LCA caluculation: a simple example

source of this example

For instance, consider a unit process (or process in short), say, **production of electricity**, which:

- uses 2 litre of fuel
- produces 10 kWh of electricity
- emits 1 kg of carbon dioxide
- emits 0.1 kg of sulphur

Then a second unit process, say **production of fuel** that

- produces 100 litre of fuel
- emits 10 kg of carbon dioxide
- emits 2 kg of sulphur dioxide
- requires 50 litre of crude oil

The process matrix can be represented as

$$\mathbf{P} = \left(\frac{\mathbf{A}}{\mathbf{B}}\right) = \left(\begin{array}{cc} -2 & 100 \\ 10 & 0 \\ \hline 1 & 10 \\ 0.1 & 2 \\ 0 & -50 \end{array}\right)$$

Matrix A is the **technology matrix**; it represents the flows within the economic systems.

Matrix B is the **intervention matrix**; it represents the environmental interventions of unit processes.

Let's represent A and B with Numpy:

In [2]:
```
import numpy as np

A = np.array([[-2, 100],[10, 0]])
B = np.array([[1, 10],[0.1, 2],[0, -50]])
A, B
```

Out[2]:
```
(array([[ -2, 100],
        [ 10,   0]]),
 array([[  1. ,  10. ],
        [  0.1,   2. ],
        [  0. , -50. ]]))
```

Next, let's define a **functional unit**, in this example the production of 1000 kWh of electricity

In [3]:
```
f = np.array([0, 1000]) # reference flow
```

We now introduce the **scaling vector s**, which tells us how much of process 1 and 2 we need to produce our functional unit

The system of equations

$$\begin{cases} a_{11} \times s_1 + a_{12} \times s_2 = f_1 \\ a_{21} \times s_1 + a_{22} \times s_2 = f_2 \end{cases}$$

can be written as

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

or even more concisely as

$$\mathbf{A}\mathbf{s} = \mathbf{f}$$

We can solve this linear system using

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{f}$$

where $\mathbf{A}^{-1}$ denotes the inverse matrix of the technology matrix $\mathbf{A}$.

In Numpy, that gives

In [4]:
```
s = np.matmul(np.linalg.inv(A), f)
s
```

Out[4]:
```
array([100.,   2.])
```

In our solution, we will need:

- 100 units of process 1 (production of electricity) and
- 2 units of process 2 (production of fuel)

to produce 1 functional unit

Environmental flows

We can now use **s** to compute the environmental flows, using **g=Bs**

In [16]:
```
g = np.matmul(B, s)
g
```

Out[16]:
```
array([ 120.,   14., -100.])
```

Thus the environmental flows will be:

- 120 kg of carbon dioxide emitted
- 2 kg of sulphur dioxide emitted
- 100 litre of crude oil consumed

In [ ]:
```

```