CGE Model Generator

Rasmus K. Skjødt Berg*

November 16, 2022

The note outlines the general framework used in the Computable General Equilibrium model generator `CGE_SmallOpen` that builds on the class `GamsPython`.[1] The `GamsPython` class is simply a collection of methods that are useful when setting up and combining models in GAMS and can thus be used more broadly. The `CGE_SmallOpen` class is a shell for a multisector, deterministic, dynamic general equilibrium model.

The class relies on balanced national accounts data (also referred to as IO data) that is split into the general modules: Production sectors, foreign production sectors, households, public sector, investment sector, inventory investment sector. Each of these six blocks are treated separately in the following.

Notation

In the `CGE_SmallOpen` class, there are some fundamental notation that cannot be altered.

Fundamental sets:

- `n`: The set of goods. The set contains final goods that are traded/consumed, but also auxiliary elements such as composite goods in a nested production function.
- `s`: The set of sectors. To keep things general, there is not a single set element that represents e.g. the households/consumers. Instead, (`'hi'`)∈ `s` can be a specific household sector that belongs to a subset of household sectors `s_hh(s)`.
- `t`: The time index.

Sets are generally aliased by repeating the symbol. For instance, `n` is aliased with `nn` and `nnn`. Also, note that some subsets are applied globally, including the time indices `t0[t]`, `tE[t]`, `txE[t]` that indicates the initial period, the terminal period, and all years except the terminal one. In general, we reserve the terminal period `tE[t]` to enforcing transversality-like conditions on dynamic problems. Thus, most equations and variables will only be defined for `txE[t]` – that is, for all years but the terminal one.

*University of Copenhagen. Oster Farimagsgade 5, 1353 Copenhagen K, Denmark. E-mail: rasmus.kehlet.berg@econ.ku.dk.

[1]The current version is GPM_v071

Fundamental variables:

Most variables are defined both as quantities, prices, and values. For variable `X`, we use the notation `qX, pX, vX` to indicate this. Dummies are generally – albeit not consistently (sorry) – denoted `d_X`. Beyond this there are a number of variables and exogenous parameters (specific to the `CGE_SmallOpen` class) that is used:

- `xD[t,s,n]`: The demand for goods n, sector s, year t (for `x=q,p,v`).
- `xS[t,s,n]`: The supply of goods n, sector s, year t (for `x=q,p,v`).
- `p[t,n]`: The equilibrium price on good n, year t.[2]
- `mu[s, n, nn]`: Share parameter used in nesting of goods `n,nn` in sector `s`.
- `sigma[s,n]`, `eta[s,n]`: Elasticities of substitution, transformation.
- `Rrate[t]`: Real interest rate factor (default 1.03).
- `R_LR`: The long run interest rate (default 1.03).
- `g_LR`: The long run growth rate (default 0.02).
- `infl_LR`: The long run inflation rate (default 0).

General equilibrium requirement:

Ultimately, the equilibrium requirement is written for all final goods `n_equi[n]` (defined from the IO data) as the condition:

$$\sum_{\texttt{d\_qS[s,n]}} \texttt{qS[t,s,n]} = \sum_{\texttt{d\_qD[s,n]}} \texttt{qD[t,s,n]},$$

where `d_qS`, `d_qD` are dummies defining what sectors participate in the market for goods `n`.

A small example

To illustrate how things work, we will generally fall back to this simple two-production sector example:

- Two domestic production sectors `(a,b)` that produce goods `(a,b)`.
- A foreign sector `F` that produces similar goods `(a_F, b_F)`.
- A household sector `h` that supplies labor `L`, consumes final goods `a,b,a_F,b_F`, and saves `s` (element in `n`).
- Two investment sectors (`I_iM,I_iB`) produce two investment goods (`I_iM,I_iB`).

[2]The differences between the equilibrium price and other, sector specific prices (`pD,pS`) are (i) mark ups and (ii) taxes.

- An inventory sector `itory` consumes the four final goods (`a,b,a_F,b_F`).
- A public sector balances the budget in each period by adjusting labor income tax level (uniform adjustment on all demand side.)

1 Modular Structure

As briefly mentioned above, the `CGE_SmallOpen` model consists of seven building blocks: (1) Domestic production sectors, (2) foreign sector, (3) household sectors, (4) investment sectors, (5) inventory sectors, (6) a public sector, and (7) a general equilibrium module.

The seven modules can be implemented in various ways – and we do not have to use the same general structure for *all* production sectors for instance. Nonetheless, there are some general rules that the modules should adhere to that makes *combining* the modules in a general equilibrium easy.

Production Module

The production module generally takes as inputs: Equilibrium prices, regulation, and supply levels. There is no restriction on how regulation is imposed: In general, we distinguish between three types of prices (`p`,`pS`,`pD`) to allow for regulation to be specific to the sector that produces the good (regulation between `pS` and `p`) and specific to the sector that demands the good (regulation between `p`, `pD`).

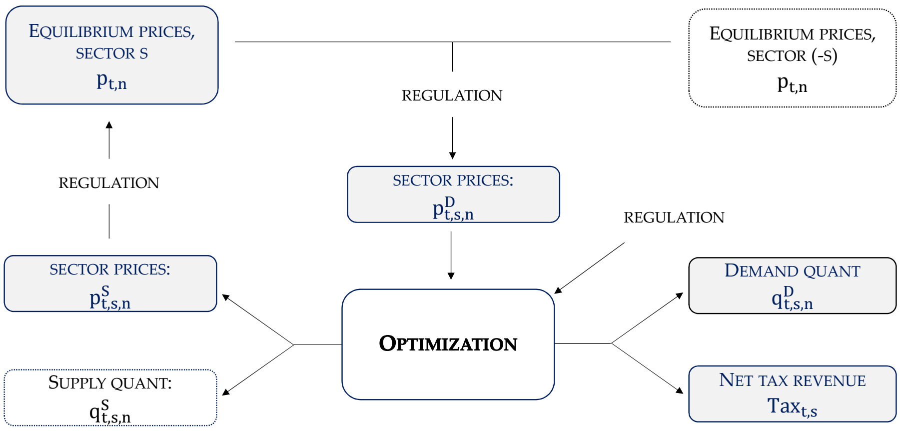
As the figure illustrates, we do not impose any specific structure on the type of optimization. We only require that it produces at least the following: Demand levels (`qD`), equilibrium prices on its outputs (`p`), and the net total tax revenue (`Tax`). The level of supply (`qS`) is also an integral part of the optimization problem, but is generally assumed to be exogenous in partial equilibrium. The simple reason here is that the *scale of output cannot be identified with CRS technology*: Rather, the CRS assumption implies that there is linear cost index (`pS`) independent of scale of output (`qS`).

The investment module works in a similar way.

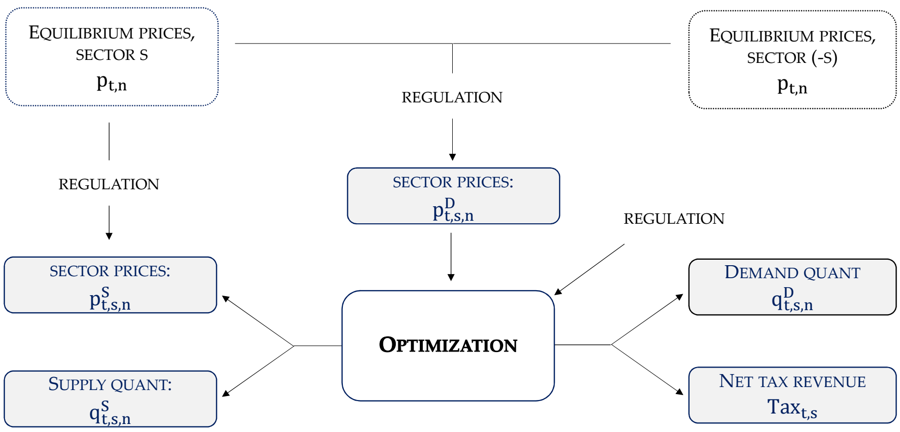
The Household Module

The household module is somewhat different – as we will elaborate on later. Generally, the module only supplies labor – and we assume that it does so in a *decreasing returns to scale* manner. This means that – compared to the CRS production module – the causality in partial equilibrium can be altered: Instead of taking the levels of supply as given (`qS`), we take equilibrium prices as given.

Modular Structure: CRS supply



Modular Structure: DRS supply



Trade module

In terms of the input-output data that the CGE model is build to replicate, the trade module only has to correctly identify the *foreign demand for domestic goods*. That is, while other sectors has an output to match (e.g. production or supply of labor), the trade module only needs to identify exports (imports are covered by the respective sectors).

The reason for this is that we adopt the small open economy assumption, implying that there is an infinite supply of foreign goods available at some fixed price. Note that this is not a general restriction on the trade module, but rather the reason why we do not require that the module identifies the same variables as production or household modules.

The government module

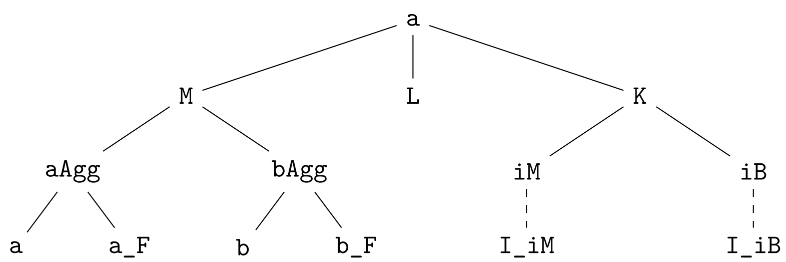
We currently have almost no restrictions to the government module. To make it easier to interact with other modules, we require that other modules always sum up their net tax revenue in the variable `TotalTax[t,s]` as shown in the previous modules.

2 The `CGE_Production` module

The module covers production sectors by specifying nested production trees. The class `nestingTree` is used to set up the nesting structure by specifying: (i) The links in the tree, (ii) the type of nesting (input/output type), (iii) the functional form applied in the nesting, and (iv) parameter values in the nest (shares, elasticities). The notebook `nestingTree.ipynb` provides some more detail on this. The following outlines the `CGE_Production` module using our small example. However, the module allows us to use arbitrarily nested production technologies – including *multiple-output* nests – and a number of functional forms (CES, CET, MNL, generalized scale preserving demand systems). A few restrictions are, however: (i) All nests feature constant returns to scale technologies, and (ii) no goods are produced by more than one sector.

In the simple example, the two sectors `(a,b)` produce a single good each `(a,b)` by combining final goods, labor, and durables. Figure 2.1 illustrates the production nest for sector `a`; a symmetric version is applied for `b`.

Figure 2.1: Production nest



The figure illustrates that – ultimately – the production depends on final goods: `a`, `b`, `a_F`, `b_F`, `L`, `I_iM`, `I_iB`. The "conventional" goods (`a`, `b`, `a_F`, `b_F`) are combined in a materials nest (`M`) and the investment goods (`I_iM`, `I_iB`) are combined in a durables nest (`K`). All (solid line) nests are simply modelled as constant elasticity of substitution (CES) with the syntax. The dashed lines indicate that the durables (`iB`, `iM`) are accumulated by accruing corresponding investment goods (`I_iM`, `I_iB`).

2.1 Equation blocks

The module produces three blocks of equations: Non-durables, durables, and profits.

Block I: Non-durables

The first one is a simple system of equations for each nest of non-durables. For a simple CES nest, this includes the following block of equations:[3]

$$\text{For all knots:} \qquad p_{(t,s,n)} \cdot q_{(t,s,n)} = \sum_{n'} q_{(t,s,n')} \cdot p_{(t,s,n')} \tag{1a}$$

$$\text{For all branches:} \qquad q_{(t,s,n)} = \mu_{(s,n',n)} \left(\frac{p_{(t,s,n')}}{p_{(t,s,n)}} \right)^{\sigma_{(s,n')}} q_{(t,s,n')}, \tag{1b}$$

where (s, n', n) indicates a nest with branch n and knot n', μ is a share parameter, σ is an elasticity of substitution. We note that (i) the code distinguishes between supply/demand prices (`pS`,`pD`), (ii) this assumes a CES-like nesting structure, but it is straightforward to include other nestst (e.g. MNL-like) and (iii) the code can also accommodate a multiple-output output-like structure.

Block II: Durables

The second block is used to handle durables. For now, let K denote a single durable good and I a corresponding investment good. The block of equations is derived from the optimization problem:

$$\begin{aligned} &\max_{t=t_0} R_{t,t_0}^{t_0-t} \left\{ p_t^k K_t - p_t^I \left[I_t + K_t \Psi \left(\frac{I_t}{K_t}, \frac{I_{t-1}}{K_t} \right) \right] \right\}, \\ &\text{s.t. } K_{t+1} = K_t(1-\delta) + I_t, \qquad K_{t_0} \text{ given.} \end{aligned}$$

where R_{t,t_0} denotes the accumulated interest rate factor between t, t_0, Ψ is the installation cost function, p_t^I is the price on investment goods (produced by another sector in the economy), and p_t^k is the value of capital. The first order conditions are then given by (for $t \geq t_0$):

$$\begin{aligned} \lambda_t R_{t+1} &= p_{t+1}^I \frac{\partial \Psi_{t+1}}{\partial (I_t / K_{t+1})} + R_{t+1} p_t^I \left(1 + \frac{\partial \Psi_t}{\partial (I_t / K_t)} \right) \\ \lambda_t R_{t+1} &= p_{t+1}^k + (1-\delta)\lambda_{t+1} - p_{t+1}^I \Psi_{t+1} \left(1 - \epsilon_{t+1}^{t+1} - \epsilon_t^{t+1} \right) \\ K_{t+1} &= (1-\delta) K_t + I_t, \end{aligned}$$

where λ_t is the continuation value of capital and ϵ_i^j is the elasticity of Ψ_j wrt. I_i / K_j. We can combine the conditions on the shadow variables into (for $t > t_0$):

$$p_t^k = p_t^I \psi_{t-1}^t + R_t p_{t-1}^I \left(1 + \psi_{t-1}^{t-1} \right) + p_t^I \Psi_t \left(1 - \epsilon_t^t - \epsilon_{t-1}^t \right) - (1-\delta) \left(p_t^I \left(1 + \psi_t^t \right) + \frac{p_{t+1}^I \psi_t^{t+1}}{R_{t+1}} \right),$$

where $\psi_i^j \equiv \partial \Psi_j / \partial (I_i / K_j)$. If we drop the I_{t-1}/K_t element from the installation costs, this simplifies to (for $t > t_0$):

$$p_t^k = R_t p_{t-1}^I \left(1 + \psi_{t-1}^{t-1} \right) + p_t^I \Psi_t \left(1 - \epsilon_t^t \right) - (1-\delta) p_t^I \left(1 + \psi_t^t \right).$$

Appendix A elaborates on final equations used in this block. We generally use a type of installation costs that adhere to assumption 1:

[3] The equations vary slightly, depending on what type of nest we are using – in particular whether or not the tree is an *input* or an *output*-like tree.

Assumption 1 (Regularity assumption, installation costs)**.**
Installation costs Ψ_t are increasing (i) in I_t/K_t, (ii) decreasing in I_{t-1}/K_t, (iii) convex, and (iv) in a long run equilibrium with $I^* = \delta K^*$:

$$\Psi^* = \psi^*_{-1} = \psi^* = 0.$$

In practice, the model is solved with a finite state T for which the assumption is that prices are in a steady state afterwards.[4] The following equations are used:

- Law of motion and TVC condition. We use a mapping `dur2inv[s,n,n']` to indicate the mapping from durables ($n = K$) to the corresponding investment good ($n' = I$):

$$\text{For } t_0 \leq t \leq T: \qquad q_{(t+1,s,n)} = q_{(t,s,n)}(1-\delta_{(t,s,n)}) + q_{(t,s,n')} \tag{2a}$$

$$\text{For } t = T: \qquad q_{(t+1,s,n)} = q_{(t,s,n)} \tag{2b}$$

- Shadow value of capital, interior $t_0 < t < T$. Assuming quadratic installation costs (again, $n = K$ and $n' = I$):

$$\begin{aligned} p_{(t,s,n)} =& R_t p_{(t-1,s,n')} \left[1 + \phi_{(s,n)} \left(\frac{q_{(t-1,s,n')}}{q_{(t-1,s,n)}} - \delta_{(t-1,s,n)}\right)\right] \\ &+ p_{(t,s,n')} \left[\frac{\phi_{(s,n)}}{2}\left(\delta^2_{(t,s,n)} - \left(\frac{q_{(t,s,n')}}{q_{(t,s,n)}}\right)^2\right) - (1-\delta_{(t,s,n)})\left(1 + \phi_{(s,n)}\left[\frac{q_{(t,s,n')}}{q_{(t,s,n)}} - \delta_{(t,s,n)}\right]\right)\right] \end{aligned} \tag{2c}$$

- Shadow value of capital, terminal $t = T$:[5]

$$p_{(t,s,n)} = R_t p_{(t-1,s,n')}\left(1 + \phi_{(s,n)}\frac{q_{(t-1,s,n')}}{q_{(t-1,s,n)}}\right) - (1-\delta_{(t,s,n)})p_{(t,s,n')}. \tag{2d}$$

- Installation costs from the sector:

$$\Psi_{(t,s)} = \sum_{(n,n')\in \texttt{dur2inv}_s} \frac{\phi_{(s,n)} \cdot p_{(t,s,n')} \cdot q_{(t,s,n)}}{2}\left(\frac{q_{(t,s,n')}}{q_{(t,s,n)}} - \delta_{(t,s,n)}\right)^2. \tag{2e}$$

Block III: Profits, prices, and tax revenues

The third block of equations accounts for final profits, taxation, and installation costs. This means that regulation (taxes) is added to each module separately. We do this to allow for a greater flexibility; regulation and thus the mapping between equilibrium prices and effective prices may be fundamentally different across sector. The downside to this approach is that adjusting some general piece of regulation, e.g. a tax on emissions, that affect all modules has to be implemented

[4] There are, naturally, alternatives to this type of TVC condition for the production module. An important lesson is, however, to ensure that the value of the terminal state is straightforward to compute, for the model to be numerically stable.

[5] The approximation used here is that the model is in steady state in T; using this, we remove all installation cost components for the terminal period.

in all modules. This is straightforward to do if we are simply considering changing the level of an existing tax rate, but takes a more work if it means introducing new variables/equations.

While taxes and profits often need to be modified to the specific model (e.g. including emissions taxation for a model on the green transition), the following describes the baseline equations used:

- Equilibrium prices (p_t^n) as a function of cost-index ($p_{(t,s,n)}$), sector-specific markup (m_s), regulation ($\tau^S_{(t,s,n)}, \tau^L_{(t,s)}$), and installation costs ($\Psi_{(t,s)}$):

$$p_t^n = (1+m_s)\left(p_{(t,s,n)} + \tau^S_{(t,s,n)} + \frac{\Gamma_{(t,s,n)}}{q_{(t,s,n)}}\left(\tau^L_{(t,s)} + \Psi_{(t,s)}\right)\right) \tag{3a}$$

$$\Gamma_{(t,s,n)} \equiv \frac{p_{(t,s,n)} \cdot q_{(t,s,n)}}{\sum_{n' \in \texttt{out}_s} p_{(t,s,n')} \cdot q_{(t,s,n')}}. \tag{3b}$$

  In this example, we use a constant, sector-specific mark up (m_s) and assume two types of regulation (on the supply side): The rate $\tau^S_{(t,s,n)}$ is a unit tax on all outputs of type n from sector s and $\tau^L_{(t,s)}$ is a lump-sum tax. The variable $\Gamma_{(t,s,n)}$ is used when a sector produces more than one type of outputs: In this case, it measures the share of value of the specific output n.

- A mapping from equilibrium prices (p_t^n) to effective input prices in sector s ($p_{(t,s,n)}$) that takes regulation into account. For instance, assuming that the input is associated with $\nu_{(t,s,n)}$ emissions and a CO_2 tax of $\tau^{CO_2}_{(t,s,n)}$ implies the mapping:

$$p_{(t,s,n)} = p_t^n + \nu_{(t,s,n)} \cdot \tau^{CO_2}_{(t,s,n)}. \tag{3c}$$

- Finally, we include a variable that sums up the net transfers to the public sector $T_{(t,s)}$ (`TotalTax` in the code). With the regulation outlined in this example, the total transfer is defined as:

$$T_{(t,s)} = \tau^L_{(t,s)} + \sum_{n \in \texttt{out}_s} q_{(t,s,n)} \cdot \tau^S_{(t,s,n)} + \sum_{n \in \texttt{in}_s} q_{(t,s,n)} \cdot \nu_{(t,s,n)} \cdot \tau^{CO_2}_{(t,s,n)}, \tag{3d}$$

  where $\texttt{out}_s$ and $\texttt{in}_s$ indicates subset of outputs/inputs from the sector (`output[s,n]` and `input[s,n]` in the code).

2.2 Causality in partial equilibrium: Baseline

The module can be solved separately from the general equilibrium. We note that when nesting trees are scale preserving – in the sense that the sum of inputs = sum of outputs – the following setup is slightly adjusted because of implicit restrictions imposed by the nature of the nesting structure.

Exogenous inputs:

In the *baseline* mode, the following variables are exogenous inputs:

- Technical parameters: Share parameters (μ), elasticities (σ, η), investment cost parameter (ϕ), depreciation rates on capital (δ), and mark-ups.
- Taxes/regulation: In the example above, this would be $\tau^{L}_{(t,s)}, \tau^{S}_{(t,s,n)}, \tau^{CO_2}_{(t,s,n)}$.
- Equilibrium prices for goods *not* produced by the sector: p^n_t for $n \in (\texttt{in}_s \backslash \texttt{out}_s)$
- Quantity of supply for all outputs: $q_{(t,s,n)}$ for $n \in \texttt{out}_s$.
- Initial stock of durables: $q_{(t_o,s,n)}$ for all $n \in \texttt{dur}_s$.

Endogenous outputs:

In the *baseline* mode, the following variables are solved for:

- Auxiliary variables: (i) Intermediate goods and price indices ($p_{(t,s,n)}$ and $q_{(t,s,n)}$ for all $n \in \texttt{int}_s$), (ii) installation costs ($\varPsi_{(t,s)}$), (iii) output shares ($\varGamma_{(t,s,n)}$ for $n \in \texttt{out}_s$).
- Demand for inputs and durables: $q_{(t,s,n)}$ for all $n \in \texttt{in}_s \cup \texttt{dur}_s$ (except t_o for durables).
- Cost index for outputs and durables: $p_{(t,s,n)}$ for $n \in (\texttt{out}_s \cup \texttt{dur}_s)$.
- Total net tax revenue: $T_{(t,s)}$.
- Effective input prices: $p_{(t,s,n)}$ for $n \in \texttt{in}_s$.
- Equilibrium prices for outputs: p^n_t for $n \in \texttt{out}_s$.

2.3 Causality in partial equilibrium: Calibration, version 1

When calibrating the model, we change what variables are exogenous/endogenous. The following variables are adjusted:[6]

- In a baseline year (t_o), we fix the quantities ($q_{(t_o,s,n)}$) for all intermediate goods and inputs ($n \in (\texttt{in}_s \cup \texttt{int}_s)$). To ensure that the system is still solved, the relevant share parameters ($\mu_{(s,n,n')}$) are endogenized.
- The equilibrium price ($p^n_{t_o}$) on all outputs from the sector in a baseline year (t_o) is fixed. In turn, the sector-specific markup is endogenized.
- The total net tax transfer ($T_{(t_o,s)}$) is made exogenous in a baseline year; in turn, the corresponding lump sum tax is endogenized.
- Fix the level of investments in the baseline year ($q_{(t_o,s,n)}$ for $n \in \texttt{inv}_s$); recall that the initial stock of the durables is always fixed. Endogenizing the share parameter on the *durable* identifies the investment level.

[6] Appendix B elaborates a bit on how/why this works.

2.4 Causality in partial equilibrium: Calibration, version 2 (exo_mu)

This version is similar to version 1, with the exception that *intermediate* goods quantities ($q_{(t,s,n)}$ for $n \in \texttt{int}_s$) are endogenous and the corresponding share parameters are fixed.

If a sector produces $i > 1$ outputs, we endogenize $i - 1$ of the share parameters in the relevant multiple output nest to make sure that the right price levels can be reached.

2.5 The Standard Approach

To end with, let's briefly go through the steps of setting up and calibrating the production module. Note: This is a suggested approach; the right one ultimately depends very much on the specific implementation.

The standard approach is as follows:

i. Collect a balanced IO set of data in values, prices, and quantities.[7] Add data on durables including the initial level of stocks and the depreciation rates.

ii. Define the global settings up front (used in all modules): This entails specifying long run parameters (growth rate, inflation, real interest rate) and e.g. the time dimension of the model, including which year is the baseline one.

iii. Initialize a production module by specifying a nesting tree. For all standard nesting trees, we need to identify parameters:

- Elasticities of substitution/transformation: This class does not include any automated way of estimating/calibrating these – so, they should be provided manually/added to the input data.
- Share parameters on inputs: Share parameters on inputs are identified in the calibration stage by targeting the IO data.[8]
- Share parameters in intermediate nests: They can be identified in one of two ways:
  - (i) If we include data on intermediate goods in the IO data, the share parameters are automatically identified in the calibration stage.[9]
  - (ii) If we do not include data on intermediate goods, the share parameters are exogenous in the calibration stage. So, identification has to be made prior to the calibration.[10] Appendix C outlines how this can be carried out using a static calibration step.

2.6 Relevant data

To calibrate the model, we need (i) a balanced IO data set (in values), (ii) specification of what goods are durables, including estimated initial stocks (in quantities) and depreciation rates, and

[7] If we only have values, the natural approach is to assume that all prices are 1 in the baseline year.

[8] If a sector has a multiple-output structure, the relevant share parameters in the relevant output nest is identified in the calibration stage as well.

[9] This is done using the class `CGE_Production.Production`.

[10] This is done using the class `CGE_Production.Production_ExoMu`.

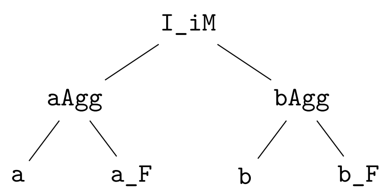
(iii) global settings.

3 The Investment module

The module covers the same production sectors as `CGE_Production`, but tracks the production of investment goods instead. The idea behind this sector is to ensure that the input output investment flow can be matched by the model. The module is essentially a simplified production module where adjustment costs and durables are dropped.

In the simple example, the two sectors (`I_iM`, `I_iB`) produce one good each (`I_iM`, `I_iB`) denoting the machine and building investment goods, respectively. Compared to the production sectors, an investment sector only draws on final goods (and not labor nor durables) in its production. Figure 3.1 illustrates the production nest for sector `I_iM`; a symmetric version is applied for `I_iB`:

Figure 3.1: Investment production nest



4 The `GmsInventory` module

The module is a small add-on used to account for inventory "investments". The only equation that is included, for now at least, is the AR(1) process without errors:

$$\texttt{qD[t,s,n]} = \texttt{inventoryAR[s,n]} \cdot \texttt{qD[t-1,s,n]},$$

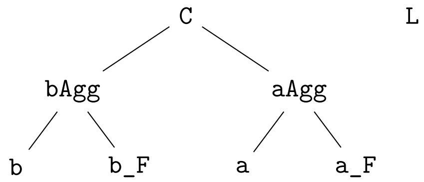
The default is behavior is to set `inventoryAR` to 1, thus keeping inventory investments constant.

5 The `CGE_Households` Module

The household module can be specified in a number of different ways. For it to be consistent with IO data and other modules, it has to specify the demand for final goods and account for assets/savings. Whether or not this is done by matching theory with frictions, exogenous labor supply, splitting it into a number of sub-sectors (e.g. representing age or income profiles) or something completely different is not essential.

In the following, we present the `SimpleRamsey` class: This is a simple Ramsey-like consumer with CRRA preferences. In our example, a single household sector combines consumption goods – intratemporally – in a nested CES-like manner as indicated in figure 5.1:

Figure 5.1: Household intratemporal nest



In the `SimpleRamsey` module, we assume that the household solves the optimization problem:

$$
\begin{aligned}
\max\ U = & \sum_{t=t_0}^{T} \beta_s^{t-t_0} \left[\frac{q_{(t,s,C)}^{1-\theta_s}}{1-\theta_s} - \frac{\xi_s}{1+\xi_s} \left(\frac{q_{(t,s,L)}}{\gamma_s} \right)^{\frac{1+\xi_s}{\xi_s}} \right] \\
\text{s.t.}\ \ v_{(t+1,s)} = & i_t \cdot v_{(t,s)} + p_{(t,s,L)} \cdot q_{(t,s,L)} - \sum_{n \in \texttt{in}_s} p_{(t,s,n)} \cdot q_{(t,s,n)} \\
v_{(T,s)} = & v_{(T-1,s)},
\end{aligned}
$$

where β_s is the time discount factor, θ_s is the CRRA coefficient of the composite consumption good $q_{(t,s,C)}$, ξ_s is the Frisch elasticity of labor supply $q_{(t,s,L)}$, γ_s is a scale parameter, $v_{(t,s)}$ is household savings, and i_t is the nominal interest rate.

5.1 Equation blocks

The module produces three blocks of equations: Intratemporal optimization, intertemporal optimization, and budgets, regulation, and prices.

Block I: Intratemporal optimization

The first block is simply a nesting tree as used in the production module. For a simple CES-like nesting, this includes the following block of equations:

$$
\text{For all knots:} \qquad p_{(t,s,n)} \cdot q_{(t,s,n)} = \sum_{n'} q_{(t,s,n')} \cdot p_{(t,s,n')} \tag{4a}
$$

$$
\text{For all branches:} \qquad q_{(t,s,n)} = \mu_{(s,n',n)} \left(\frac{p_{(t,s,n')}}{p_{(t,s,n)}} \right)^{\sigma_{(s,n')}} q_{(t,s,n')}. \tag{4b}
$$

In addition, the condition for optimal supply of labor is given by:[11]

$$
q_{(t,s,L)} = \gamma_s \left(\frac{p_{(t,s,L)}}{p_{(t,s,C)} \cdot q_{(t,s,C)}^{\theta_s}} \right)^{\xi_s}. \tag{4c}
$$

Block II: Intertemporal optimization

The second block concerns the dynamic optimization. Given the simple CRRA assumptions, this contains a law of motion for savings, an Euler equation for optimal trajectory for consumption,

[11] In the code, the L and C components are identified using the `labor[s,n]` and `output[s,n]` subsets; the two are related through the mapping `L2C[s,n,nn]`.

and a transversality condition for the assets:

$$q_{(t,s,C)} = q_{(t-1,s,C)} \left(\beta_s i_t \frac{p_{(t-1,s,C)}}{p_{(t,s,C)}} \right)^{\frac{1}{\theta_s}} \tag{5a}$$

$$v_{(t+1,s)} = i_t \cdot v_{(t,s)} + p_{(t,s,L)} \cdot q_{(t,s,L)} - \sum_{n \in \texttt{in}_s} p_{(t,s,n)} \cdot q_{(t,s,n)} - \tau^L_{(t,s)} \tag{5b}$$

$$v_{(T,s)} = v_{(T-1,s)}, \tag{5c}$$

Block III: Regulation and prices

The final block is similar to the one applied in the production module: It accounts for regulation by mapping equilibrium prices to sector-specific ones and computes the total net transfer of taxes from the households. In the current example, we use the equations:

$$\text{For } n = L: \qquad p_{(t,s,n)} = p^n_t - \tau^S_{(t,s,n)} \tag{6a}$$

$$\text{For } n \in \texttt{in}_s: \qquad p_{(t,s,n)} = p^n_t + \tau^D_{(t,s,n)} \tag{6b}$$

$$T_{(t,s)} = \tau^L_{(t,s)} + \tau^S_{(t,s,L)} \cdot q_{(t,s,L)} + \sum_{n \in \texttt{in}_s} \tau^D_{(t,s,n)} \cdot q_{(t,s,n)} \tag{6c}$$

5.2 Causality in partial equilibrium: Baseline

The module can be solved separately from the general equilibrium.

Exogenous inputs:

In the *baseline* mode, the following variables are exogenous inputs:

- Preference parameters: $\sigma, \mu, \theta, \xi, \gamma, \beta$.
- Regulation: $\tau^S_{(t,s,L)}, \tau^D_{(t,s,n)}, \tau^L_{(t,s)}$.
- Stock of savings in initial period $v_{(t_0,s)}$ and interest rates i_t.
- Equilibrium price on labor p^L_t.

Endogenous inputs:

In the *baseline* mode, the following variables are identified:

- Intermediate goods and prices: $p_{(t,s,n)}$, $q_{(t,s,n)}$ for $n \in \texttt{int}_s$.
- Demand for – and effective prices on – inputs: $q_{(t,s,n)}, p_{(t,s,n)}$ for $n \in \texttt{in}_s$.
- Supply of – and effective price on – labor supply: $q_{(t,s,L)}$, $p_{(t,s,L)}$.
- Savings for all years after the baseline $v_{(t,s)}$.
- Total net tax revenue from the sector: $T_{(t,s)}$.

5.3 Causality in partial equilibrium: Calibration

When calibrating the model, we change the status of exogenous/endogenous variables:

- The demand for all inputs in the baseline year is made exogenous ($q_{(t_0,s,n)}$ for $n \in \texttt{in}_s$); corresponding share parameters in nesting preferences are made endogenous (μ).
- The supply of labor is made exogenous ($q_{(t,s,L)}$); the scale parameter γ_s is endogenized.
- To avoid linear dependence in the system, we fix the price on the upper-most level in the consumption nest in the baseline period and unfix the discount rate β_s.
- The total net tax transfer ($T_{(t_0,s)}$) is made exogenous in the baseline year; in turn, the lump sum tax rate is endogenized.

6 The `CGE_Trade` Module

The trade module determines the foreign demand for domestically produced goods. At the moment, we model financial markets in a very simple manner – capital is perfectly mobile with a fixed interest rate of `Rrate`. The implication is that we do not need to specify the demand for domestic assets; the asset market is simply represented by the interest rate.

As an example, the following outlines `SimpleArmington` module. This uses a simple mapping $\texttt{D2F}_{n,n'}$ from domestic goods (n) to the equivalent foreign type of this good (n'):

$$\text{For all } (n,n') \in \texttt{D2F}_{n,n'}: \qquad q_{(t,s,n)} = \gamma_s \left(\frac{p_t^{n'}}{p_{(t,s,n)}}\right)^{\sigma_{(s,n)}} \tag{7a}$$

$$\text{For all } n \in \texttt{D}_s: \qquad p_{(t,s,n)} = p_t^n + \tau^D_{(t,s,n)} \tag{7b}$$

$$T_{(t,s)} = \tau^L_{(t,s)} + \sum_{n \in \texttt{D}_s} \tau^D_{(t,s,n)} \cdot q_{(t,s,n)}. \tag{7c}$$

Here $q_{(t,s,n)}$ indicates the demand for domestically produced goods (n) from foreign sectors (s). The second condition states that exports are taxed by $\tau^D_{(t,s,n)}$ with $\texttt{D}_s$ denoting the subset of domestic goods n demanded by the foreign sector s, and the final equation reports the total tax from foreign sectors.

In the baseline mode, the tax revenue $T_{(t,s)}$ and the price on/demand for domestic goods ($q_{(t,s,n)}$ for $n \in \texttt{D}_s$) are endogenous. When calibrating the model, the demand levels and tax transfers are exogenized; instead, the scale parameters γ_s and lump sum taxes in the baseline year ($\tau^L_{(t_0,s)}$) are endogenized.

7 The `CGE_Government` Module

The government sector is currently simply modelled as one that balances its budgets in each year. This means that we include a single condition:

$$\sum_{s \in \mathtt{tax}_s} T_{(t,s)} + v_{(t,G)} i_t = v_{(t,G)}, \tag{8}$$

where $\mathtt{tax}_s$ is the subset of sectors that have transfers to/from the government, $v_{(t,G)}$ is the total government assets. To ensure this, one of the many tax rates of the economy has to be endogenized.[12]

Note, however, that there is no guarantee that the budget is balanced in the baseline year – in fact, it is highly improbable. To account for this, we only enforce the balanced budget for $t > t_0$.

8 The `GmsEquilibrium` Module

In the small open economy, the modules are coupled by adding the equilibrium condition for all goods:

$$\sum_{s \in \mathtt{dS}_n} q_{(t,s,n)} = \sum_{s \in \mathtt{dD}_n} q_{(t,s,n)}, \tag{9}$$

where $\mathtt{dS}_n$ indicates all sectors that supply good n, and $\mathtt{dD}_n$ all sectors that demand it.

9 Simulations in General Equilibrium

As a proof of concept, the following includes a couple of simple simulations in the model.

9.1 A 25% tax on the domestic good `a`

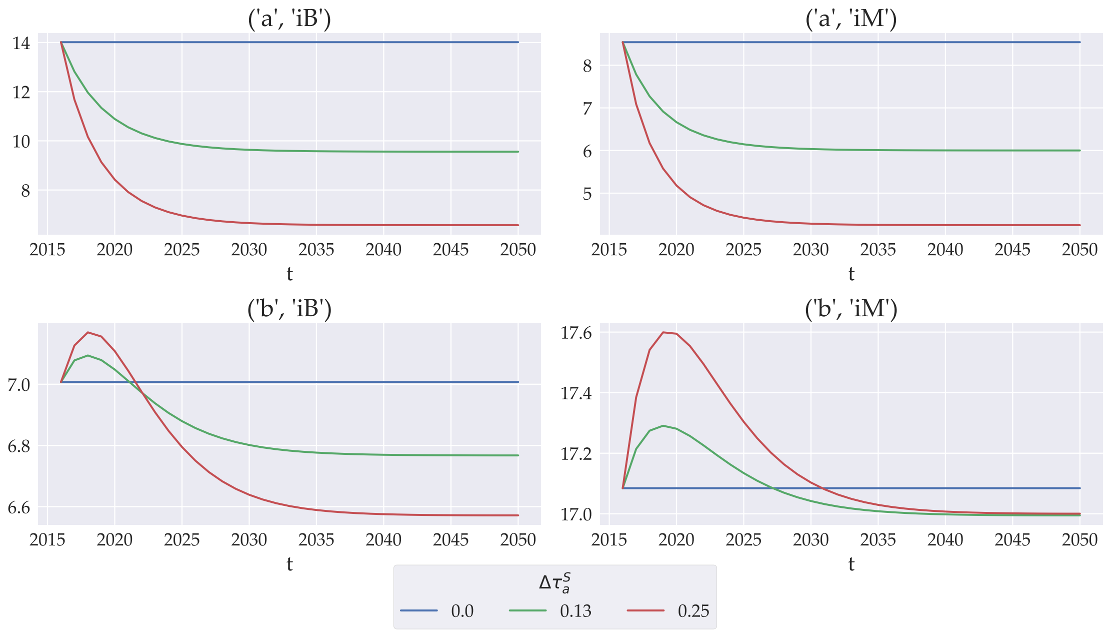
This is quite a substantial shock to the model as the sector `a` comprises roughly 1/3 of the entire domestic production.

Figure 9.1 shows the effect on building (`iB`) and machine capital (`iM`) in the domestic production sectors (`a`,`b`). The immediate effect of the tax is that the sector `a` becomes less competitive/more costly; thus, naturally, the sector reduces its stock of capital towards a new, lower level. The competing sector, `b`, faces a number of effects.

- **Costs of inputs:** The effective input price on `a` increases $\Rightarrow$ higher production costs and demand for durables $\downarrow$.
- **The relative price/substitution channel:** The price on its output, `b`, is now lower relative to `a` $\Rightarrow$ increased demand for `b` and demand for durables $\uparrow$.

[12] Currently, we use the labor income tax rate to do this.

Figure 9.1: Durables



- **Labor market channel:** As the price on consumption increases, households compensate – partially – by increasing its labor supply for a given wage rate. Thus, the production costs decreases and demand for durables ↑.
- **Price on investment goods:** The investment sector that produces building capital (`I_iB`) relies heavily on inputs from `a` – thus, the price of the investment good `I_iB` increases. This suggests that the demand for `iB` falls by more than for `iM`.

The net effect, in the short run, is that sector `b` increases its stock of durables. In the longer run, however, the price on `a` increases gradually as its stock of durables is brought down. This brings the stock of durables down for sector `b`.

Figure 9.2 illustrates the shadow cost of durables. These are a function of two fundamental components: (1) Installation costs of accumulating capital and (2) the cost of producing the corresponding investment goods.

For the sector `a`, the price initially decreases due to the sector decumulating capital. In the longer run, however, as installation costs tends to zero again, the costs are marginally higher, due to the costs of the investment goods (`I_iB`, `I_iM`).

Figure 9.2: Price on durables

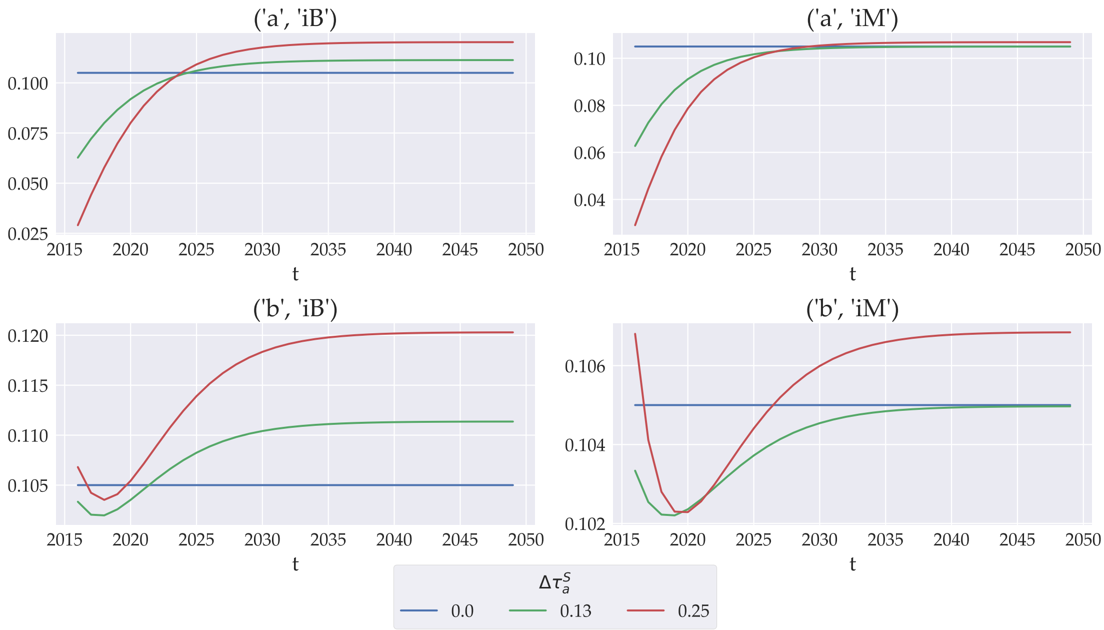


Figure 9.4 shocks the total supply of domestic goods. It confirms that `a` and durables decrease, whereas labor supply and `b` increases. Appendix F shows how the demand is split onto the different sectors.

Figure 9.3: Equilibrium prices

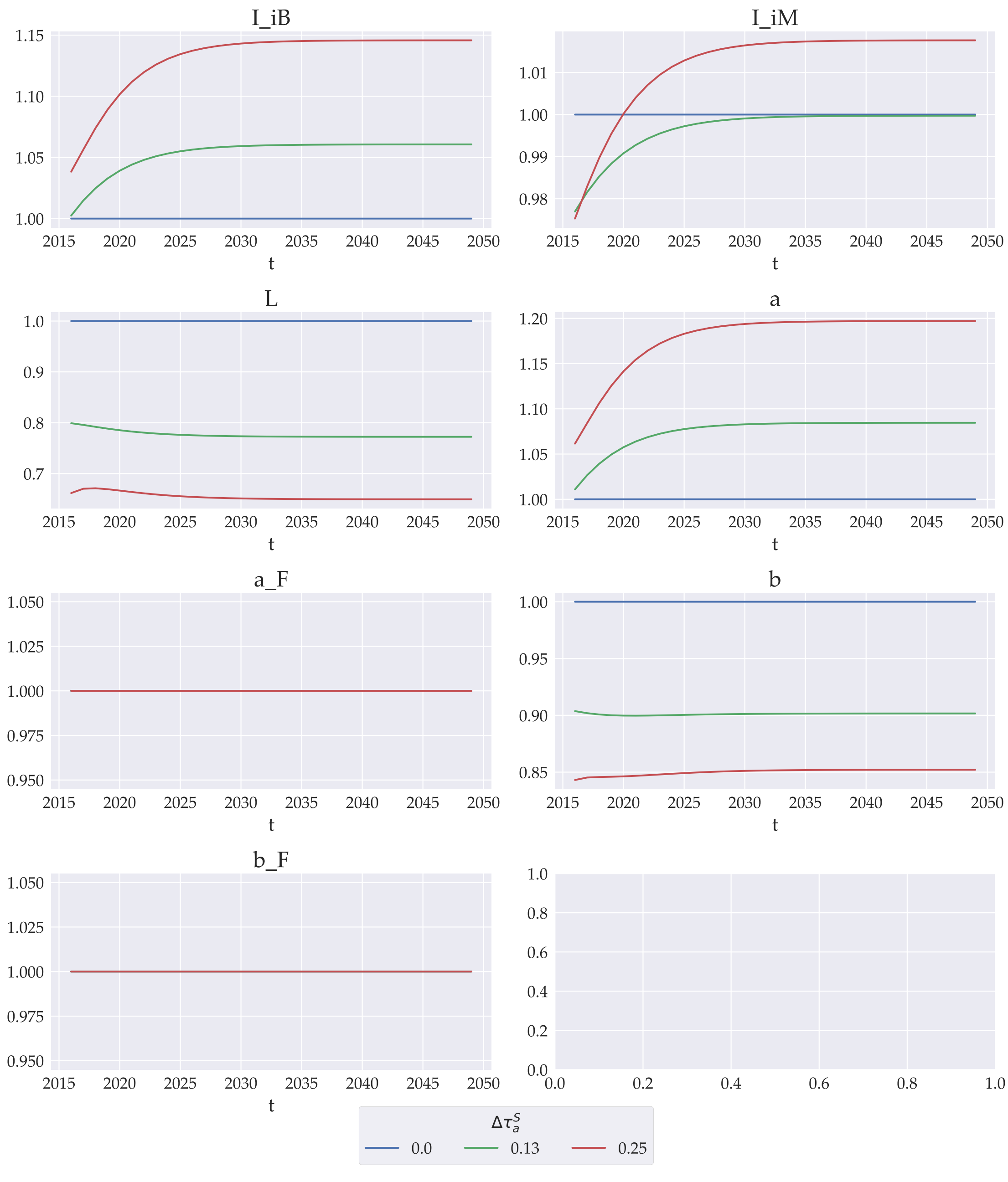
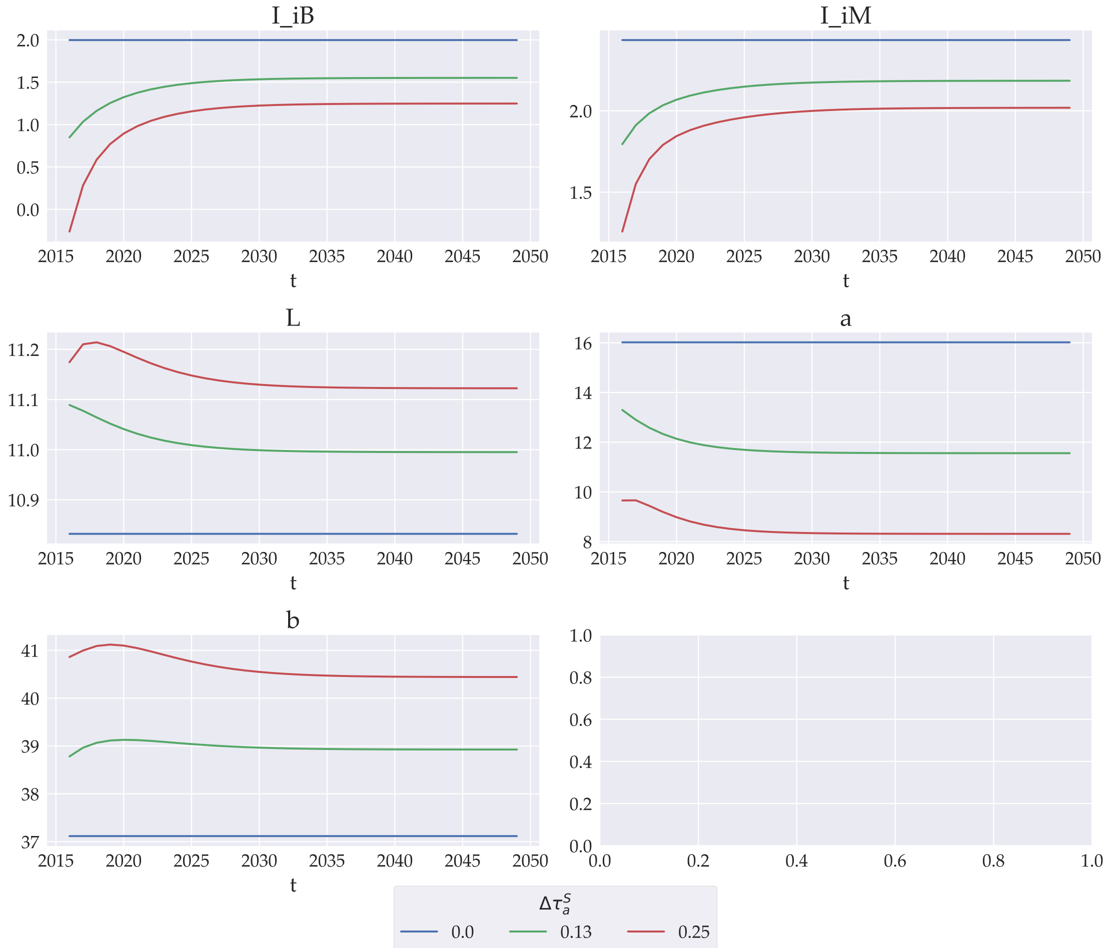


Figure 9.4: Domestic supply



9.2 A 25% reduction in initial capital

Figures 9.5-9.6 shows that the model gradually converges on *almost* the same long run equilibrium: Durables are gradually accumulated with the corresponding price decreasing as installation costs taper off.

Figure 9.5: Durables

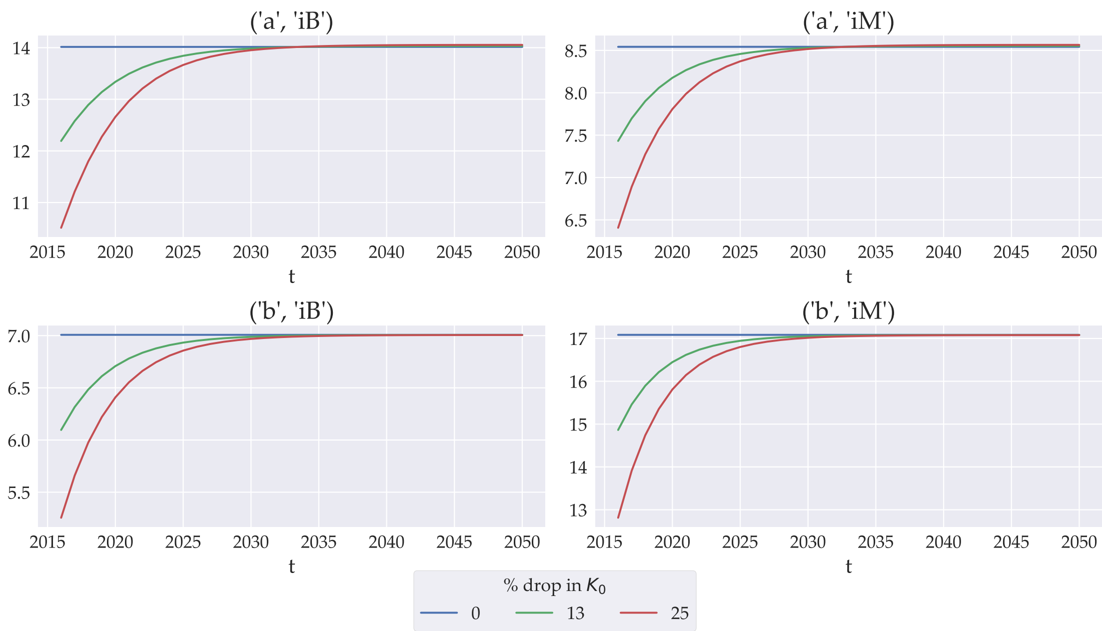
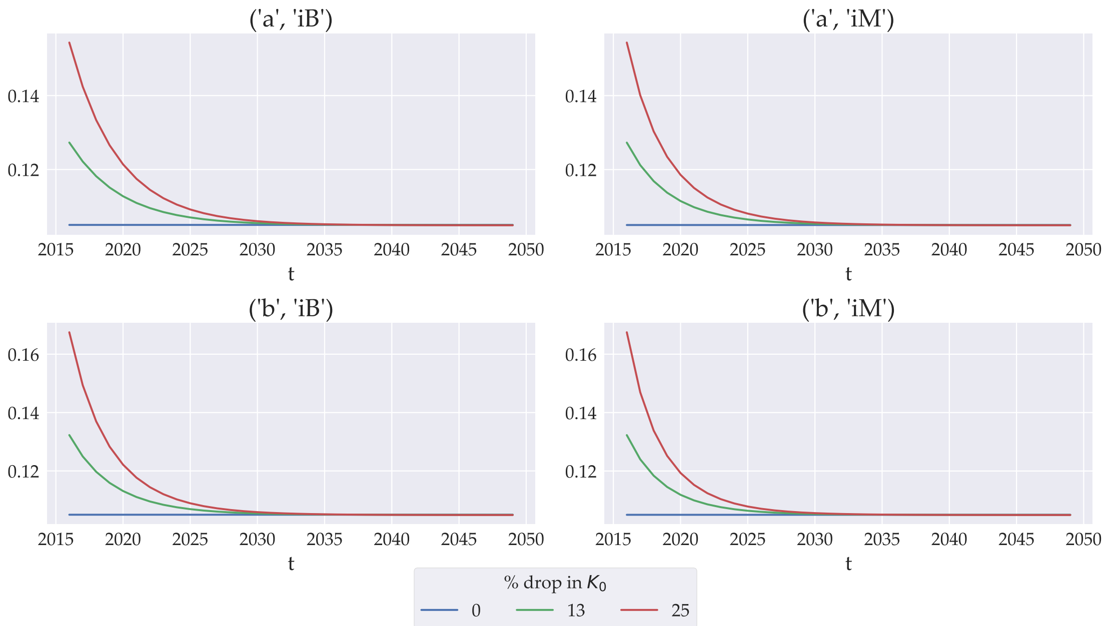


Figure 9.6: Price on durables



The reason why the model only *almost* converges on the same long run state, is that consumers' long run level of wealth is affected by the temporary drop – which in turn affects its preference for labor supply. Figures 9.7-9.8 shows that consumers smooth out its consumption levels, thus converging on a lower level of savings in the long run. The lower level of income and consumption is partially offset by an increase in the labor supply for a given wage rate.

Figure 9.7: Household savings

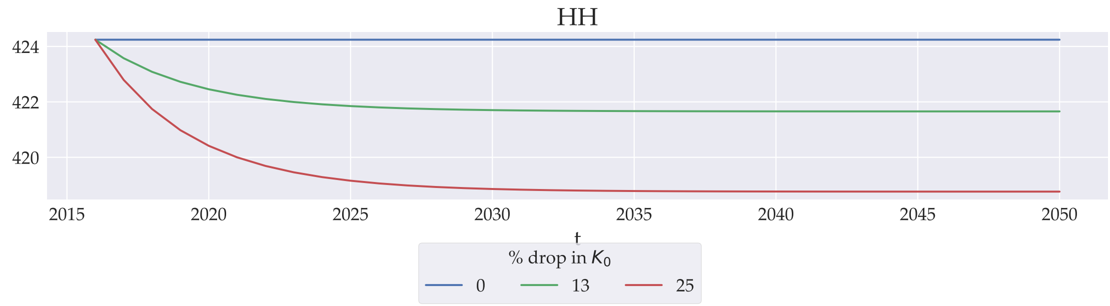
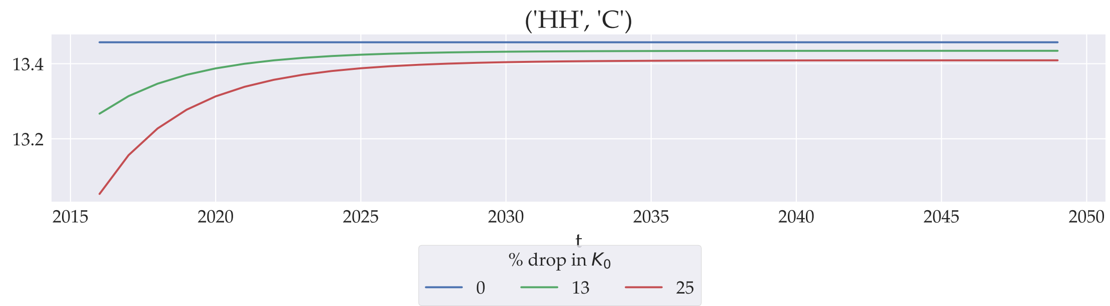
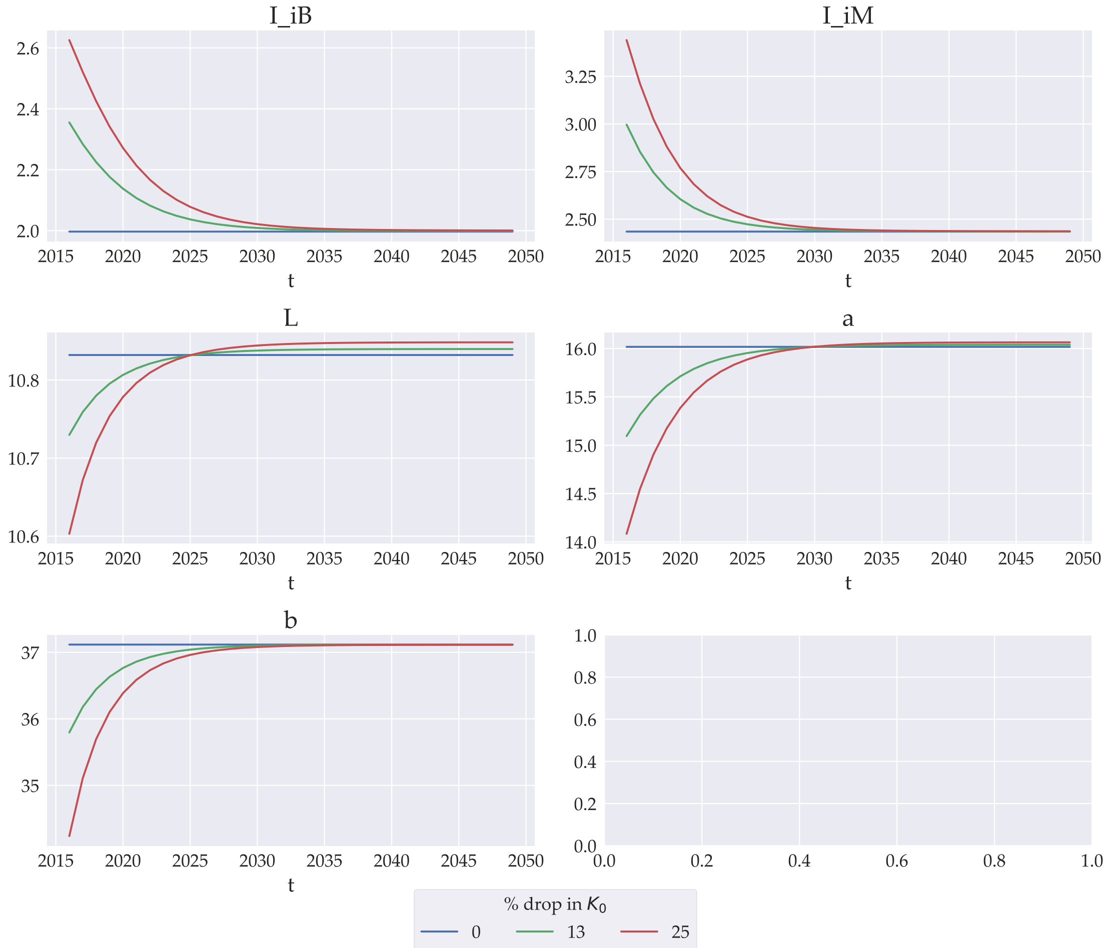


Figure 9.8: Household consumption



The small, but permanent lower consumption levels and higher labor supply (for a given wage rate) ripples over into other sectors. Figure 9.9 illustrates the effect on the supply of domestically produced goods: Labor supply increases and `a` slightly increases as well.

Figure 9.9: Domestic supply



Appendices

A Optimal investment in durables

Consider the problem of optimal accumulation of the durable K_t as presented in section 2.1. A couple of standard examples are:

Quadratic, static installation costs

We assume specifically that $\Psi_t = \phi/2(I_t/K_t - \delta)^2$. In this case, the shadow value on capital becomes (for $t_0 < t < T$):

$$p_t^k = R_t p_{t-1}^I \left[1 + \phi\left(\frac{I_{t-1}}{K_{t-1}} - \delta\right)\right] + p_t^I \left\{\frac{\phi}{2}\left[\delta^2 - \left(\frac{I_t}{K_t}\right)^2\right] - (1-\delta)\left[1 + \phi\left(\frac{I_t}{K_t} - \delta\right)\right]\right\} \tag{10a}$$

Imposing that $K_{T+1} = K_T$ such that $I_T = \delta K_T$ in the terminal period, this simplifies to:

$$p_T^k = R_T p_{t-1}^I \left(1 + \phi\left(\frac{I_{t-1}}{K_{t-1}}\right)\right) - (1-\delta)p_T^I. \tag{10b}$$

Quadratic, lagged installation costs

A second traditional way of introducing quadratic installation costs are on the form:

$$\Psi_t = \frac{\phi}{2}\left(\Delta_t\right)^2, \qquad \Delta_t \equiv \frac{I_t - I_{t-1}}{K_t}. \tag{11a}$$

In this case, the shadow value of capital becomes (for $t_0 < t < T$):

$$p_t^k = R_t p_{t-1}^I \left(1 + \phi\Delta_{t-1}\right) - p_t^I \phi\Delta_t \left(2 - \delta + \frac{\Delta_t}{2}\right) + (1-\delta)\left(\frac{p_{t+1}^I}{R_{t+1}}\phi\Delta_{t+1} - p_t^I\right) \tag{11b}$$

In the terminal period, we approximate this function by assuming no installation costs:[13]

$$p_T^k = R_T p_{t-1}^I \left(1 + \phi\Delta_{t-1}\right) - (1-\delta)p_t^I. \tag{11c}$$

In this case, we note that the condition in $t_0 + 1$ requires data on I_{t_0-1} to compute Δ_{t-1}; here, we would generally assume that $I_{t_0-1} = \delta K_{t_0}$ (if we don't have data on it, that is).

B Multiple output structure

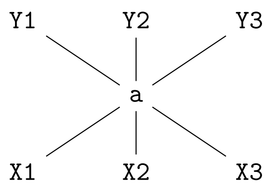
B.1 Without block III: Profits and prices

Consider the case where a sector produces multiple outputs. The standard calibration method of nested trees is to (1) exogenize quantities and (2) endogenize share parameters. For a sector that

[13] This is only an approximation because, in general, $I_{T_1} \neq I_T$; however, if T is sufficiently large, the model should naturally be close to this equilibrium. We use the approximation because it significantly improves the numerical stability of the system.

produces multiple outputs, however, this does not work. Consider for example the case in figure B.1 with a sector that relies on three inputs (X) and produces three outputs (Y): In the baseline

Figure B.1: Production nest



mode:

- **Exogenous:** Input prices (`pD[x]`), share parameters (`mu`), elasticities (`sigma`,`eta`), and supply (`qS[y]`).
- **Endogenous:** Input quantities (`qD[x]`), intermediate price + quantity (`qD[a]`, `pD[a]`), output prices (`pS[y]`).
- **Equations:** Demand equations for `qD[x]`, zero profit for `qD[a]`. Demand equations for `qD[y]`, zero profit for `qD[a]`. 8 equations in total.

In the calibration mode, we would initially try to make all share parameters endogenous (6 of them here), and in turn make quantities exogenous. Considering the two nests separately the logic would be:

- Input nest: Make demand exogenous (`qD[x]`) and share parameters (`mu[a,x]`) endogenous. This fits – there are 3 of each.
- Output nests: Make demand exogenous (`qD[a]`) and share parameters (`mu[y,a]`) endogenous. The problem is that we only have 1 quantity, but 3 share parameters. Instead, we'll assume that the two residual share parameters are determined by price levels, i.e. by fixing two of the three supply prices (`pS`[Y]).

B.2 With block III: Profits and prices

Now, assume that we on top of this system had an equation for each output:

$$\texttt{p[Yi]} = (1+\texttt{markup})(\texttt{pS[Yi]}+\texttt{tauS[Yi]})$$

In this case, including this equation we could:

- In baseline: Endogenize `p[Yi]`. This works as long as all goods are produced by only one sector. If two sectors produced the same good, this would imply more equations than variables, however.
- In calibration: Exogenize `p[Yi]`, endogenize `markup`, and endogenize the corresponding cost prices `p[Yi]` – where two of them were kept exogenous before. Note that this would work

even if two sectors produced the same goods. This is because the calibration stage has access to technology parameters that can fundamentally ensure that the cost index of the two sectors are exactly the same.

The fundamental reason why this happens – the calibration method works even when sectors produce the same goods - is because these constant returns to scale models have an embedded equilibrium logic: The supply side of the model (with constant returns to scale) ultimately determines the long run cost/price level of the goods. The levels are determined by the demand side, where some type of nonlinearity needs to be in play for an equilibrium to exist (otherwise, we have an indeterminate scale of the economy). So, allowing two sectors to produce the same good (i.e. the equilibrium price has to be the exact same) can only occur if technologies and prices conspire to produce the exact right outcome. This is what happens in the calibration, but, can only be done by relying on technology adjustments.

C Share parameters from value shares

One way to establish share parameters for intermediate goods is:

i. Set up a model without investments – and treat durables as simple non-durables. Define the value of durables from $p_K qK$ with p_K being defined using a static user cost definition (e.g. $(R/(1+\pi)+\delta-1)$.

ii. Using IO data on values on inputs (`vD`) and outputs (`vS`), solve for value shares for the intermediate nests by: (i) Let all knots in nesting trees be defined as the sum of related branches, (ii) define `mu[s,n,nn]` as the value of branch (`vD[s,nn]`) relative to knot (`vD[s,n]`).

iii. For all intermediate goods, define the quantities `qD[s,n]` $=$ `vD[s,n]` from the value shares. Use the `CGE_Production.Production` module to make a static calibration, i.e. an identification of share parameters that make sure that all intermediate prices $=1$.

iv. Use the static calibration solution to initialize an instance of `CGE_Production.Production_ExoMu`. Add durables to this setup and re-calibrate dynamically.

D A simple Ramsey setup

Consider the following model:

$$U = \sum_t \beta^t u_t \tag{12a}$$

$$u_t = \frac{C_t^{1-\theta}}{1-\theta} - \frac{\xi}{1+\xi}\left(\frac{L_t}{\gamma}\right)^{\frac{1+\xi}{\xi}} \tag{12b}$$

$$s_{t+1} = s_t R_t + w_t L_t - p_t C_t. \tag{12c}$$

Optimization wrt. s_t, C_t, L_t yields:

$$C_t^{-\theta} = \lambda_t p_t \tag{13a}$$
$$(L_t/\gamma)^{1/\xi} = \lambda_t w_t \tag{13b}$$
$$\lambda_t = \beta R_{t+1}\lambda_{t+1}. \tag{13c}$$

where λ_t is the shadow variable on the law of motion. Combining the three we have:

$$L_t = \gamma\left(\frac{w_t}{p_t C_t^{\theta}}\right)^{\xi}$$
$$C_t = \left(\beta R_t \frac{p_{t-1}}{p_t}\right)^{\frac{1}{\theta}} C_{t-1}$$

This is the basic system applied in the main section, except for the component C being a CES-aggregate.

D.1 Calibration issues

First, note that we only impose the Euler equation for $T > t > t_0$ (emphasized in the main text). In general, the terminal period of the model (T) is only used to include TVC conditions. Thus, we do not e.g. have a price p_T on the consumption good. If we were to impose a TVC condition for consumption – assuming the model reaches a (balanced) steady state in T – the Euler equation and TVC conditions would read (long run values with asterisks):

$$C_T = \left(\beta R^* \frac{1}{1+\pi^*}\right)^{\frac{1}{\theta}} C_{T-1}$$
$$C_T = (1 + g_{TVC}^C) C_{T-1}.$$

This illustrates that we cannot impose both conditions. Instead of imposing any TVC condition, we rather note that the TVC condition for consumption is *implicit* by the combination of parameters $(\beta, R^*, \pi^*, \theta)$.

A second issue with regards to the calibration comes from standard IO identification of prices/quantities. To see the issue, consider the case with $T = 3$ years and assume that the first year $t = 1$ is the baseline year for which we have data on C_1, p_1, L_1, w_1. The system of equations that we solve for are then given by:

$$C_2 = C_1\left(\beta R_2 \frac{p_1}{p_2}\right)^{\frac{1}{\theta}}$$
$$L_1 = \gamma\left(\frac{w_1}{p_1 C_1^{\theta}}\right)^{\xi}$$
$$L_2 = \gamma\left(\frac{w_2}{p_2 C_2^{\theta}}\right)^{\xi}$$
$$s_2 = s_1 R_1 + w_1 L_1 - p_1 C_1$$

$$s_3 = s_2 R_2 + w_2 L_2 - p_2 C_2$$
$$s_3 = (1+g)s_2.$$

In the calibration state, we use γ to ensure the observed level L_1. The level of consumption, C_1, should, however, also be targeted.

We can combine the equations into a single requirement on the form:

$$\begin{aligned}
s_1 &= C_1\gamma_1 - C_1^{-\theta\xi}\gamma_2, \\
\gamma_1 &\equiv \frac{p_2\left(\beta R_2 \frac{p_1}{p_2}\right)^{\frac{1}{\theta}} + p_1\left(R_2-(1+g)\right)}{(R_2-(1+g))R_1} \\
\gamma_2 &\equiv \frac{\gamma\left(\frac{w_1}{p_1}\right)^{\xi}\left(R_2-(1+g)\right) + \gamma\left(\frac{w_2}{p_2}\right)^{\xi}\left(\beta R_2\frac{p_1}{p_2}\right)^{-\xi}}{(R_2-(1+g))R_1}
\end{aligned}$$

That is, they are both positive functions of parameters (where $R > 1+g$).

E Partial equilibrium simulation – Production

The appendix includes simulations of the production module in five different shocks.

E.1 Add a 100% tax on demand for the good `a`

Prior to the shock, the model has been calibrated to a balanced growth path. The price on `a` is set to one – and the shock is then to add a tax of one as well. The shock is done in two steps to show the effect of an intermediate level shock.

The main results can be summarized as follows:

- The supply of the two production sectors is kept constant in this partial equilibrium. Thus, the most prominent effect is that they substitute from `a` to other, cheaper inputs.
- Thus, both sectors start accumulating durables (`iB`,`iM`) cf. figure E.1; with installation costs of durables, the shadow price on them increase temporarily (E.2).
- Generally, for non-durables other than `a`, the demand increases permanently and the equilibrium price of both `a` and `b` increase (figures E.3, E.5). This is in part driven by the direct higher cost of the input `a`, but in the short run, also because of the transition costs of accumulating durables.

Figure E.1: The effect on durables, q

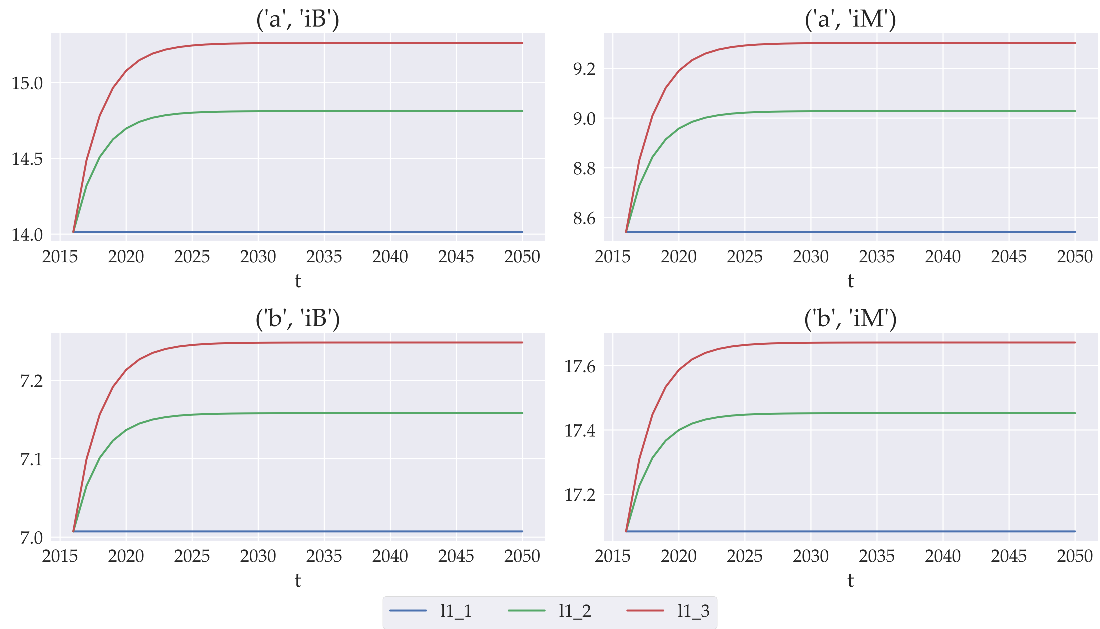


Figure E.2: The effect on durables, p

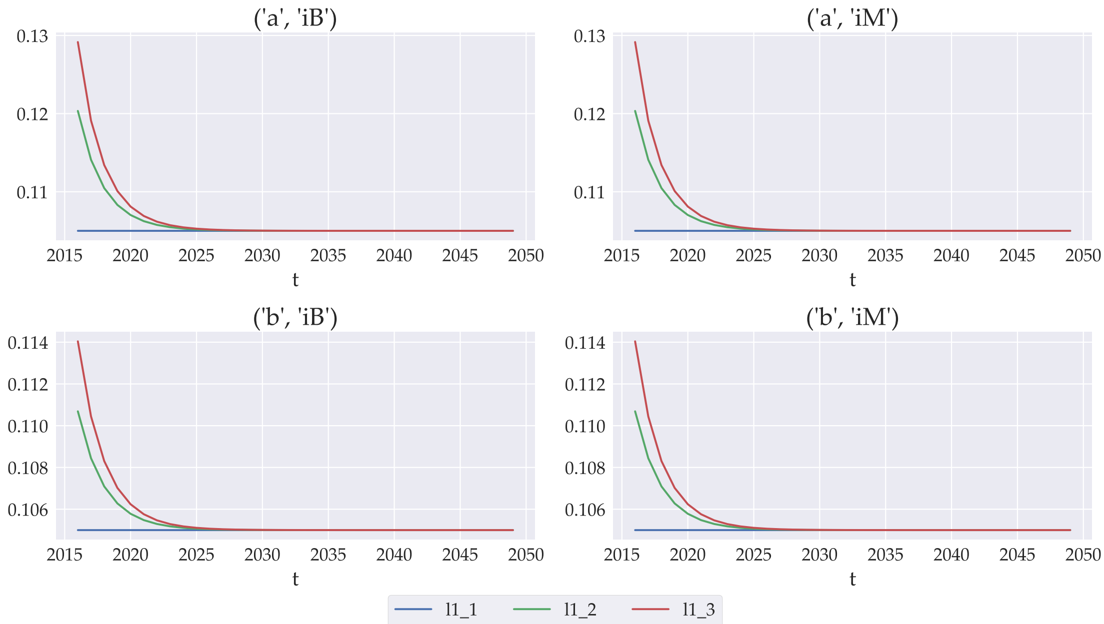


Figure E.3: The effect on non-durables, q

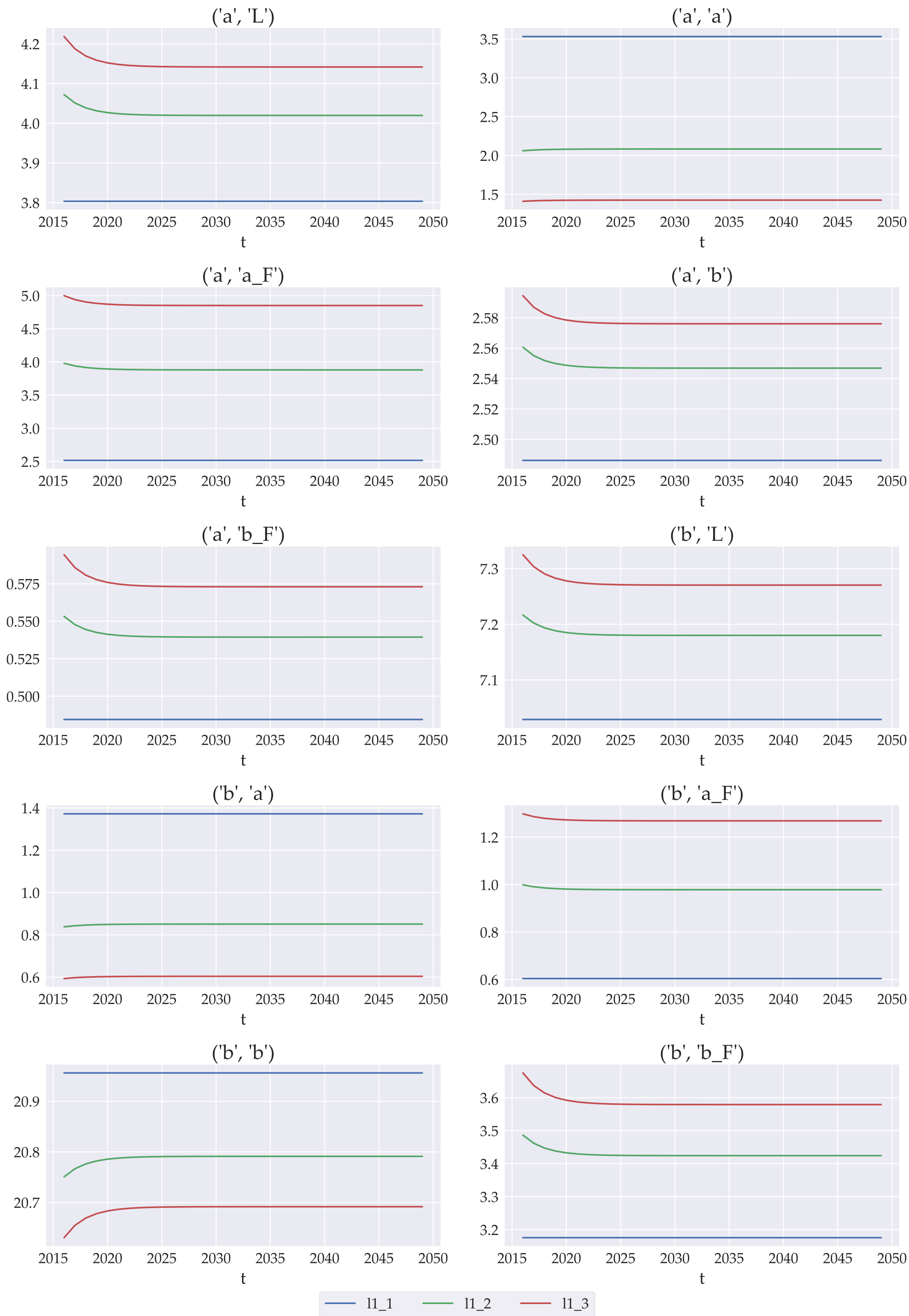


Figure E.4: The effect on non-durables, p

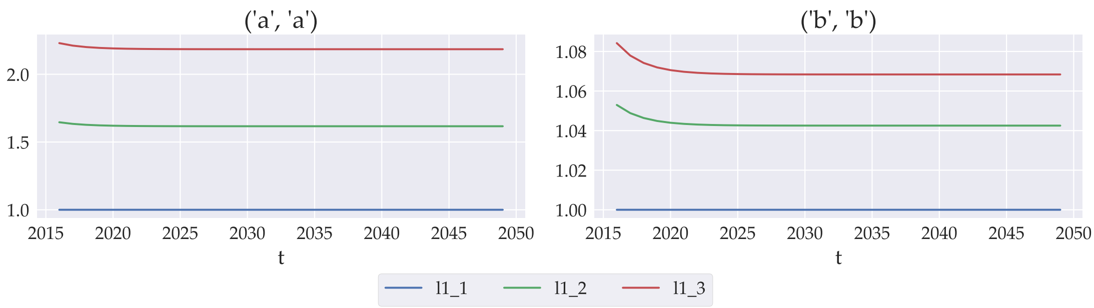
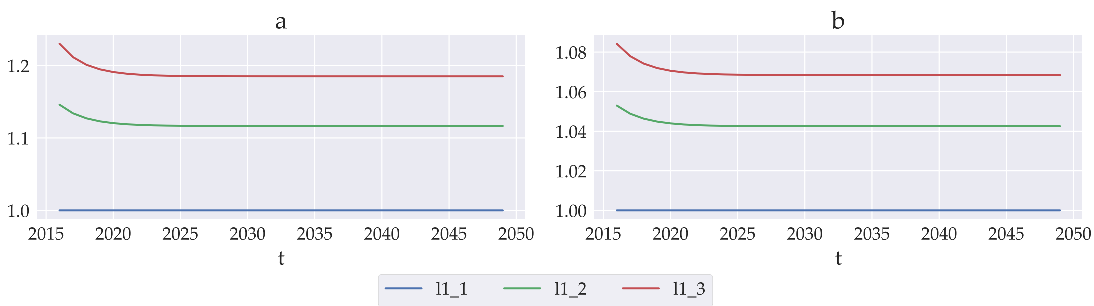


Figure E.5: The effect on equilibrium prices, p



E.2 Increase in output of 25%

The shock increases output from both sectors by 25%. The main mechanism at play here is again the gradual build-up in durables: Initially, the demand for non-durables increase sharply. As the equilibrium price on `a` and `b` increases in the short run – due to the adjustment costs – the increase in demand for these two is smaller than other goods. Over time, as the durables adjust, the demand for non-durables tapers off.

Figure E.6: The effect on durables, q

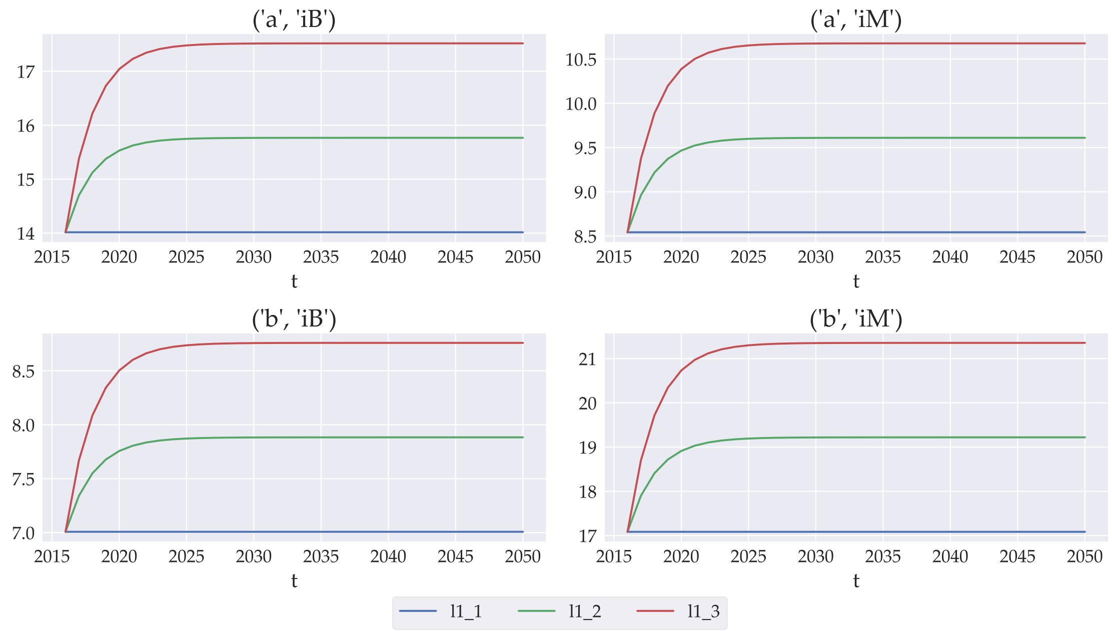


Figure E.7: The effect on durables, p

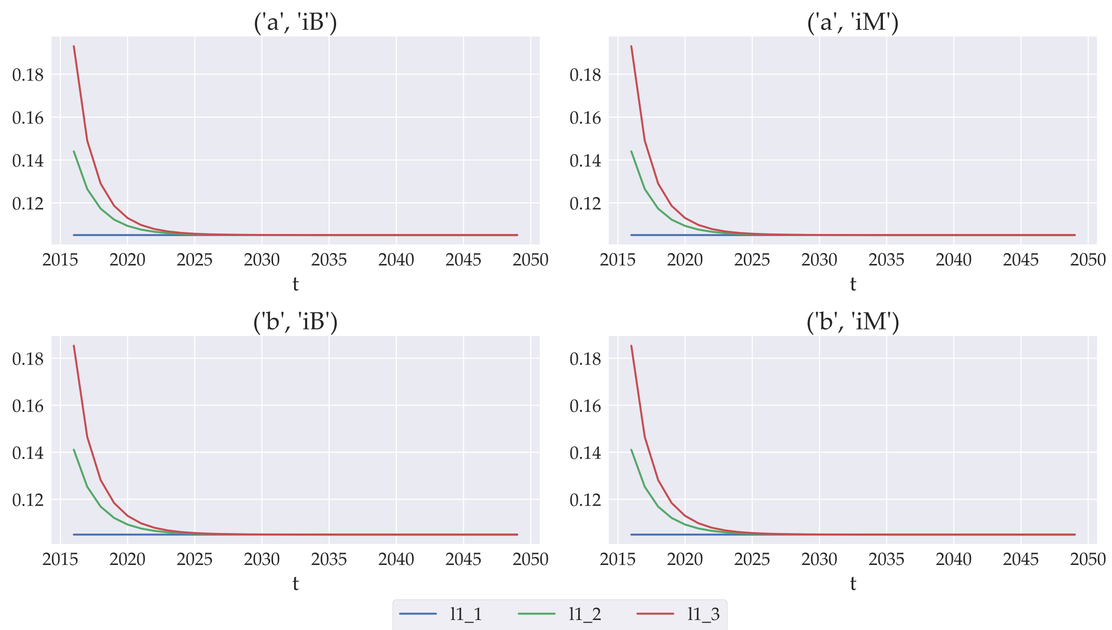


Figure E.8: The effect on non-durables, q

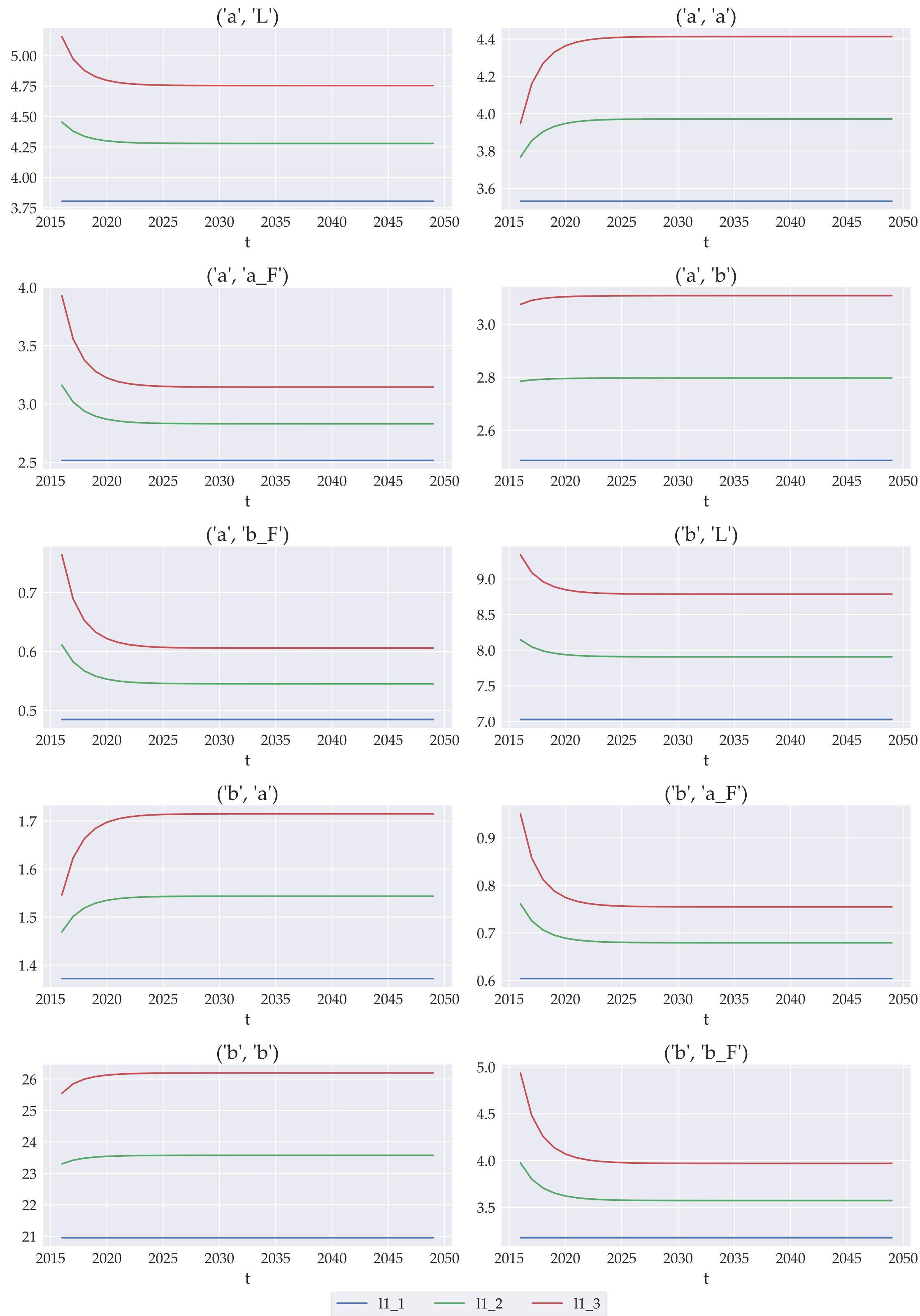


Figure E.9: The effect on non-durables, p

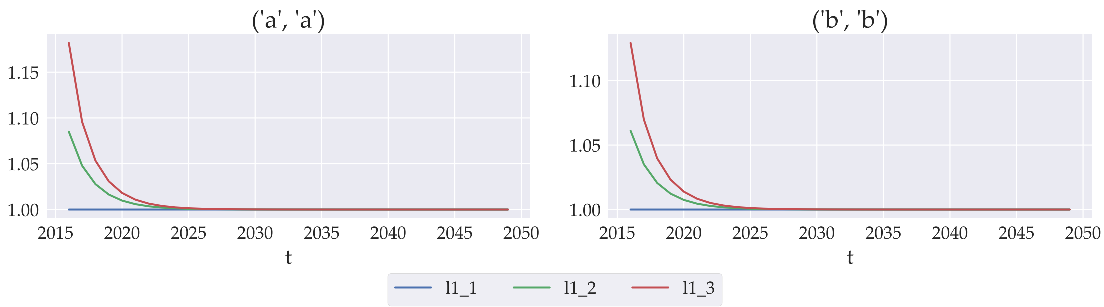
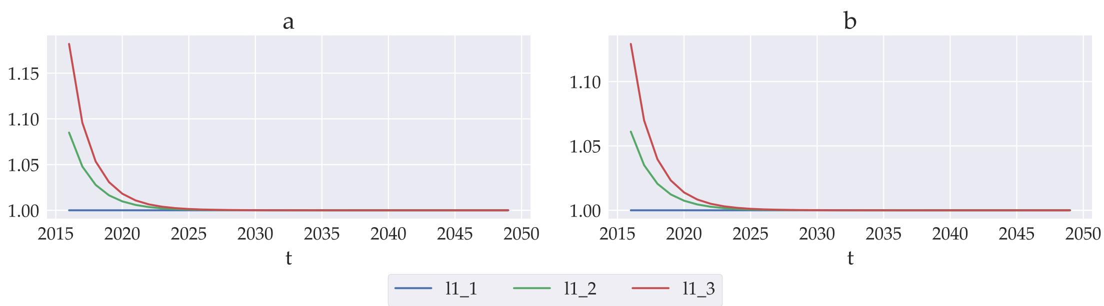


Figure E.10: The effect on equilibrium prices, p



E.3 Decrease in output of 25%

The shock lowers output from both sectors by 25%. The main mechanism is the same one as in the previous shock – now just reversed.

Figure E.11: The effect on durables, q

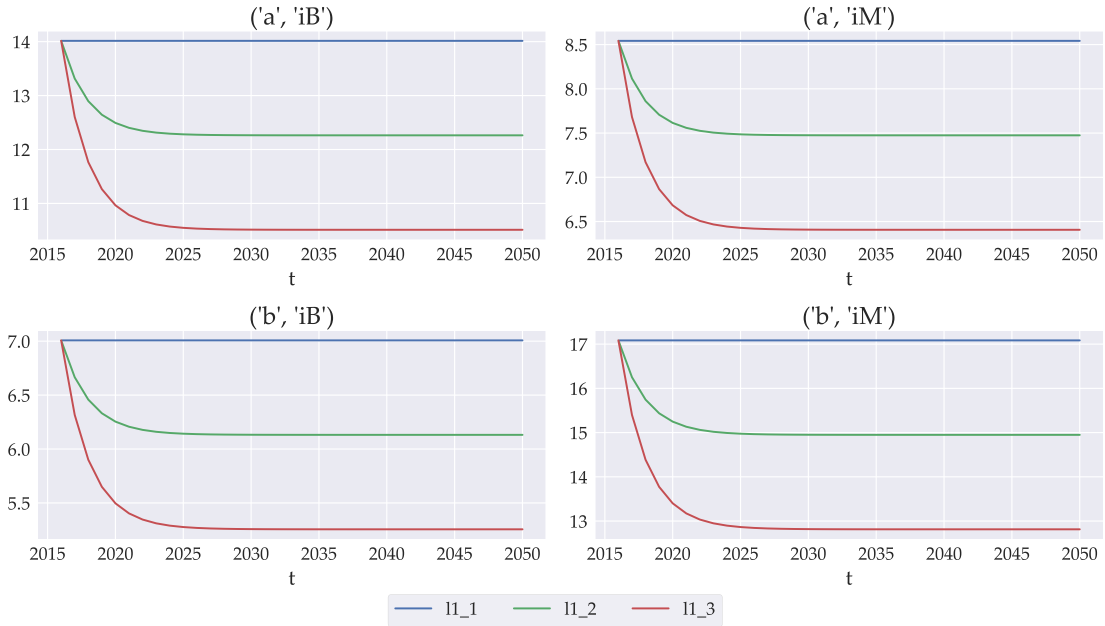


Figure E.12: The effect on durables, p

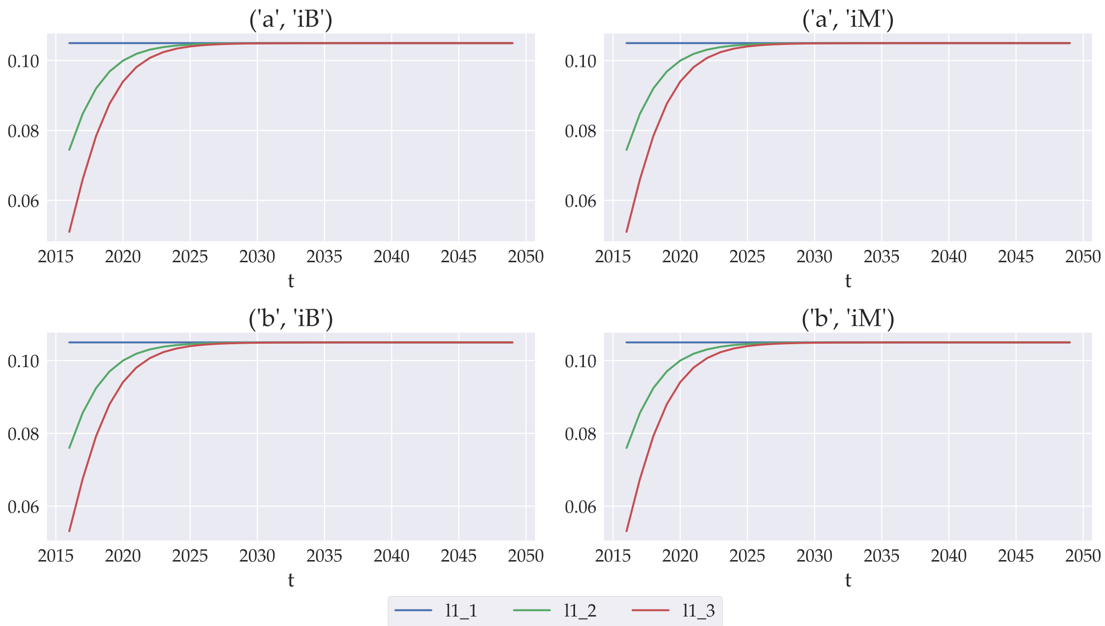


Figure E.13: The effect on non-durables, q

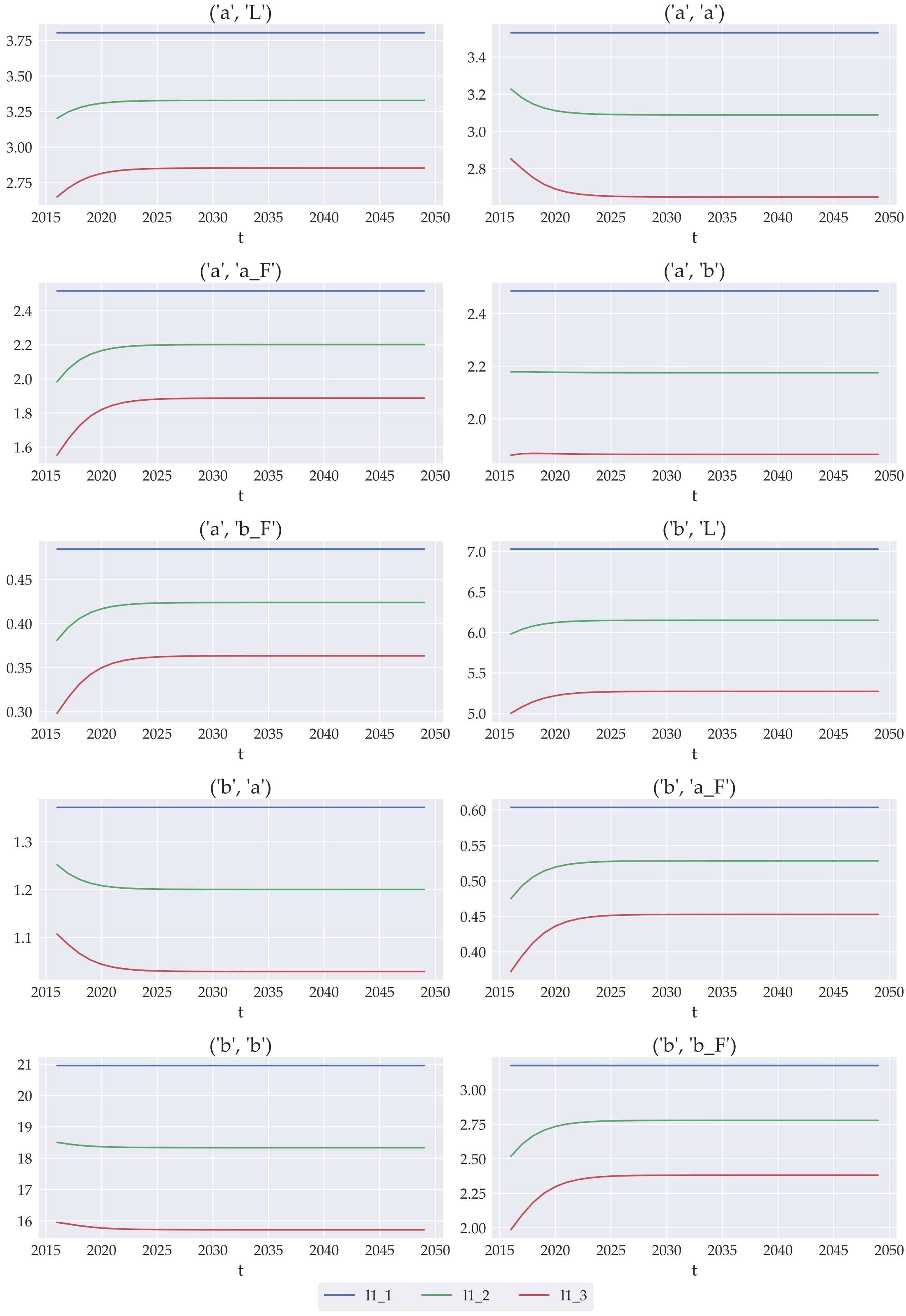


Figure E.14: The effect on non-durables, p

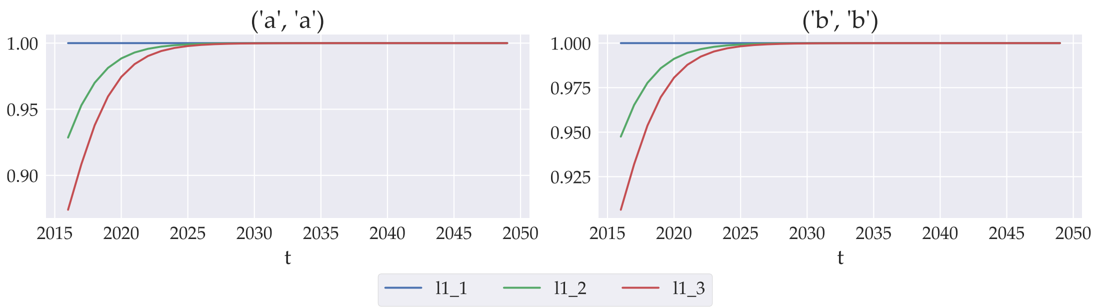
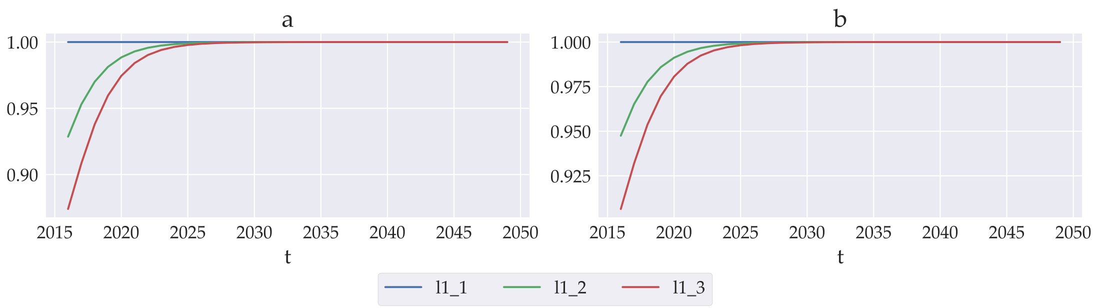


Figure E.15: The effect on equilibrium prices, p



E.4 Increase in initial durables stock of 25%

The shock increases the initial stock of durables by 25%.

Figure E.16: The effect on durables, q

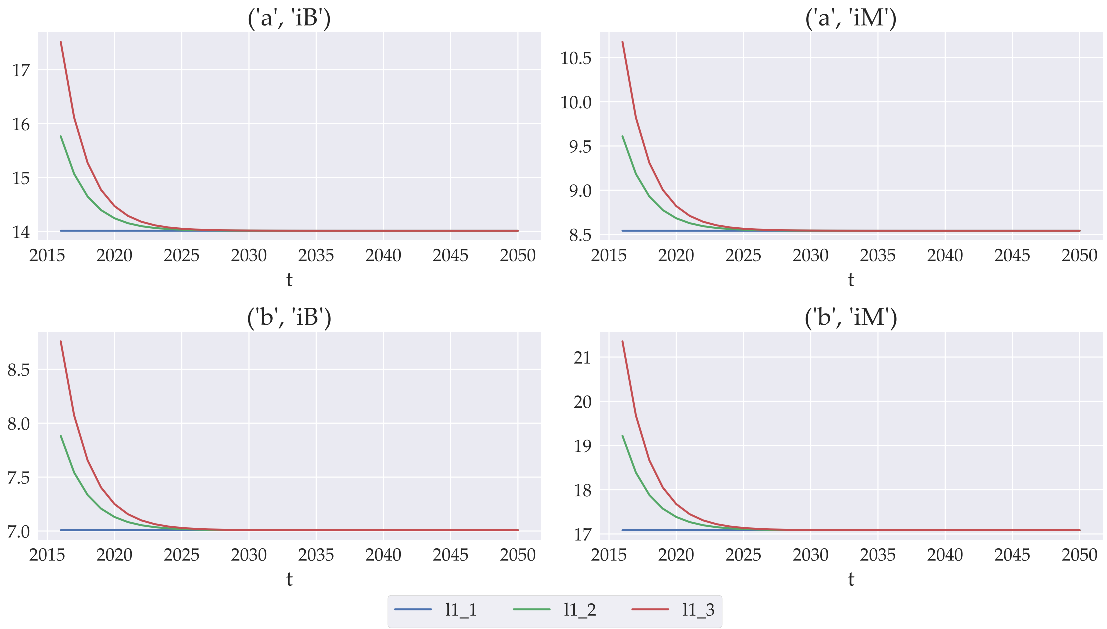


Figure E.17: The effect on durables, p

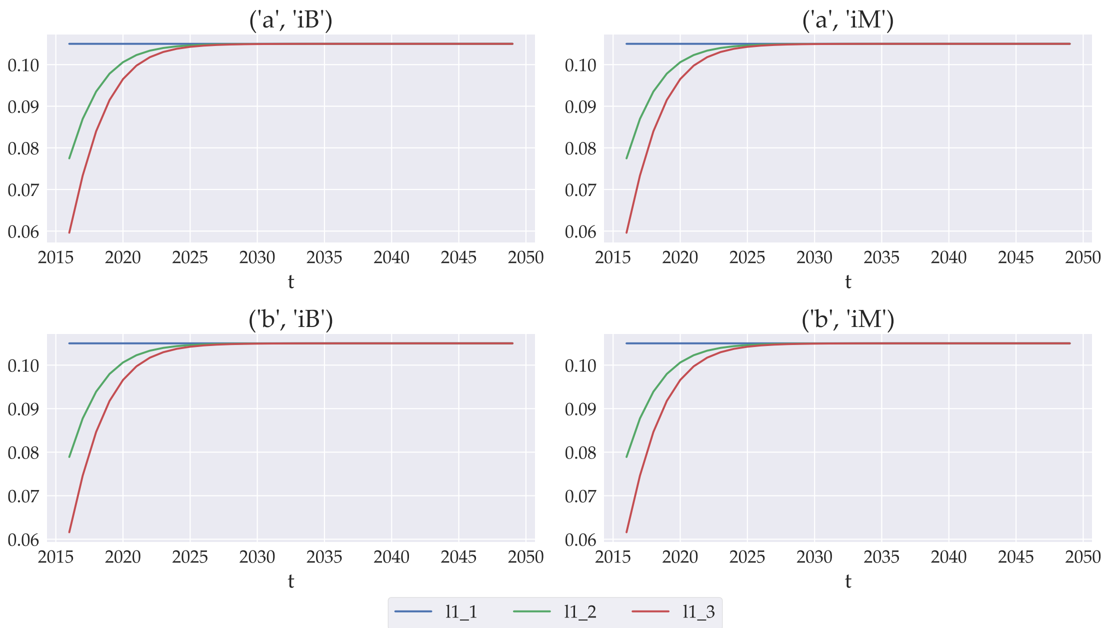


Figure E.18: The effect on non-durables, q

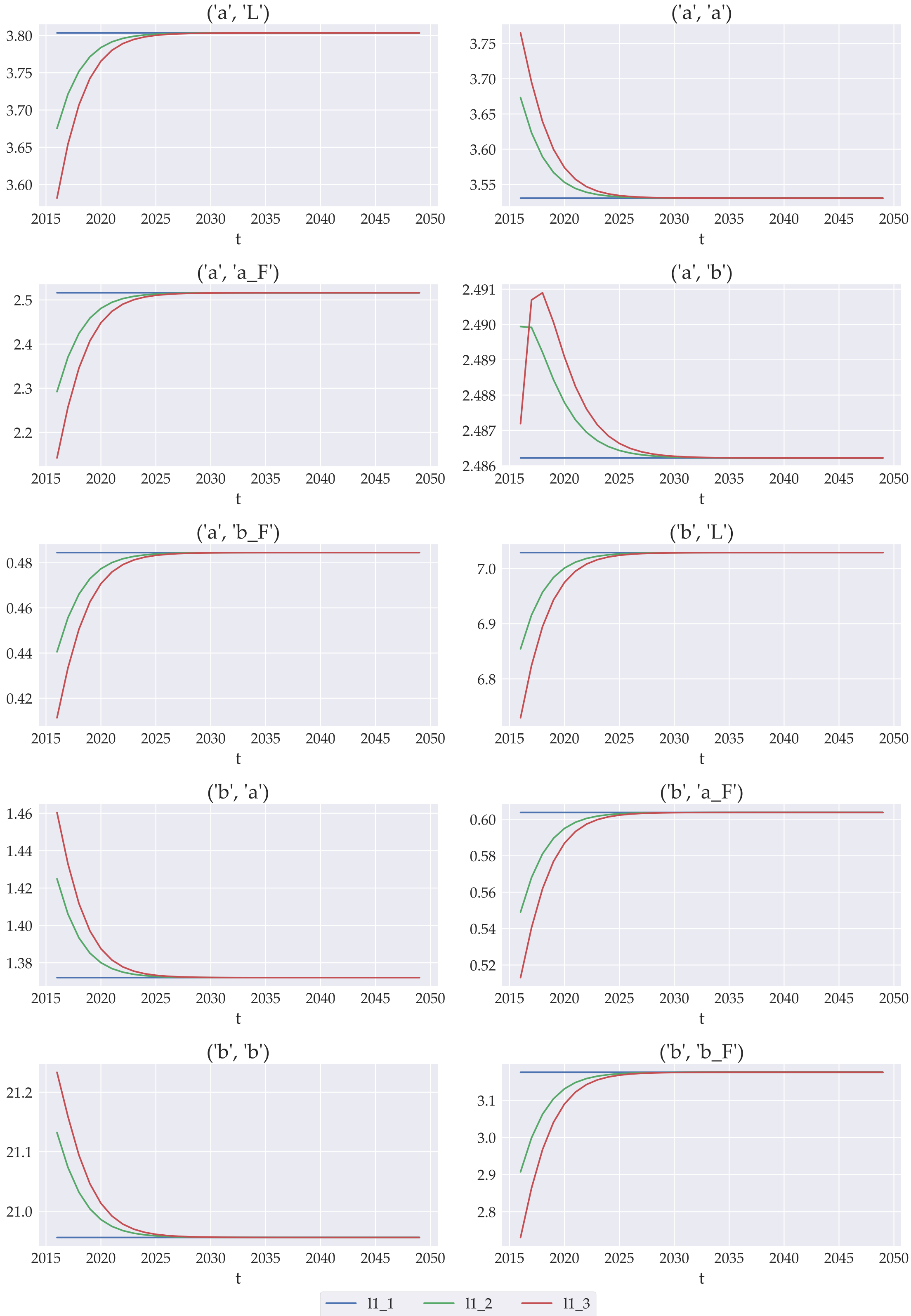


Figure E.19: The effect on non-durables, p

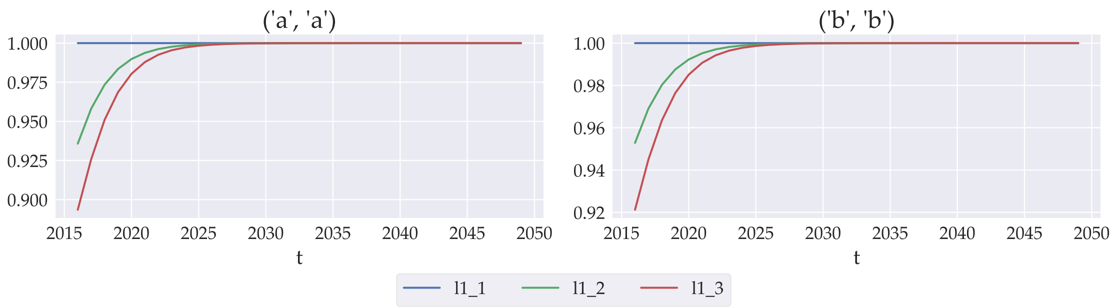
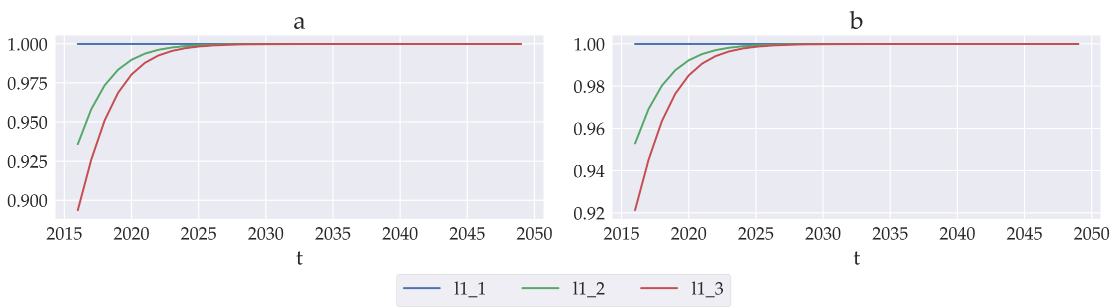


Figure E.20: The effect on equilibrium prices, p



E.5 Decrease in initial durables stock of 25%

The lowers the initial stock of durables by 25%.

Figure E.21: The effect on durables, q

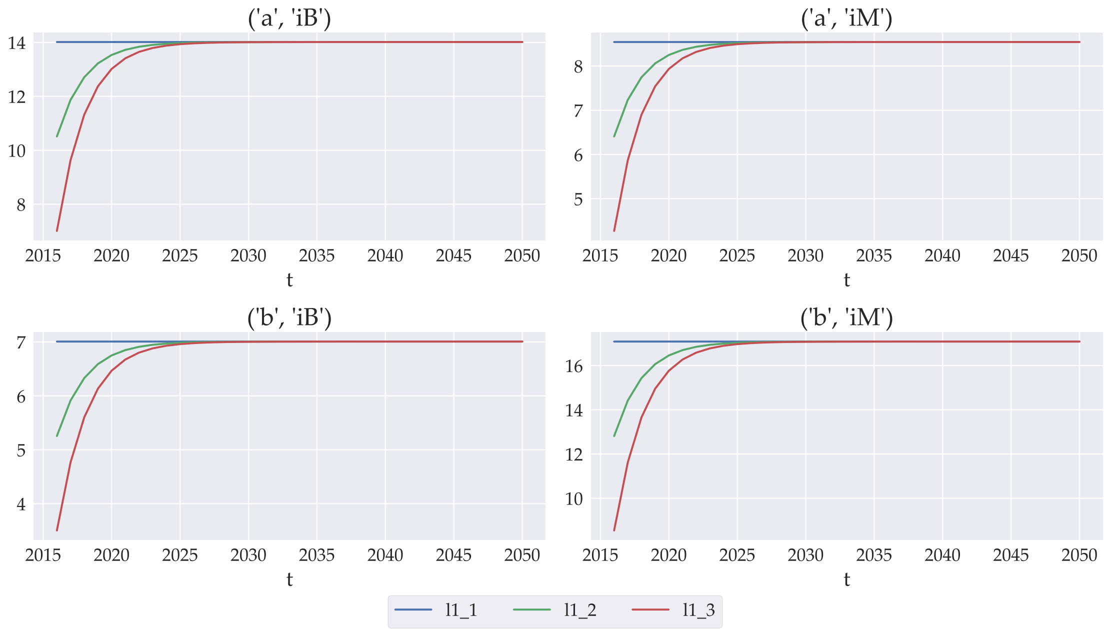


Figure E.22: The effect on durables, p

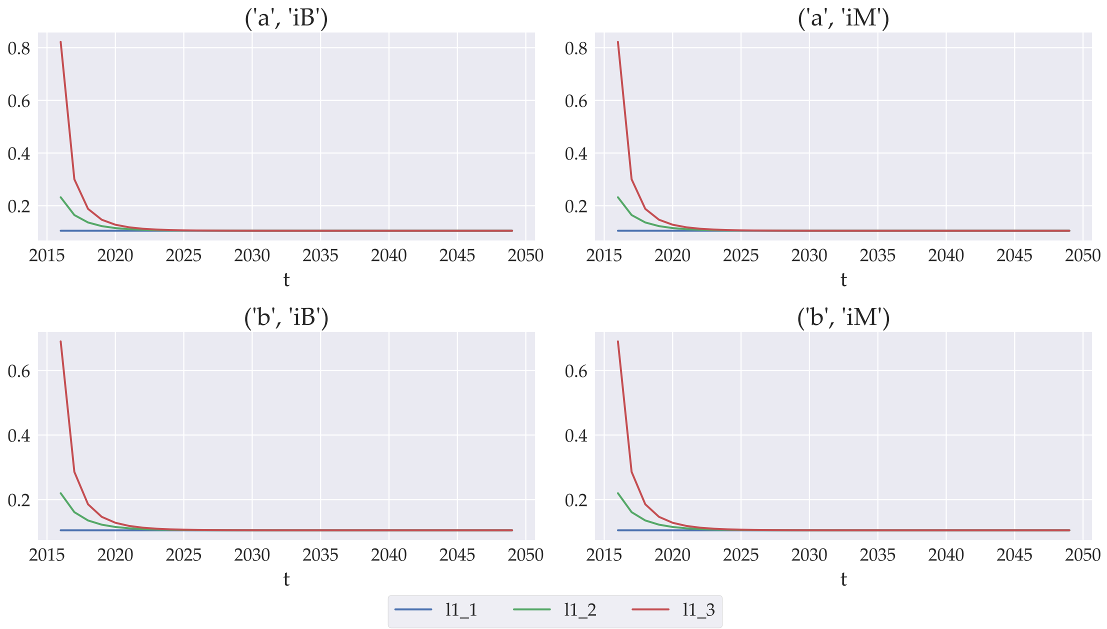


Figure E.23: The effect on non-durables, q

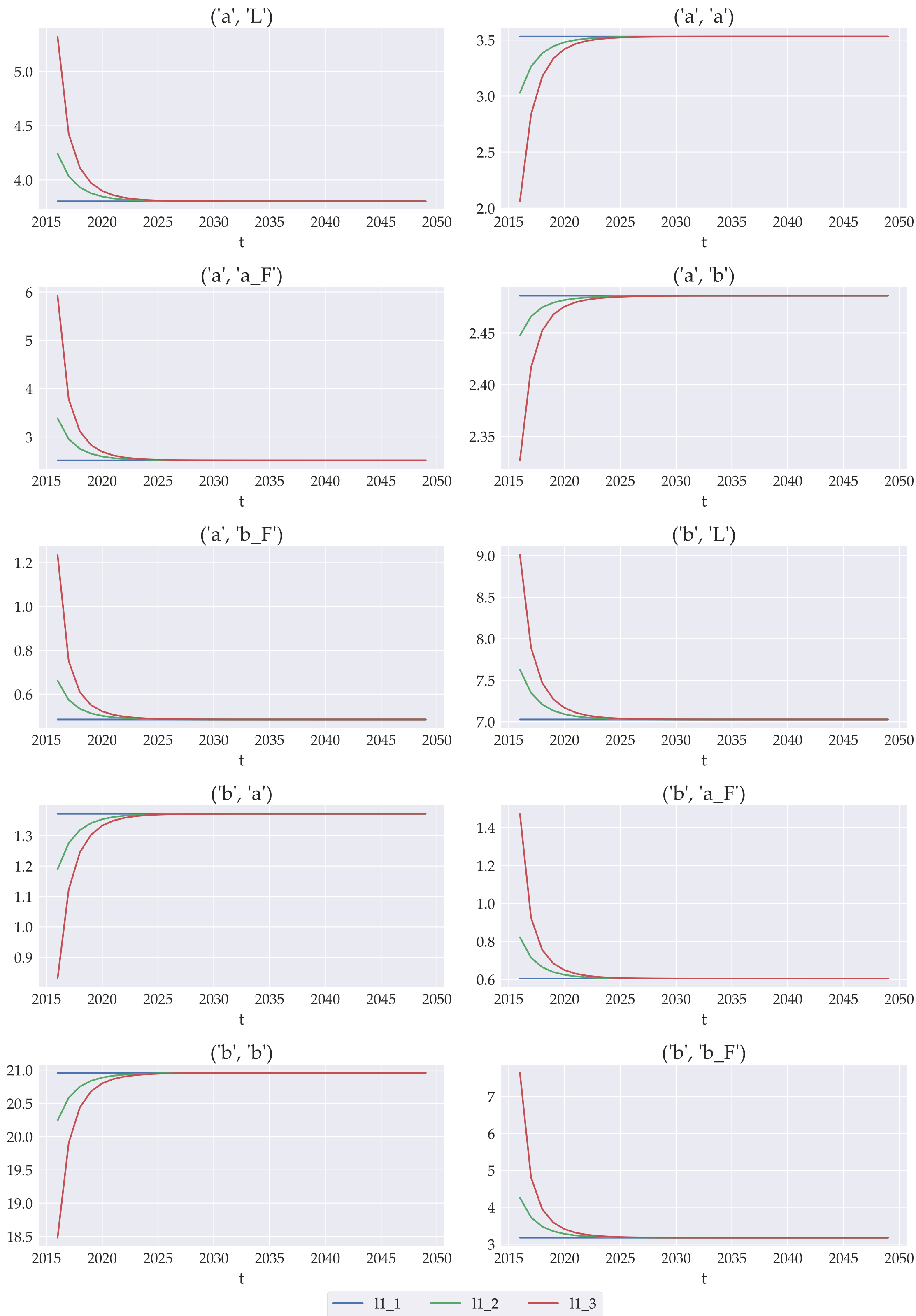


Figure E.24: The effect on non-durables, p

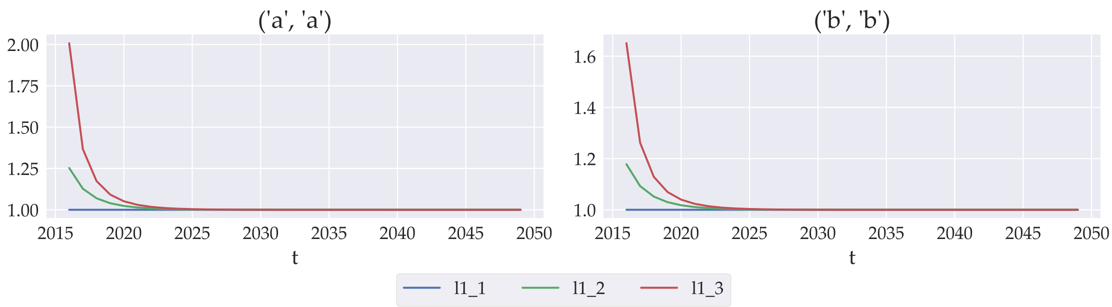
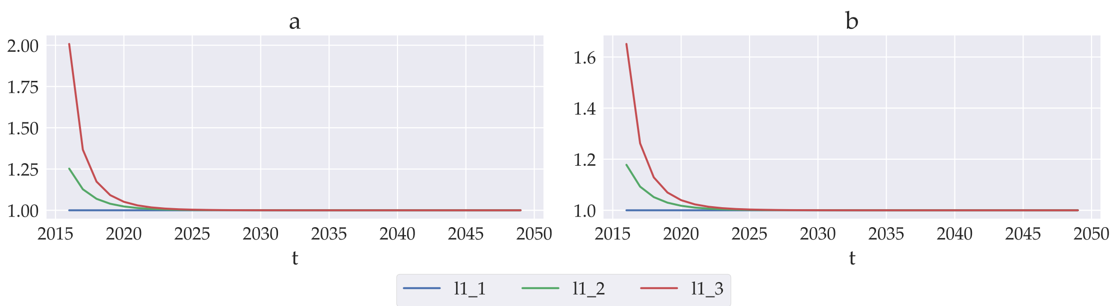


Figure E.25: The effect on equilibrium prices, p



F General Equilibrium Simulation

The appendix includes supplementary figures for simulations in the general equilibrium model.

F.1 A 25% tax on the domestic good a

Figure F.1: Investment sectors' demand

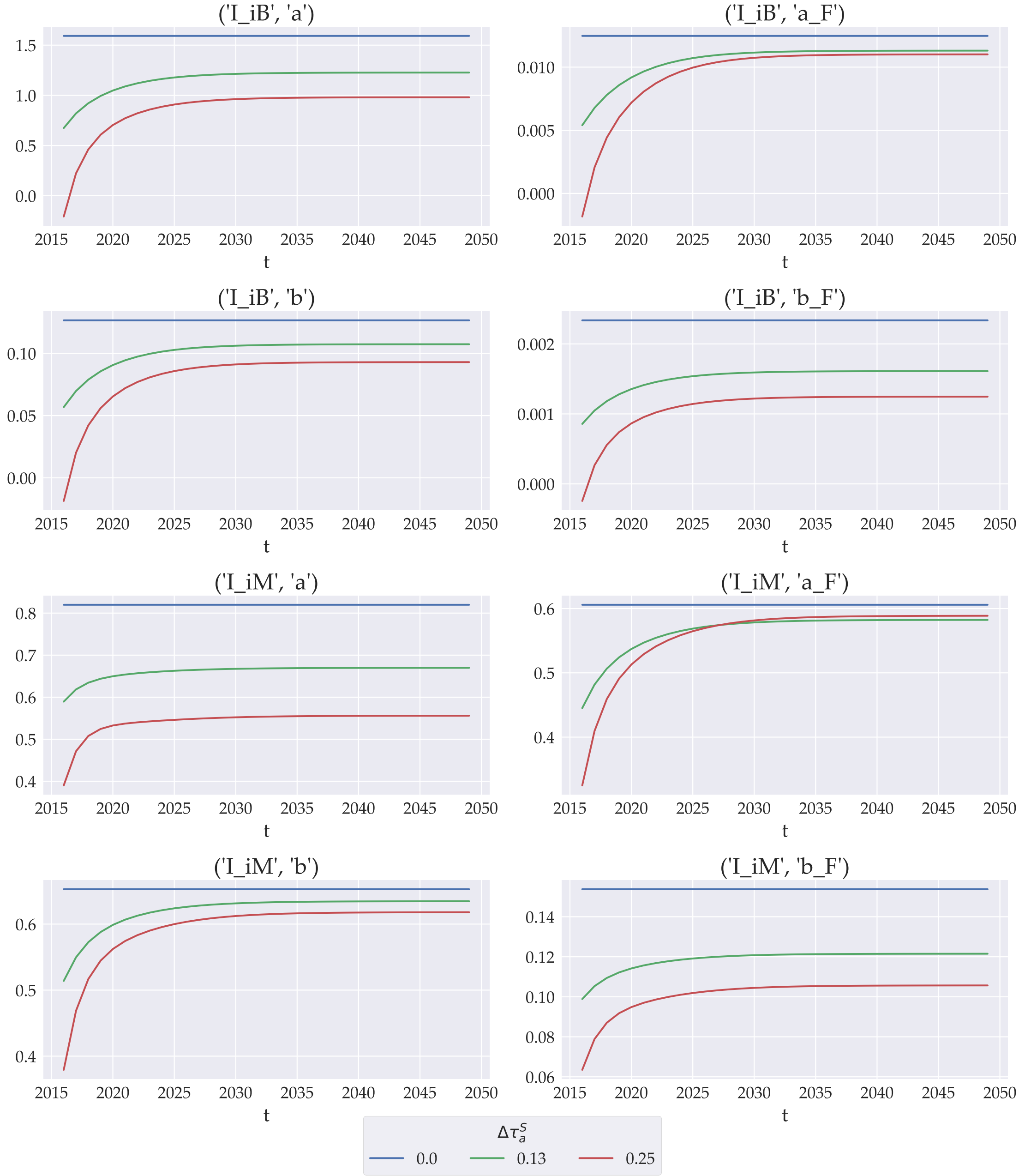


Figure F.2: Production sectors' demand

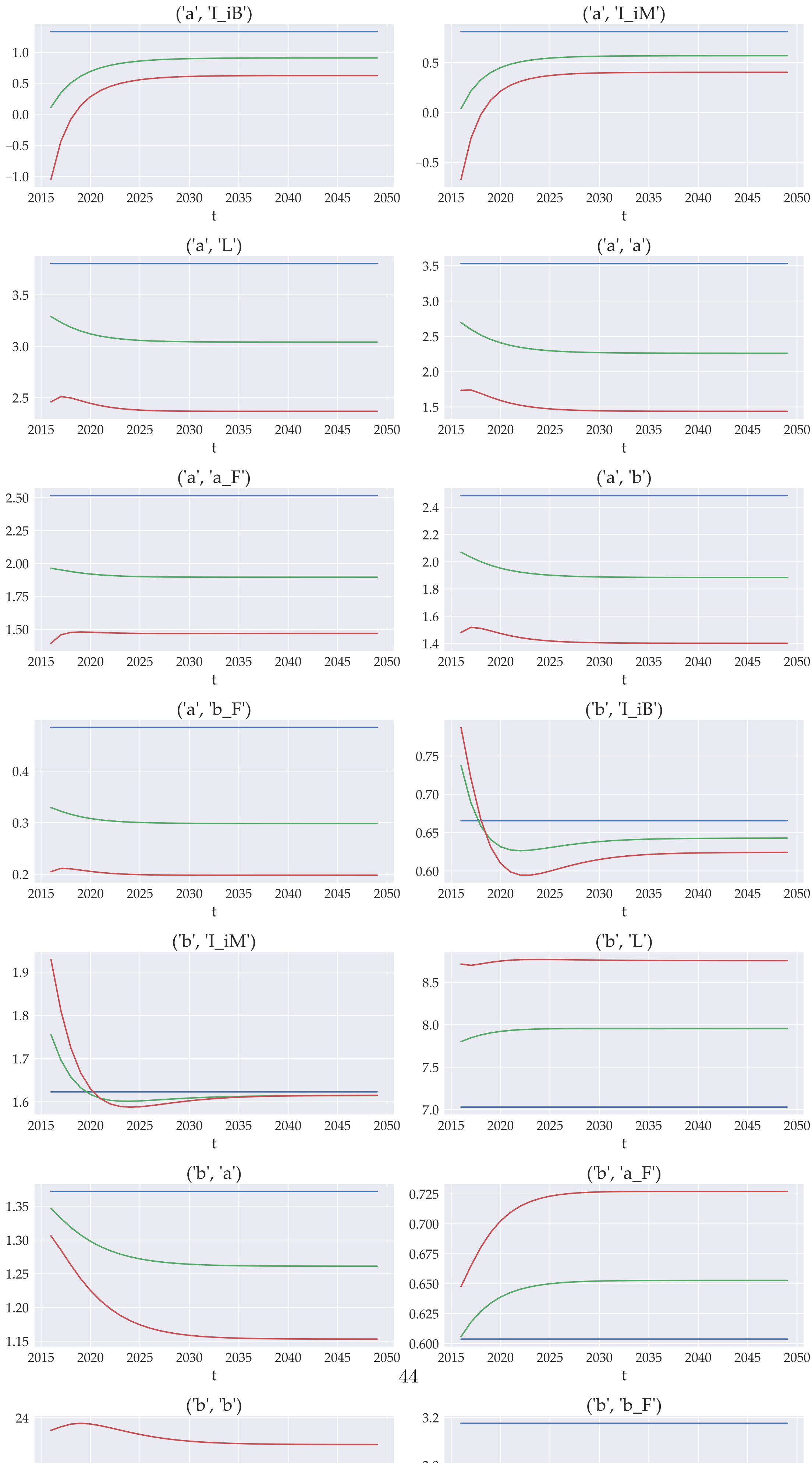


Figure F.3: Household sector's demand

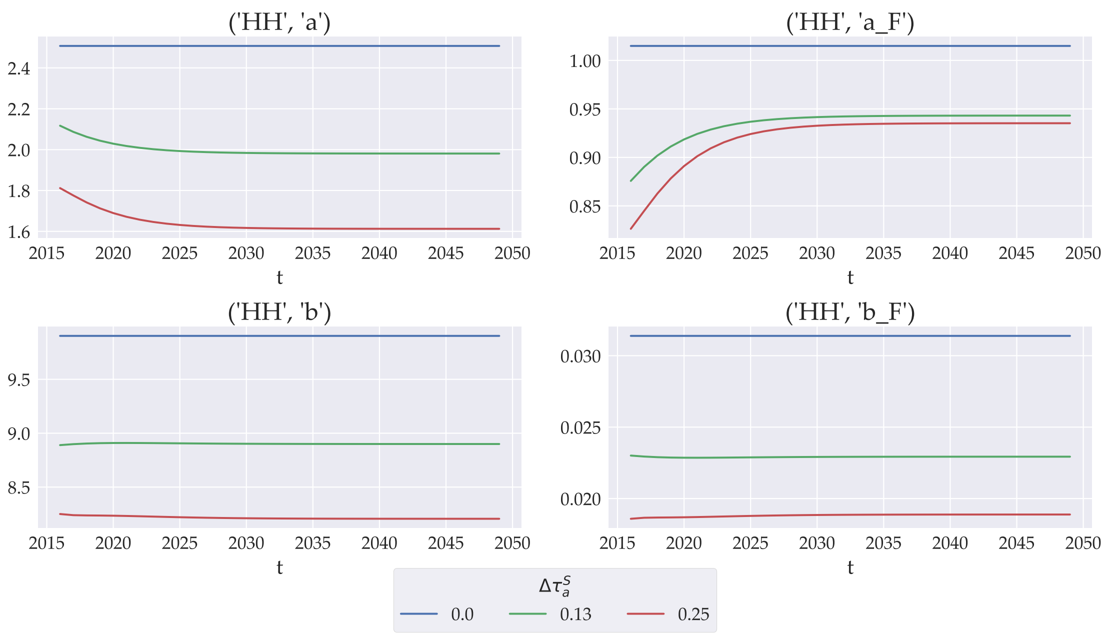
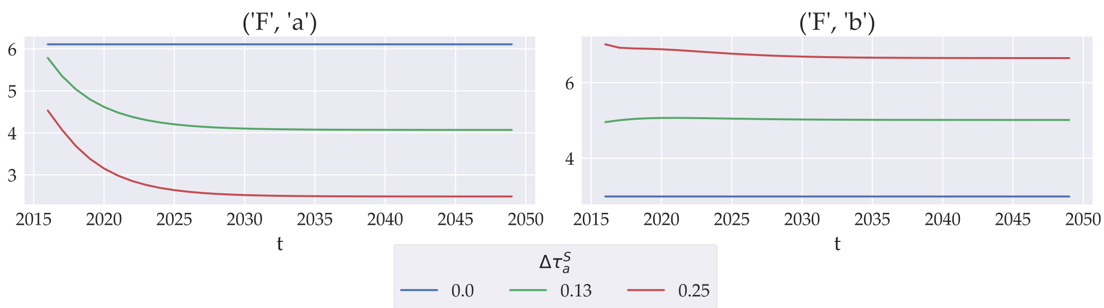


Figure F.4: Foreign sector's demand (export)



G A RAS-like method for balancing data

Consider the 2-dimensional data as outlined in table 1 where $v_{i,j}$ indicate cell values and R_i/ C_j indicate row/column sums.

Table 1: A representation of 2D data

	x	**y**	
a	$v_{a,x}$	$v_{a,y}$	R_a
b	$v_{b,x}$	$v_{b,y}$	R_b
c	$v_{c,x}$	$v_{c,y}$	R_c
	C_x	C_y	

We consider the case where some values in this table is fixed at zero for some reason (negative domains/small values). The following procedure attempts to cells in a way that keeps row/column sums intact and leaves the new data as true to the original as possible.

1. Fix data

We start by defining a couple of auxiliary variables:

- Let z^0 denote initial data values in general.
- Let $\overline{z}$ denote manual data adjustments to the data – i.e. fixing a value to zero $\overline{v}_{a,x} = 0$, and $\mathcal{D}$ denote the subset of (i,j) that are fixed.
- Given these adjustments, define the new step with a superscript 1:

$$v^1_{i,j} = \begin{cases} \overline{v}_{i,j}, & \text{if } (i,j) \in \mathcal{D} \\ v^0_{i,j}, & \text{else.} \end{cases}$$

- Define the column, row distributions $(\gamma_{i,j}, \omega_{i,j})$ as the share of the respective value:

$$\gamma_{i,j} = \frac{v^1_{i,j}}{\tilde{C}^1_j}, \qquad \tilde{C}^1_j \equiv C^1_j - \sum_{i' \in \mathcal{D}_j} \overline{v}_{i',j},$$
$$\omega_{i,j} = \frac{v^1_{i,j}}{\tilde{R}^1_i}, \qquad \tilde{R}^1_i \equiv R^1_i - \sum_{j' \in \mathcal{D}_i} \overline{v}_{i,j'}.$$

  This ensures that summing $(\gamma_{i,j}, \omega_{i,j})$ over rows/columns that are not fixed by data $(i,j) \notin \mathcal{D}$ sums to 1.

- Define the percentage change in row/column sums from the manual data adjustments:

$$\Delta R_i \equiv R^1_i - R^0_i, \qquad \Delta r_i \equiv \frac{\Delta R_i}{\tilde{R}^1_i}$$
$$\Delta C_j \equiv C^1_j - C^0_j, \qquad \Delta c_j \equiv \frac{\Delta C_j}{\tilde{C}^1_j}$$

2. Minimize distortions to data

To fix ideas, let's start by considering the case where we only had to achieve the same column sum as before the change. In this case, we would suggest using the following adjustment procedure for all $(i,j) \notin \mathcal{D}$:

$$\begin{aligned} v_{i,j} &= v^1_{i,j} - \gamma_{i,j}\Delta C_j \\ &= v^1_{i,j}\left(1 - \Delta c_j\right). \end{aligned}$$

The idea is that the row changes are mapped using the distributions $\gamma_{i,j}, \omega_{i,j}$ as weights. Consider the case in table 2 and assume that we want to fix $v_{a,x} = 0$. If we only cared about keeping the col-

Table 2: A representation of 2D data – Example 1

	x	y	
a	1	2	3
b	1/3	2/3	1
c	2/3	4/3	2
	2	4	

Step 0 – Initial data

$\Rightarrow$

	x	y	
a	-	2	2
b	1/3	2/3	1
c	2/3	4/3	2
	1	4	

Step 1 – Manual adj.

$\Rightarrow$

	x	y	
a	-	2	2
b	2/3	2/3	4/3
c	4/3	4/3	8/3
	2	4	

Step 2 – Column sol.

umn sums constant, our approach would suggest that we double both $v_{b,x}$ and $v_{c,x}$ – i.e. increasing the remaining values proportionally. As the tables above illustrate, however, this approach does not keep row sums intact. As a natural extension – if it was feasible – we would ideally extend the formula above to include row-sums:

$$v_{i,j} = v^1_{i,j}\left(1 - \Delta c_j - \Delta r_i\right).$$

This, however, does not (generally at least) fix the issue of keeping row/column sums fixed. Instead, we consider the more flexible quadratic program:

$$\min_{\{\eta^r_{i,j},\eta^c_{i,j},v_{i,j}\}_{(i,j)\notin\mathcal{D}}} \sum_{(i,j)\notin\mathcal{D}} \left[\left(\eta^r_{i,j} - 1\right)^2 + \left(\eta^c_{i,j} - 1\right)^2\right] \tag{14a}$$

$$\text{s.t. } v_{i,j} = v^0_{i,j}\left(1 - \eta^r_{i,j}\Delta r_i - \eta^c_{i,j}\Delta c_j\right), \qquad \forall (i,j) \notin \mathcal{D} \tag{14b}$$

$$C^0_j = \sum_{i\in\mathcal{D}_j} \overline{v}_{i,j} + \sum_{i\notin\mathcal{D}_j} v_{i,j}, \qquad \forall j \tag{14c}$$

$$R^0_i = \sum_{j\in\mathcal{D}_i} \overline{v}_{i,j} + \sum_{j\notin\mathcal{D}_i} v_{i,j}, \qquad \forall i \tag{14d}$$

$$v_{i,j} \geq 0, \qquad \forall (i,j) \notin \mathcal{D}, \tag{14e}$$

where the last inequality can be dropped if this is not essential for the type of data. Naturally, there are still bounds to how many manual adjustments we can impose on the data, but this can be solved relatively flexibly. In particular, note that – if this is feasible – this simply chooses the optimum $\eta^r_{i,j} = \eta^c_{i,j} = 1$.

3. Ensuring the problem is feasible

Let n denote the number of active values i.e. $\#(i, j) \notin \mathcal{D}$, n_c the number of column constraints, and n_r the number of row constraints. The quadratic problem can be reduced to identifying $2n$ variables $(\eta^r_{i,j}, \eta^c_{i,j})$ such that the rows/column sums hold. Thus, feasibility consists of $n_r + n_c$ linear constraints in $2n$ variables. Thus, for feasibility, we need at least one unique element $v_{i,j}$ in the active set for each i and for each j that is constrained.

Let us assume that the objective of this adjustment is to obtain a sparse, non-negative matrix. One way to identify a feasible active set is then:

i. Let $\boldsymbol{v}^{\text{o}}$ denote the initial data matrix. Identify the maximum of $v_{i,j}$ for each i - $\boldsymbol{v}^i_{max}$. Define $\tilde{\boldsymbol{v}}^{\text{o}} = \boldsymbol{v}^{\text{o}} \backslash \boldsymbol{v}^i_{max}$.

ii. Identify the maximum of $v_{i,j}$ for each j from $\tilde{v}$. Define $\tilde{\boldsymbol{v}}^1 = \tilde{v}^{\text{o}} \backslash \boldsymbol{v}^j_{max}$.

iii. Given that $(\boldsymbol{v}^i_{max}, \boldsymbol{v}^j_{max})$ are in the active set, we are certain of feasibility.

In this crude algorithm the order of row/columns, unfortunately, may make a difference in the final active set; this effect is, however, minor, as long as we do no not remove large values.