Week 17 Exercises: SDA
(April 26th 2022)

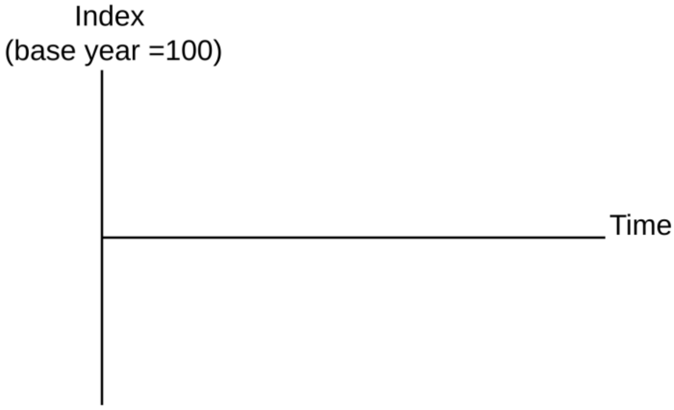
Objectives

- Interpret SDA results
- Construct SDA in Python
- Implement SDA in MRIO

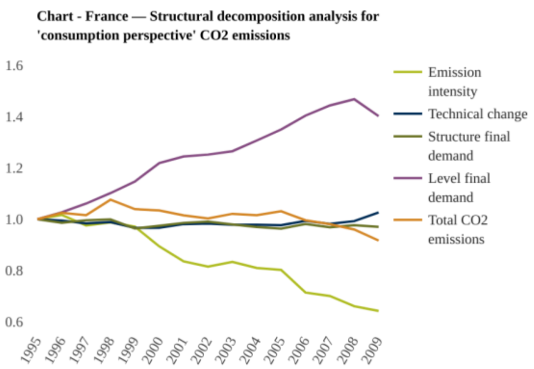
Conceptual exercises

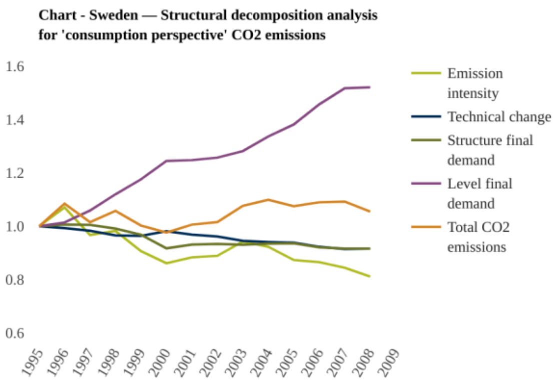
Part 1: Understanding absolute and relative decoupling

Draw a graph depicting i) relative decoupling and ii) absolute decoupling of economic activity and resource use. The y-axis of the graph can represent an index of economic activity and resource use where 100 indicates the base year. The x-axis can represent time.



Part 2: Interpreting SDA results





a) Between 2002 and 2003 what was the relative change in consumption-based CO_2 emissions in France and Sweden
b) Which factor(s) were responsible for these observed trends
c) To what extent is relative and absolute decoupling of emissions and economic growth achieved in both countries?
d) Identify three other factors not listed above that may influence the emissions footprint of economic activity and could be incorporated into SDA
e) How might rebound effects offset the emissions savings of energy efficiency gains?

Python exercises

Part 1 (no python code)

Write a structural decomposition analysis Python program that recreates the values in Table 13.1 of Miller & Blair (page 593-602). The table and the data needed are shown below.

$$\mathbf{Z}^0 = \begin{bmatrix} 10 & 20 & 25 \\ 15 & 5 & 30 \\ 30 & 40 & 5 \end{bmatrix}, \quad \mathbf{f}^0 = \begin{bmatrix} 45 \\ 30 \\ 25 \end{bmatrix}, \quad \mathbf{Z}^1 = \begin{bmatrix} 12 & 15 & 35 \\ 24 & 11 & 30 \\ 36 & 50 & 8 \end{bmatrix}, \quad \mathbf{f}^1 = \begin{bmatrix} 50 \\ 35 \\ 26 \end{bmatrix}$$

Table 13.1 Alternative Structural Decompositions

	Technology Change Contribution	Final-Demand Change Contribution	Interaction Term
Equation (13.3)	$\begin{bmatrix} 0.90 \\ 8.62 \\ 9.01 \end{bmatrix}$	$\begin{bmatrix} 11.10 \\ 11.38 \\ 10.99 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
Equation (13.4)	$\begin{bmatrix} 0.78 \\ 9.66 \\ 9.96 \end{bmatrix}$	$\begin{bmatrix} 11.22 \\ 10.34 \\ 10.04 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
Equation (13.5)	$\begin{bmatrix} 0.90 \\ 8.62 \\ 9.01 \end{bmatrix}$	$\begin{bmatrix} 11.22 \\ 10.34 \\ 10.04 \end{bmatrix}$	$+\begin{bmatrix} -0.12 \\ 1.04 \\ 0.95 \end{bmatrix}$
Equation (13.6)	$\begin{bmatrix} 0.78 \\ 9.66 \\ 9.96 \end{bmatrix}$	$\begin{bmatrix} 11.10 \\ 11.38 \\ 10.99 \end{bmatrix}$	$-\begin{bmatrix} -0.12 \\ 1.04 \\ 0.95 \end{bmatrix}$
Equation (13.7)	$\begin{bmatrix} 0.84 \\ 9.14 \\ 9.49 \end{bmatrix}$	$\begin{bmatrix} 11.16 \\ 10.86 \\ 10.51 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$$\Delta \mathrm{x} = L^1 (f^0 + \Delta f) - (L^1 - \Delta L) f^0 = (\Delta L) f^0 + L^1 (\Delta f) \tag{13.3}$$

$$\Delta \mathrm{x} = (L^0 + \Delta L) f^1 - L^0 (f^1 - \Delta f) = (\Delta L) f^1 + L^0 (\Delta f) \tag{13.4}$$

$$\Delta \mathrm{x} = (L^0 + \Delta L)(f^0 + \Delta f) - L^0 f^0 = (\Delta L) f^0 + L^0 (\Delta f) + (\Delta L)(\Delta f) \tag{13.5}$$

$$\Delta \mathrm{x} = L^1 f^1 - (L^1 - \Delta L)(f^1 - \Delta f) = (\Delta L) f^1 + L^1 (\Delta f) - (\Delta L)(\Delta f) \tag{13.6}$$

$$\Delta \mathrm{x} = (1/2)(\Delta L)(f^0 + f^1) + (1/2)(L^0 + L^1)(\Delta f) \tag{13.7}$$

Python exercises

Part 1 (with python code)

Write a structural decomposition analysis Python program that recreates the values in Table 13.1 of Miller & Blair (page 593-602). The table and the data needed are shown above.

```
#%%
# Input data
# t=0 transaction matrix and final demand vector
Z_0 = np.array([
    [10, 20, 25],
    [15, 5, 30],
    [30, 40, 5]])

f_0 = np.array([
    [45],
    [30],
    [25]])

# t=1 transaction matrix and final demand vector
Z_1 = np.array([
    [12, 15, 35],
    [24, 11, 30],
    [36, 50, 8]])

f_1 = np.array([
    [50],
    [35],
    [26]])

#%%
# Calculate total requirements matrix
x_0 = Z_0.sum(axis=1, keepdims=True) + f_0
A_0 = Z_0 @ invdiag(x_0[:, 0])
L_0 = np.linalg.inv(np.eye(3) - A_0)

x_1 = Z_1.sum(axis=1, keepdims=True) + f_1
A_1 = Z_1 @ invdiag(x_1[:, 0])
L_1 = np.linalg.inv(np.eye(3) - A_1)

# All input data for our SDA is now complete.
# Start decomposing the effect of technology change and final demand change on the
# total output. A two factor decomposition.
```

```
#%%
# change in the total requirements matrix
L_delta = L_1 - L_0
# change in the final demand
f_delta = f_1 - f_0
# change in total output - used as check
x_delta = x_1 - x_0

#%%
# decomposition (eq 13.3)
tech_change = L_delta.dot(f_0)
fd_change = L_1.dot(f_delta)

# check if our result is complete
tot_change = tech_change + fd_change
check = x_delta - tot_change
check_decomposition(check)

result = {"13.3": [tech_change, fd_change]}

print(f"\n13.3:\ntech_change: {tech_change}\nfd_change: {fd_change}")

#%%
# decomposition (eq 13.4)
tech_change = np.dot(L_delta, f_1)
fd_change = np.dot(L_0, f_delta)

# check if our resul is complete
tot_change = tech_change + fd_change
check = x_delta - tot_change
check_decomposition(check)

result["13.4"] = [tech_change, fd_change]

print(f"\n13.4:\ntech_change: {tech_change}\nfd_change: {fd_change}")

#%%
# decomposition (eq 13.5)
tech_change = np.dot(L_delta, f_0)
fd_change = np.dot(L_0, f_delta)
interaction = np.dot(L_delta, f_delta)

# check if our result is complete
tot_change = tech_change + fd_change + interaction
check = x_delta - tot_change
check_decomposition(check)

result["13.5"] = [tech_change, fd_change, interaction]

print(
```

```
    (f"\n13.5:\ntech_change: {tech_change}\nfd_change: {fd_change}"
    f"\ninteraction: {interaction}"))

#%%
# decomposition (eq 13.6)
tech_change = np.dot(L_delta, f_1)
fd_change = np.dot(L_1, f_delta)
interaction = np.dot(L_delta, f_delta)

# check if our result is complete
tot_change = tech_change + fd_change - interaction
check = x_delta - tot_change
check_decomposition(check)

result["13.6"] = [tech_change, fd_change, interaction]

print((f"\n13.6:\ntech_change: {tech_change}\nfd_change: {fd_change}"
    f"\ninteraction: {interaction}"))

#%%
# decomposition (eq 13.7)
tech_change = 0.5 * np.dot(L_delta, f_0 + f_1)
fd_change = 0.5 * np.dot(L_0 + L_1, f_delta)

# check if our result is complete
tot_change = tech_change + fd_change
check = x_delta - tot_change
check_decomposition(check)

result["13.7"] = [tech_change, fd_change]

print(f"\n13.7:\ntech_change: {tech_change}\nfd_change: {fd_change}")
```