A tool box for a climate neutral housing sector: Modeling economic effects with IO-tables

Anna Hornykewycz* Jakob Kapeller† Jan David Weber‡ Bernhard Schütz§ Lukas Cserjan¶

2024-10-26

1 Preparations

1.1 Load Packages

1.2 Importing data

In this step we import IO-Tables (`io_data_raw`), intermediate results on costs from the last section (`yearly_cost_data`) a specifically prepared file based on a detailed study on construction costs for German buildings that helps to allocate overall renovation costs (`cost_distribution`) to specific sectors (see here for more details).[1]

Although based on German data, the file could also be applied to allocate costs to sectors in countries outside Germany, if construction techniques and material employed were similar to Germany.

```r
io_data_raw <- read.csv(here::here("Data/IO_data_EN2019.csv"))
io_data_raw <- io_data_raw %>% rename(sectors=X)
sector_labels <- as.list(select(io_data_raw, sectors))

yearly_cost_data <- read.csv(
                    here::here("Intermediate_Results/results_for_io.csv"))

cost_distribution <- read.csv(here::here("Data/cost_factors_renovation_IO.csv"))
```

1.3 Adding parameters

In this section we define a few parameters to make subsequent calculations more general. First, we denote the number of interconnected sectors represented by the data (which is 72 for the *German case*) and denote

*Institute for Comprehensive Analysis of the Economy (ICAE), Johannes Kepler University, Linz, Austria

†Institute for Socio-Economics, University of Duisburg-Essen, Germany and Institute for Comprehensive Analysis of the Economy (ICAE), Johannes Kepler University, Linz, Austria

‡Institute for Socio-Economics, University of Duisburg-Essen, Germany

§Vienna Institute for International Economic Comparisions (wiiw) and Institute for Socio-Economics, University of Duisburg-Essen, Germany

¶Institute for Comprehensive Analysis of the Economy (ICAE), Johannes Kepler University, Linz, Austria

[1] For insulation material we consulted summaries of a related [business study](https://ceresana.com/produkt/marktstudie-daemmstoffe-welt) to provide as a yard stick for allocating costs related to insulation material.

the rows/columns containing final consumtion and wages, which are variables employed in the analysis to detect additional impact of investements on consumption, that is mediated by additional wages paid out.

In addition, this section offers the possibility to modify the temporal setup of the application.

```
count_sectors <- 72
location_final_consumption <- 74
location_wages <- 78

start_year <- 2024
end_year <- 2050
duration <- end_year-(start_year-1)
```

1.4 Creating manipulated versions of the IO data

Here we create four different versions of the IO-tables:

(1) A better labeled version of the original.
(2) A transposed version collecting demand in columns.
(3) A version just employing the parts needed for the IO-analysis.
(4) A version that is suitable to collect impulse vectors for all years.

```
# standard view: columns represent costs for each sector
io_cost <- select(io_data_raw, -sectors)
rownames(io_cost) <- pull(io_data_raw, sectors)

# transposed version: columns represent demand for each sector
io_demand <- data.frame(t(io_cost))

# focus on the part relevant for the mechanics of the analysis
io_cost_reduced <- slice(select(io_cost,
          c(1:count_sectors, all_of(location_final_consumption))),
          c(1:count_sectors, all_of(location_wages)))

# version for collecting impulse vectors as implied by cost estimation
io_impulse_empty <- data.frame(year=start_year:end_year,matrix(
            c(rep(0,(count_sectors+1)*duration)),
            nrow = duration))

names(io_impulse_empty) <-
  append(names(io_demand[c(1:count_sectors,all_of(location_wages))]),
         "year", after=0)
```

1.5 Calculcation factor shares and linear relationships for main variables.

In this section we map the relationsship between key macroeconomic variables and total production, which is – assuming linearity – helpful to gain estimates on how additional investments impact on macroeconomic conditions.

```
factors_and_aggregates <- io_demand %>%
  reframe(effect_employed=(Employed.persons..in.Germany.[1:count_sectors]/
            Output[1:count_sectors])/1000,
```

```
          effect_employment=(Employees..in.Germany.[1:count_sectors]/
            Output[1:count_sectors])/1000,
          share_imports=Use.of.imports[1:count_sectors]/
            Output[1:count_sectors],
          share_profit=Net.operating.surplus[1:count_sectors]/
            Output[1:count_sectors],
          share_wages=
            Compensation.of.employees..domestic.concept.[1:count_sectors]/
            Output[1:count_sectors],
          share_VA=Gross.value.added[1:count_sectors]/
            Output[1:count_sectors])
```

2 Creating impulse vectors representing additional investments

2.1 Allocating costs to sectors

In this section we project the information on the distribution of costs across different renovation activities on economic sectors as represented in the input-output tables. We build in the considerations found in `cost_distribution` and show how the information collected there maps costs unto sectors.

```
cost_distribution <- cost_distribution %>%
  mutate(share_installation = share_roof * roof_inst +
                              share_wall * wall_inst +
                              share_ceiling * ceiling_inst +
                              share_window * window_inst,
         share_material = share_roof * roof_mat +
                          share_wall * wall_mat +
                          share_ceiling * ceiling_mat,
         other_chemistry = share_material * share_chem,
         other_ceramics = share_material * share_keramik,
         other_glass = share_material * share_glass,
         other_plastics = window_mat * share_window)
```

2.2 Creating impulse vectors

In this section we use the general mapping just developed to create vectors mapping the (additional) annual investment on the specific layout the underlying IO-tables.

The table `yearly_cost_redux` can be used to specify which series to analyze.

```
yearly_cost_redux <- yearly_cost_data %>%
  rename(hp_cost = yearly_io_hp_cost) %>%
  select(., year, io_cost_prio, hp_cost) %>%
  mutate(., other_costs = io_cost_prio - hp_cost)

yearly_cost_io <- io_impulse_empty[1:(count_sectors+1)] %>%
  mutate("Specialised.construction.works"=
           yearly_cost_redux$other_costs*cost_distribution$share_installation
           + yearly_cost_redux$hp_cost*cost_distribution$hp_inst,
         "Chemicals.and.chemical.products"=
           yearly_cost_redux$other_costs*cost_distribution$other_chemistry,
```

```
          "Ceramic.products..processed.stone.and.clay"=
            yearly_cost_redux$other_costs*cost_distribution$other_ceramics,
          "Glass.and.glassware"=
            yearly_cost_redux$other_costs*cost_distribution$other_glass,
          "Rubber.and.plastics.products"=
            yearly_cost_redux$other_costs*cost_distribution$other_plastics,
          "Electrical.equipment"=
            yearly_cost_redux$hp_cost*cost_distribution$hp_electricity,
          "Machinery"=
            yearly_cost_redux$hp_cost*cost_distribution$hp_machines,
          "Wages.and.salaries"=
            cost_distribution$Wages.and.salaries
          )
```

3 Applying the IO model

3.1 Basic IO matrix operations

In this section we derive the Leontief-Inverse from the empirical IO-tables. An IO-table typically represent gross transaction values between sectors, so we divide by total sector production to detect relative contributions (which is achieved by multiplying with `diag(x_hat)`) and create the Leontief-Inverse from there.

```
x <- slice(select(io_cost,Total.uses.of.products),1:count_sectors) %>%
     add_row(slice(select(
       rename(io_demand,Total.uses.of.products=Wages.and.salaries),
       all_of(location_wages)), count_sectors+1))

rownames(x)[(count_sectors+1)] <- "Wages.and.salaries"

x_hat <- diag(pull(x,Total.uses.of.products),
              (count_sectors+1),(count_sectors+1))

A <- data.matrix(io_cost_reduced) %*% Inverse(x_hat, tol=.Machine$double.eps)

L_inverse <- Inverse(diag(1,(count_sectors+1),(count_sectors+1))-A,
                     tol=.Machine$double.eps)
```

3.2 Applying the model to all years

In this section we apply the model to all years and collect results.

```
#calculate gross output induced for all periods and reconvert to data.frame

new_output_matrix <- L_inverse %*%
                     t(as.matrix(yearly_cost_io[2:(count_sectors+2)]))

yearly_new_output <- as.data.frame(new_output_matrix)

names(yearly_new_output) <- start_year:end_year
rownames(yearly_new_output) <- names(yearly_cost_io[2:(count_sectors+2)])
```

```r
sector_impact_employed <-   slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$effect_employed
sector_impact_employment <- slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$effect_employment
sector_impact_imports <-    slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$share_imports
sector_impact_profit <-     slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$share_profit
sector_impact_wages <-      slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$share_wages
sector_impact_VA <-         slice(yearly_new_output,1:count_sectors)*
                            factors_and_aggregates$share_VA

io_results <- data.frame(year = start_year:end_year,
                         growth_employed = colSums(sector_impact_employed),
                         growth_employment = colSums(sector_impact_employment),
                         growth_employment_construction =
                           t(sector_impact_employment[
                             rownames(sector_impact_employment) %in%
                            c("Specialised.construction.works"), ]),
                         growth_imports = colSums(sector_impact_imports),
                         growth_profit = colSums(sector_impact_profit),
                         growth_wages = colSums(sector_impact_wages),
                         growth_VA = colSums(sector_impact_VA),
                         initial_stimulus = rowSums(yearly_cost_io)) %>%
                         rename(growth_employment_construction=
                                  "Specialised.construction.works")

io_results <- io_results %>%
  mutate(multiplier = growth_VA / initial_stimulus) %>%
  relocate(multiplier, .after=1)

io_results_mean <- data.frame(mean=colSums(io_results[-1])/nrow(io_results))

io_results_mean
```

```
##                                        mean
## multiplier                     1.155087e+00
## growth_employed                1.026695e+06
## growth_employment              8.957348e+05
## growth_employment_construction 2.737235e+05
## growth_imports                 1.748373e+10
## growth_profit                  1.979397e+10
## growth_wages                   3.694692e+10
## growth_VA                      6.800368e+10
## initial_stimulus               5.889343e+10
```

4 Exporting final outcomes

Here we save data on mean outcomes and annual outcomes. We also export annual data on sector-specific employment effects.

```r
write.csv(io_results_mean, here::here("Results/io_results_mean.csv"))
write.csv(io_results, here::here("Results/io_results_annually.csv"))
write.csv(sector_impact_employment,
          here::here("Results/io_results_labor_sectors.csv"))
```