette kan øke materialeffektiviteten i *Process 2* til \$90\%\$.

Hva ville vært innstrømmen av primæraluminium til *Process 1* med en slik forbedring, i enhet enhet kg/person/år? Angi svaret med Python-variabelen `x01_new`.

In [25]:



Run

```

1 # Skriv svaret ditt her 😊
2 A_new = np.array([[1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 1],
3                  [0, 1, 0, -1, -1, 0, 0, 0, 0, 0, 0],
4                  [0, 0, 0, 1, 0, -1, -1, 0, 0, 0, 0],
5                  [0, 0, 0, 0, 0, 1, 0, -1, -1, 0, 0],
6                  [0, 0, 1, 0, 1, 0, 1, 0, 1, -1, -1],
7                  [0.72, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0.72],
8                  [0, 0.9, 0, -1, 0, 0, 0, 0, 0, 0, 0],
9                  [0, 0, 0, 1/200, 0, 0, -1, 0, 0, 0, 0],
10                 [0, 0, 0, 0, 0, 0.85, 0, 0, -1, 0, 0],
11                 [0, 0, 0.025, 0, 0.025, 0, 0.025, 0, 0.025, -1, 0],
12                 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]])
13 y_new = np.array([[0], [0], [0], [0], [0], [0], [0], [0], [0], [0], [1.32]])
14 x_new = np.linalg.inv(A_new)@y_new
15 x01_new = x_new[0]
16 print(x01_new)

```

[0.24423184]

Kjør cella under for å sjekke formattering i svaret ditt

In [30]:



```
1 assert type(x01_new) in [int, float, np.int64, np.float64] or (type(x01_new)==np.nda
2     and np.shape(x01_new)==(1,)), 'Svaret må være ett tall'
3
```

Problem 7 – materialstrøm for klær i Norge (40 poeng)

Denne oppgaven er tema for flipped clasroom-timen om materialstrømsanalyse.

Tallene er hentet fra [masteroppgaven til Maria Carolina Mora Sojo fra 2021](https://hdl.handle.net/11250/2824793)
(<https://hdl.handle.net/11250/2824793>), med noen tilpasninger.

a) Vi sorterer 60% av brukte klær til gjenbruk hos Fretex og andre. Regn ut hvor mye klær som faktisk blir gjen solgt i Norge.

Kjør cella under for å generere oppgaven.

In [27]:



```
1 assignments.getAssignment('assignment_7')
```

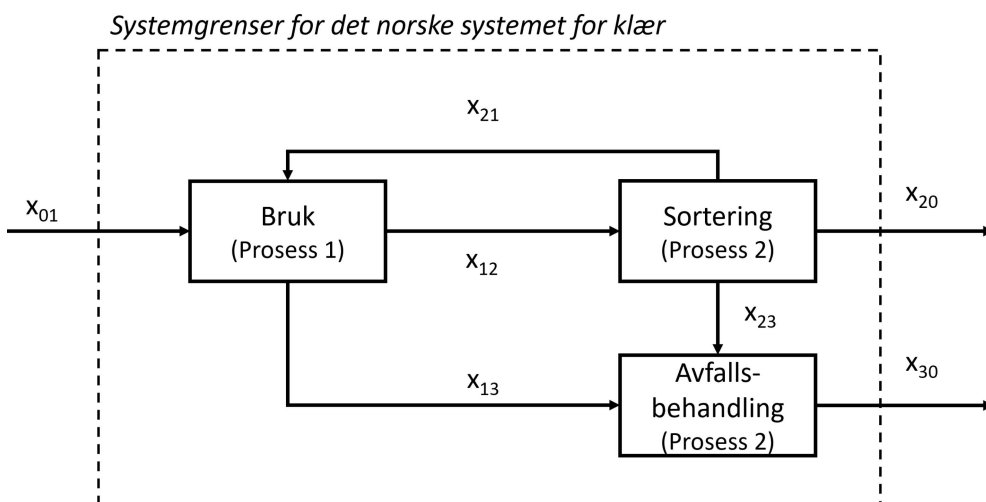
Out[27]:

Vi skal nå se på det norske systemet for klær.

I 2018 kjøpte nordmenn til sammen \$50\,000\$ tonn med klær. Hvis vi ser på hvor mye klær som ble tatt i mot hos ulike sorteringsmottakere (som Fretex), fikk disse inn \$60\%\$ av dette og resten (\$40\%\$) gikk til avfallsbehandling. Av det som Fretex og andre fikk inn dette året ble \$3\%\$ solgt tilbake til norske forbrukere og \$3\%\$ levert til avfallsbehandling i Norge. Det resterende ble sortert for gjenbruk utenfor Norge og kan her defineres som eksport.

Når vi setter opp dette systemet så definerer vi tre prosesser:

- **Bruk (1)** der norske forbrukere kjøper klær, og enten leverer dette til sortering, eller avfallsbehandling etter bruk.
- **Sortering (2)** der Fretex og andre tar inn brukte klær, og leverer dette til gjensalg i Norge, gjenbruk i utlandet (ut av systemet), eller til avfallsbehandling i Norge.
- **Avfallsbehandlingen (3)** der klær fra husholdninger og sorteringen sluttbehandles. Regn at alt som får inn til avfallsbehandling ender som en strøm ut av systemet.



Dette systemet har tre prosesser, og i alt 7 strømmer (x_{01} , x_{12} , x_{13} , x_{20} , x_{21} , x_{23} , x_{30}). Med tre prosesser kan dere sette opp tre balanseligninger (inn = ut for hver prosess). Hvis vi ser på prosessen *Bruk* har vi at $x_{01} + x_{21} = x_{12} + x_{13}$, og tilsvarende balanse kan vi lage for de andre to.

Da trenger dere i tillegg 4 modell-ligninger for å løse det. Modell-ligninger kan lages med utgangspunkt i at dere vet:

- (1) størrelsen på x_{01} (hvor mye nye klær som kjøpes årlig).
- (2) andel klær som går fra bruk til sortering; x_{12} .
- (3) andel klær fra sortering som går til gjensalg i Norge; x_{21} .
- (4) andel klær som går fra sortering til avfallsbehandling i Norge; x_{23} .

Sett opp koeffisientmatrisa og regn ut hvor mye klær som ble gjensolgt i Norge i 2018, det vil si størrelsen på strøm x_{21} . Oppgi svaret som python-variabelen `gjensalg` i enhet tonn/år.

In [38]:



Run

```

1 # Skriv svaret ditt her 😊
2 #           x01,x12,x13,x20,x21,x23,x30
3 en = 50000
4 to = 0.6
5 tre = 0.03
6 fire = 0.03
7
8 N = np.array([[1, -1, -1, 0, 1, 0, 0],
9               [0, 1, 0, -1, -1, -1, 0],
10              [0, 0, 1, 0, 0, 1, -1],
11              [1, 0, 0, 0, 0, 0, 0],
12              [-to, 1, 0, 0, -to, 0, 0],
13              [0, -tre, 0, 0, 1, 0, 0],
14              [0, -fire, 0, 0, 0, 1, 0]])
15 j = np.array([[0], [0], [0], [en], [0], [0], [0]])
16 i = np.linalg.inv(N)@j
17 gjensalg = i[4]
18 print(gjensalg)

```

[916.49694501]

Kjør cella under for å sjekke formattering i svaret ditt

In [39]:



```

1 assert type(gjensalg) in [int, float, np.int64, np.float64] or (type(gjensalg)==np.n
2     and np.shape(gjensalg)==(1,)), 'Svaret må være ett tall'
3

```

b) Bonusoppgave: vi blir mer villige til å kjøpe brukte klær

Det som sorteres og sendes til gjensalg utenfor Norge blir ikke nødvendigvis brukt til noe. Mye av dette ender faktisk som avfall i andre land, som dokumentert av [NRK blant annet \(https://www.nrk.no/urix/pa-klesindustriens-kirkegard-i-chiles-orken-1.15802311\)](https://www.nrk.no/urix/pa-klesindustriens-kirkegard-i-chiles-orken-1.15802311).

Det er to faktorer som bestemmer hvor mye av klesavfallet som ender i gjensalg i Norge; henholdsvis andel som sorteres til gjenbruk, og andel av sortert klær for gjensalg som selges i Norge. Vi er rimelig gode til å sortere klær i Norge så vi skal undersøke litt nærmere den siste faktoren og hva som skjer om vi øker denne. Tenk deg et system hvor vi kjøper like mye klær som nå men der andelen av sortert som går til gjensalg økes fra 3% til 15%. Hvor mye brukte klær er det som omsettes nå?

Du kan enkelt regne det ut ved å skalere opp strømmen x_{21} fra forrige oppgave. Forsøk likevel å lage en ny matrise som sist og regn det ut slik. Vi vil fortsette å bruke den matrisa i de neste oppgavene.

Regn ut x_{21} med denne nye faktoren.

In []:



Run

1 # Skriv svaret ditt her 😊

c) Bonusoppgave: kjøp av brukte klær erstatter kjøp av nye klær

La oss nå gjøre bruk av den nye systemdefinisjonen. Hittil har vi lagt til grunn at strømmen x_{01} er fast lik 50 000 tonn med klær. Tenk deg nå at vi i stedet har at det skal **gjennomføres 50 000 kjøp** men at disse fordeler seg mellom *nye klær* (x_{01}), og *brukte klær* (x_{21}), altså slik at $x_{01} + x_{21} = 50\,000$.

- vi har de samme strømmene som før, så alle balanseligninger er de samme som før.
- vi bruker den nye antagelsen fra (b) at 15% av utstrømmen fra prosess 2 går til gjensalg i Norge.
- vi har en ny modell-ligning for hvor mye som omsettes av klær, slik at trinn 1 i (a); $x_{01} = \$50\,000$, erstattes med $x_{01} + x_{21} = \$50\,000$.

Regn ut nå hvor mye nye klær som systemet trenger, altså størrelsen på strømmen x_{01} med denne endringen.

In []:



Run

1 # Skriv svaret ditt her 😊

d) Bonusoppgave: hva må til for å oppnå 20% reduksjon i produksjon av nye klær

Hvor stor må andelen som går fra sortering til bruk være for at produksjonen av *nye klær* skal reduseres til 40 000 tonn per år. For å finne svaret kan du gjøre følgende:

- fortsett med forutsetningen fra oppg (c) om antall salg, at $\$x_{01} + \$x_{21} = \$50\,000$;
- prøv deg så frem med ulike verdier for koeffisienten som angir $\$x_{21}$, den du endret i oppg (b).

In []:



Run

1 # Skriv svaret ditt her 😊