GreenREFORM-EU 2.0 Model Documentation

Theoretical Framework and Architecture

DREAM, Danish Research Institute for Economic Analysis and Modelling

November 2026

www.dreamgruppen.dk

Foreword

This document is the model documentation of GreenREFORM-EU 2.0. GreenREFORM-EU is a dynamic general equilibrium model for fiscal and environmental policy analysis, built and maintained by the DREAM group (Danish Research Institute for Economic Analysis and Modelling). It was developed for the finance ministries, planning bureaus and emissions agencies of EU Member States under the European Commission's Technical Support Instrument. It descends from GreenREFORM, the environmental-economic model used for climate policy analysis by the Danish government, and through it from MAKRO, DREAM's macroeconomic model for the Danish Ministry of Finance. The code is written in GAMS, supported by Python.

The model is a dynamic, multi-sectoral computable general equilibrium model of a small open economy, and it integrates the economic national accounts directly with physical energy balances and greenhouse gas emissions. Production is resolved across the industries of the underlying input-output data—57 in the Danish benchmark data set—through nested CES functions that combine non-energy materials, labour, and capital-energy aggregates. Capital is disaggregated into buildings, machines and transport, each paired with the energy inputs it requires, so that a change in the energy price acts on the capital stock that consumes that energy rather than on a single undifferentiated aggregate. Firms compete under monopolistic competition and set prices as a mark-up over unit costs. Households optimise intertemporally with perfect foresight over a Ramsey utility function with habit formation, alongside a cohort that is credit-constrained and myopic. Wages adjust along an empirical Phillips curve and export demand responds sluggishly, which is what gives the model realistic Keynesian multipliers over the short and medium term.

The distinctive feature is how the model treats technology. Decisions such as switching fuel, electrifying a process or fitting carbon capture are discrete in reality, and representing them normally means either leaving them out of the economic model or solving a separate engineering model and passing results back and forth. GreenREFORM-EU instead represents them continuously. Capital intensity is assumed to follow a log-normal distribution across the firms of an industry, so that a representative firm adopts competing technologies progressively along a merit-order curve as relative prices move. The resulting technology sub-model is hard-linked to the macroeconomic core and solves simultaneously with it, which is what allows a technology-specific policy to be scored for its effect on emissions and on output, employment, investment and the public finances at the same time.

The principle that makes this tractable is modularity. As much as possible, the text and code of this model are constructed in modules such that each one can: i) be understood and developed independently in partial equilibrium, ii) be combined with other modules to make up a larger partial equilibrium model which can also be studied independently, and iii) be put together with all other modules in order to study a complete general equilibrium model. To this end, both text and code are written so that a module can be changed as long as it preserves the key objects it

takes as inputs and the key objects it produces as outputs, so that the rest of the model runs without change to text or code. The goal is a general framework for models whether or not they contain a climate component.

How to read this document

The document follows the agents of the model and then closes with the markets that connect them. Chapter 1 sets out the architecture. Chapters 2 to 5 take the household, the firm, the government and the rest of the world in turn. Chapter 6 is where the pieces meet. Appendix A collects the CES and CET algebra that the earlier chapters use without deriving, and explains the indexing device that allows an entire nesting tree to be written as a single group of GAMS equations.

A reader who wants to follow one part of the model rather than the whole of it should read Chapter 1 first, then the chapter that documents it, and then the section of Chapter 6 that closes it. A reader who wants to work with the code should read Chapter 1 and Appendix A first, because between them they explain almost all of the notation, and Section 1.3 is also where the conventions of the text are set out: how code objects are named, which factors the equations leave out, and how the documentation files map onto the code.

One convention is worth stating here, because it applies from the first page. Where the text makes a claim about what the model does, a note gives the equation, set, dummy or switch in the code that supports it, and where a simplification hides something that matters, the note says so. This keeps the argument readable and still lets a reader verify it. The notes are endnotes: a superscript number refers to a numbered entry in the *Notes* list at the end of that chapter, and numbering restarts with each chapter. In the electronic version the number in the list links back to the point in the text where it was raised.

Status

This document describes the model as it stands under GreenREFORM-EU 2.0, and it is revised as the model is. The implementation it documents is the GAMS and Python codebase. Where the text and the code have been checked against each other and still disagree, the passage is marked, and the marked passages are listed separately rather than left for the reader to find.

Names follow the code. One is worth flagging, because DREAM's own project reports use the other: the sub-model that chooses between energy technologies is called the *abatement model* there, and the *energy technology model* here, which is the name the code uses. They are the same object.

Contents

Foreword **2**

1 Introduction **6**

1.1 Background, Motivation, and Purpose 6

1.2 The Core Technical Challenge 6

1.3 Code Structure and Modularity 7

1.4 The Model Economy 10

1.5 Structure of the Documentation and Code Files 13

2 Households **16**

2.1 Exogenous Savings and Income 17

2.2 Endogenous Savings (Ramsey) 19

2.3 The Consumption Tree 20

2.4 Energy Consumption 23

2.5 Housing 25

3 Firms (Private Production) **29**

3.1 Output, Cost and Price 30

3.2 The Production Function 31

3.3 Factor Demand 33

3.4 Capital Goods and Intermediate Inputs 34

3.5 Energy End Purposes 34

3.6 The Energy Technology Model 36

3.7 The Multi-Output Firm 43

3.8 Corporate Finance 46

4 Government and Public Finances **53**

4.1 Public Revenue 55

4.2 Public Expenditure and Transfers 58

4.3 Public Production 61

4.4 Net Flows to the Other Sectors 62

4.5 Fiscal Closure and Sustainability 63

5 Rest of the World (Foreign Trade) 68

5.1 Export Demand 70

5.2 Energy Exports 71

5.3 Import Demand 73

5.4 The External Balance and the Trade Elasticities 75

6 Markets and Equilibrium 78

6.1 The Input-Output System (Goods Market) 81

6.2 Price Setting and Inflation 88

6.3 The Labour Market 91

6.4 Energy Markets and Distributors 94

6.5 Emissions and Environmental Accounts 101

A CES Trees and Smart Indexing 111

A.1 CES algebra within a level 111

A.2 Multiple levels 113

A.3 GAMS code. Smart indexing. 114

A.4 The same algebra elsewhere in the model 117

A.5 Calibration and numerical notes 119

References 123

1 Introduction

This chapter sets out what a reader needs before any of the chapters that follow can be read: why the model exists, the technical problem that shapes its architecture, how the code is organised and named, the dimensions of the model economy in goods and in time, and the map from chapters to code files.

1.1 Background, Motivation, and Purpose

The conceptual foundation for the GreenREFORM model originated in Denmark to address the need for analytical tools capable of integrating climate and environmental considerations into the design of macroeconomic policy. Today, the motivation for such a framework has expanded into a pan-European imperative. The transition to climate neutrality by 2050 requires Member States to assess the macro-fiscal implications of green transition policies, green technology investments, and climate-related physical and transition risks. This necessity is formally institutionalised under the revised EU Economic Governance Framework, notably through amendments to Directive 2011/85/EU concerning national budgetary frameworks.

Meeting these legal requirements—and addressing European Semester Country-Specific Recommendations regarding sustainable public finances—demands that Ministries of Finance and other institutions move beyond static or fragmented modelling approaches. They require integrated, dynamic tools capable of evaluating real-world climate policies, such as carbon pricing, technology-specific subsidies, renewable energy mandates, and capital investments. These tools must be able to simultaneously capture economy-wide feedback on GDP, employment, wages, trade, and fiscal balance sheets.

To address this analytical gap, the European Commission funded the TSI 2024 project "Strengthening capacity to model the macroeconomic effects of 'green' policies and investments through the GreenREFORM model". This pilot initiated the deployment of the framework across five Member States: Belgium, Finland, Austria, Italy, and Poland.

Building on this, the Danish KR Foundation funded GreenREFORM-EU 2.0, an extension project to refine the model's capabilities, generalise its architecture for a broader European scale, and produce this documentation. The purpose of the GreenREFORM-EU model is to provide a standardised dynamic Computable General Equilibrium (CGE) framework. By equipping European policymakers with this tool, the extension project allows them to conduct climate risk assessments and green policy costings within a unified macro-fiscal environment.

1.2 The Core Technical Challenge

A conundrum arises when attempting to bridge environmental engineering and macroeconomics: existing sector-specific energy and climate models are typically

formulated as discrete optimisation problems, whereas macroeconomic CGE models operate within a differentiable space. In bottom-up energy system models, a central planner assigns demand for "energy services" to specific technologies through discrete choices to minimise total costs. Top-down CGE models usually depict technological change using constant substitution elasticities within a production function, a method that can overestimate or misrepresent responses to specific carbon price thresholds at the micro level. These two frameworks do not readily align.

The technical challenge for the development team was determining how to reformulate the discrete sector-specific models into a differentiable problem, thereby enabling integration with the macroeconomic model. The common alternative is *soft-linking*: the two models are solved separately and information is passed between them until the results agree. GreenREFORM-EU is instead *hard-linked*, which here means that the technology equations enter the same equation system as the macroeconomic ones and are solved simultaneously with them. Nothing is passed back and forth, because nothing is solved apart. The motivation is partly consistency, since an iterated solution is only as good as the convergence criterion, and partly computational efficiency.

This is the role of the *energy technology model*. Instead of a discrete central planner, individual technology-specific energy suppliers are depicted as small, decentralised units driven by individual cost-minimising behaviour. Because non-linear programming solvers cannot directly solve discrete choice problems, the model introduces a continuous approximation of the discrete indicator function. By smoothing this on/off mechanism with a probability distribution, all technology options can be described as part of the same non-linear equation system. This allows the granular, bottom-up energy data to solve simultaneously with the top-down macroeconomic core. Section 3.6 sets out the construction.

This integration is achieved within a dynamic CGE framework characterised by optimising agents and forward-looking behaviour. A dynamic recursive model employing a standard top-down approach would not capture the transitional impact of climate policy in a comparable manner, and a static model would be of limited use. By embedding the continuous approximation of discrete technologies into a model where agents engage in intertemporal optimisation, firms and households anticipate future carbon taxes and technology costs. This forward-looking mechanism ensures the model captures transitional dynamics, such as anticipatory investment in abatement capital before a carbon tax is fully implemented.

1.3 Code Structure and Modularity

The GreenREFORM-EU model is constructed using a modular design approach, inheriting and expanding upon the technical framework built for the Danish macroeconomic model MAKRO and carried forward by GreenREFORM, the environmental-economic model derived from it. This architecture is designed to manage the complexity of integrating a macroeconomic system with granular, sector-specific environmental data without sacrificing computational stability.

The platform and the preprocessor

The model's technical framework relies on the General Algebraic Modelling System (GAMS), a preferred platform for CGE models because of its non-linear programming solvers. To facilitate modularity, the model uses a preprocessor called gamY, which translates custom macros into standard GAMS code. The defining features of gamY are the `$Group`, `$SetGroup` and `$Block` statements, which enable batch operations on sets of variables (groups), of dummies (set groups), and of equations (blocks). Within this framework, a *module* is a distinct group of endogenous variables paired with its corresponding block of equations. A `$Block` header carries exactly three arguments: the name of the block of equations, the name of the group of associated endogenous variables, and a logical condition applied to all equations and endogenous variables in the block.

How equations and variables pair

When reading the underlying gamY code, it is instructive to understand how variables are determined and how the modules interact. gamY pairs each equation with exactly one endogenous variable. The variable can be named explicitly before the double dot `..` that opens the equation; when the name is omitted, the first variable on the left-hand side of the equation is used. Naming the variable explicitly is what allows an equilibrium condition to be assigned to a variable that does not appear in the equation itself: a market clearing equation can look like `p[t].. D[t] =E= S[t]`.[1] All other variables appearing in an equation are exogenous to it. Because the pairing is one-to-one, the full model always forms a square system. The same existence condition attached to a variable is applied automatically to the equation for which it is endogenous.

Combining and overriding modules

Integrating the various sub-models with the main CGE model is achieved through a flexible methodology. Combining modules generally involves adding the respective groups and blocks together and allowing the solver to process the combined system. When a specialised sub-model presents an alternative representation of a sector, as the energy technology model does for industrial energy demand, it can dynamically override the main CGE model. This is achieved either by removing specific equations from the main model using block statements, or by introducing scale parameters that prompt the main CGE model to recalibrate around the sub-model's outputs.

This modular architecture ensures that both the macroeconomic model and the detailed sub-models can coexist. It provides users with the flexibility to run the main CGE model independently for standard macroeconomic forecasting, or to hard-link specialised sub-models when conducting targeted climate policy analysis.

The phase sequence

Given any model selection, that is, any subset of modules to be run together, the code is run in a set of consecutive phases which is replicated within each code module. The phase sequence is: variable definitions, equation definitions, importing data and setting exogenous parameters, starting values, calibration of the remaining parameters using moment matching, and testing of the model. Set definitions are shared across modules and are read before the module phases, from the files in `model/sets/`. The upper code file `base_model.gms` selects the modules, and runs the resulting model by reading the respective fraction of code in each module in the same phase.

Naming conventions

Code objects appear in monospace throughout this document: variables such as `qCHh[cf,t]`, sets such as `cf_mapping`, and files such as `model/modules/03_production.gms`. File paths are relative to the root of the repository.

Nominal variables, that is price times quantity, start with the letter `v` for value; quantities start with `q`; prices start with `p`; and rates or ratios start with `r`. These first three prefixes also control the removal of trends, as described below. The remaining prefixes in use are `s` for shares and structural variables, `u` for calibrated scale parameters, written μ in this documentation, `e` for elasticities and other exponents, `t` for tax rates, `n` for numbers of persons, `f` for unspecified multiplicative factors, `j` for additive residual terms and `jf` for multiplicative ones, and `d1` for the dummies that control which index combinations exist. Groups shared across modules are prefixed `G_` and shared set groups `SG_`, while the equations generated by gamY carry the prefix `E_`.

Detrending

Most variables in the code are detrended, and the prefix decides which. Every variable that carries a time index is sorted by the first letters of its name into `price_variables`, `quantity_variables` or `value_variables`; the first group is then divided by $f^p = 1 + g^p$ per year, the second by $f^q = 1 + g^q$, and the third by $f^v = f^p f^q$, with $g^p = 0.02$ and $g^q = 0.01$ set in `settings.gms` and the exponent counted from the base year. A lagged nominal variable therefore enters an equation as V_{t-1}/f^v and a lagged quantity as q_{t-1}/f^q, and the solved model is stationary in these variables.

Everything else is left alone: variables with no time index, and the rates, shares, calibrated coefficients, elasticities, tax rates and person counts whose prefixes the sorting does not match. Physical units are not a criterion. Emissions in kilotonnes and energy in petajoules both carry the `q` prefix and are detrended like any other quantity, while the coefficients that tie them to activity, such as the emission intensities, are not.[2] Throughout this text these detrending factors are omitted, because they clutter

the appearance of every equation without changing its economics, but the reader should be aware of them when moving from the text to the code. The same applies to the calibrated scale parameters, elasticities and shares, which are written with Greek letters here and with their code names in the model.

1.4 The Model Economy

Two dimensions fix the size of the model: the partition of the data, which determines how many markets and prices are solved for, and the length and structure of the time axis, which determines what "forward-looking" means in practice. This section sets out both, and then summarises the decisions each agent makes.

The partition of the data

The finest partition of the economy is the industry index $i \in [i]$, which is loaded from `data.gdx` and contains 57 industries in the Danish benchmark data set that the model is currently calibrated to.[3] Every market clearing and aggregation identity in the input-output system is written over this index, and it is at this level that industry prices are formed; all other prices in the model are derived from them. From the industry level the model constructs objects at coarser levels of aggregation, and in particular at the National Income and Product Accounts level, where the macroeconomic aggregates that any model must calibrate to are found.

The industry index is one of several sets that the model reads from the data rather than declaring in the code. The demand components $[d]$, their subsets for private consumption $[c]$, government consumption $[g]$, capital $[k]$, exports $[x]$ and intermediate inputs, and the energy sets of energy goods $[e]$ and energy end purposes $[es]$, are all loaded from `data.gdx`. Everything that organises these elements—the nests of the demand and production systems, the named subsets, and the mappings between sets—is declared in `model/sets/`. The practical implication is worth stating once and holds throughout the document. How a demand or production tree branches—how many nests it has, and which goods compete with which—is written in the code and can be changed by editing it. Which goods sit at the bottom of the tree is settled by the data set the model is calibrated to.

A second and much coarser partition runs alongside the industry index. The financial side of the model is organised by the five sectors of the national accounts—financial corporations, non-financial corporations, general government, households, and the rest of the world—and for each of them the model carries net financial assets, the division of those assets into debt instruments and equity, and the interest, dividend and revaluation flows that move them. This is also what limits what the model can say about who holds what. Within a sector there is one balance sheet and one interest rate, so questions about distribution inside the household sector, or between large and small firms, are outside its reach. The rest of the world is the residual holder of equity, and interest flows are balanced through it as well, so that each of these accounts sums to zero across the five sectors; a test stops the run if one of them does

not.[4]

Time

The model is annual, and it is solved as one system over the whole horizon rather than period by period. This is a consequence of forward-looking behaviour rather than a numerical choice: the user cost of capital contains next period's investment good price and next period's marginal cost, and the household Euler equation contains next period's consumption, so this period's prices depend on next period's prices. It is also the reason a policy announced for a later year moves prices today.

The code separates two markers on the time axis that a reader of the present version could easily take for one. The *first model year*, `t1`, is the first year the solver treats as endogenous. The *last data year*, `tDataEnd`, is the last year for which the model is confronted with observation. They are kept apart deliberately, because the intention is that a later version will be calibrated against several years of data, and the two will then fall in different places. In the version documented here there is one data year, and `t1` and `tDataEnd` both sit at 2020, which is also the base year in which prices are normalised.

The solved model therefore runs from 2020 to 2050, thirty-one years.[5] The year set `t` begins in 2015 and the earlier years carry loaded data, but they lie outside the window the solver treats as endogenous. In the *projection years*, 2021 to 2050, the model runs forward under the policy and trend assumptions the user supplies. The *terminal year*, 2050, carries the conditions that pin down the forward-looking blocks.

This is also what lies behind a phrase that recurs in the chapters that follow: a parameter is *calibrated in the first model year* and then *held flat after the last data year*. The first half says the parameter is an unknown of the calibration in `t1`, backed out so that the model reproduces the observation for that year. The second half is a consequence of the way GAMS declares things. The parameter is defined over the whole of `t`, so it needs a value in every year, but beyond the last data year there is no observation left to calibrate it against. An equation therefore sets it equal to its last data-year value in each remaining year. It is a projection rule for calibrated parameters, not an economic assumption about the parameter itself, and a chapter that relies on it says so.[6]

The pattern is the same in every case: the equation that looks a year ahead is replaced in the terminal year by one that does not. The household Euler equation gives way to a requirement that detrended net financial assets are unchanged from the year before, so that the household stops accumulating at the end of the horizon. The user cost of capital, which elsewhere contains next year's investment good price and next year's marginal cost, is evaluated at this year's prices instead. The Phillips curve drops its term in expected wage inflation. Where the energy technology model is active and a technology outlives the horizon, its levelised cost is completed by assuming that costs stay at their terminal-year level.[7]

What is not pinned down matters as well. Nothing fixes the capital stock itself in

the terminal year, and nothing fixes the net financial assets of the corporate sectors; both follow from the accumulation identities and from what the terminal user cost implies. The condition that holds government net financial assets constant at the end of the horizon belongs to the calibration model and is not imposed when a policy experiment is solved.[8]

One further year matters for reading the equations. Prices are normalised to one in the base year `tBase`, which is the calibration year 2020, and objects such as the marginal utility of consumption are defined relative to that year so that they equal one in it by construction.

Agents and their decisions

Households choose how much to consume and how much to save, and how to compose the consumption bundle. In principle the household problem also covers portfolio allocation, a household housing stock, and labour supply or search, and modelling any of these requires either a utility function or a direct specification of the first order conditions. The version documented in Chapter 2 covers the savings decision and the consumption bundle; portfolio allocation, labour supply and a household housing stock are not part of it. The dwelling stock sits as building capital of the housing-services industry (Chapters 2 and 3).

Firms make optimal input choices, some of them static and some dynamic and forward-looking where they involve the accumulation of a stock. On the firm side technology has to be specified, and it is described by layers of CES functions, with an additional CET structure where a firm produces more than one output.[9] Energy is a particular kind of good and production input, with dedicated model specifications, but it is still traded in a market with firms and consumers like any other. Firms may be price takers or price setters; in the current version they are price setters with a constant markup.

Government enters the model as an accounting framework rather than as an optimising agent. There is no public sector objective function and no reaction function: every revenue item is a rate or a share applied to a base determined elsewhere in the model, and every expenditure item is either such a rate or an exogenous path. What the government block delivers is a complete and internally consistent public budget, the demand the public sector places on the input-output system, and a closure rule that keeps the government's net financial position from drifting. Several of these rules can nonetheless be read as reduced forms for decisions an elected government could take, and Chapter 4 makes that reading explicit, because it determines who bears the incidence of a shock.

Foreign demand in a small open economy model is typically an exogenous function of the difference between home and foreign prices. This summarises the modelling of the rest of the world and is consistent with the Armington trade model.

Population is exogenous, and it is the single most important exogenous input: it sets the scale of the economy and with it the scale of emissions and energy use.

There is no central bank in the current version, and therefore no monetary policy rule, no nominal exchange rate and no policy interest rate. Goods prices are not rigid—each industry sets its price as a markup over unit cost, and the markup is a variable—but nothing in the model pins down the nominal level of those prices. That is anchored instead by the wage, by exogenous world prices and by exogenous energy supply prices, with the interest rate held at a constant exogenous rate.[10] Section 6.2 sets out the consequences.

1.5 Structure of the Documentation and Code Files

The documentation is structured to align with the codebase. Because the model is built from self-contained modules, each module file carries a numeric prefix equal to the number of the chapter that documents it, so that `03_production.gms` is the production module and Chapter 3 is where to read about it, and Table 1.5.1 lists the modules chapter by chapter. Where a chapter documents only part of a module, or draws on a module owned by another chapter, the opening note of the chapter says exactly which files and which parts are involved; the table is deliberately kept at chapter level so that it does not have to be re-synchronised every time a section is renumbered.

Chapter	**Modules in** `model/modules/`
1 Introduction	`01_submodel_template.gms`
2 Households	`02_households.gms`, `02_ramsey_household.gms`, `02_consumption_disaggregated.gms`, `02_consumption_disaggregated_energy.gms`
3 Firms (Private Production)	`03_production.gms`, `03_production_CET.gms`, `03_production_CES_energydemand.gms`, `03_factor_demand.gms`, `03_factor_demand_energy.gms`, `03_pricing.gms`, `03_energy_technology.gms`, `03_financial_accounts.gms`
4 Government and Public Finances	`04_government.gms`, `04_energy_and_emissions_taxes.gms`
5 Rest of the World (Foreign Trade)	`05_exports.gms`, `05_exports_energy.gms`, `05_imports.gms`
6 Markets and Equilibrium	`06_input_output.gms`, `06_non_energy_markets.gms`, `06_labor_market.gms`, `06_energy_markets.gms`, `06_emissions.gms`

Table 1.5.1: Documentation chapters and the corresponding code files. The 23 files listed are the whole of `model/modules/`, and `base_model.gms` registers all of them, together with the reporting file `Report/All.Report.gms` that lies outside the directory. `01_submodel_template.gms` is registered and solved like the rest; it contributes one test equation and serves as the pattern for new modules. Appendix A documents no module of its own; it derives the algebra used by the modules of Chapters 2, 3 and 6.

Notes

1. The labour market is the clearest case in the code: `pW[t].. qL[t] =E= qProductivity[t] * nL[t]` in `model/modules/06_labor_market.gms` makes the wage the variable that clears the market, although the wage does not appear in the equation.

2. The sorting is in `model/variable_groups.gms`: a name beginning `p`, `jp`, `sp`, `Ep` or `mp` goes to `price_variables`, one beginning `q`, `jq`, `sq`, `Eq` or `mq` to `quantity_variables`, and one beginning `v`, `jv`, `sv`, `Ev`, `mv` or `nv` to `value_variables`. Variables without a time index are collected in `constant_variables` beforehand and are never adjusted. The group header notes that exceptions may be added by hand; none are in the current source. The factors are built in `model/growth_adjustments.gms` and applied by `base_model.gms` after the `exogenous_values` stage. For the emissions case see `model/modules/06_emissions.gms`, where `qEmmE_BU[em,es,e,d,t]` is declared in kilotonnes and the matching intensity `uEmmE_BU[em,es,e,d,t]` in kilotonnes per petajoule. Employment, `nL[t]`, is a head count and is not detrended.

3. The set is declared as `Set i[d<] "Production industries."` in `model/sets/input_output.sets.gms` and its elements are loaded from `data.gdx`. The number 57 is nowhere in the GAMS source; it is a property of the data set, and the aggregation mapping `iagg2i` in `model/sets/report.sets.gms` lists the 57 codes and groups them into five aggregates for reporting.

4. The set is declared in `model/sets/sectors.sets.gms` with the elements `FinCorp`, `NonFinCorp`, `Gov`, `Hh` and `RoW`. The balance sheet equations, the residual rules for `vNetEquity[RoW,t]` and `jrInterests_s[RoW,t]`, and the test all sit in `model/modules/03_financial_accounts.gms`. The test takes `vNetInterests`, `vNetDividends`, `vNetRevaluations`, `vNetDebtInstruments`, `vNetEquity` and `vNetFinAssets` in turn and aborts if any of them departs from zero by more than 10^{-6}. Note that the rest of the world's net financial assets are not themselves a residual: they accumulate from imports, exports and net government transfers abroad.

5. `model/settings.gms` sets `first_data_year` to 2015, `calibration_year` and `base_year` to 2020 and `terminal_year` to 2050. `model/sets/time.sets.gms` declares the year set `t` over 2015 to 2050 and fixes `tDataEnd` and `tBase` at 2020. The macro `set_time_periods(start, end)` moves `t1`, `t0`, `tEnd` and the endogenous window `t_endoyrs`. `base_model.gms` calls it with `(2015, 2050)` while variables are declared and data is loaded, but `calibration.gms` then calls it with `(2020, 2020)` for the static calibration and `(2020, 2050)` for the dynamic calibration, and nothing calls it again before `Solve main`. At the solve, `t1` is 2020, `t0` is 2019 and `tEnd` is 2050. That the calibration meets one year of data is enforced in `base_model.gms` by `$GROUP data_covered_variables data_covered_variables$(tDataEnd[t]);`. The shock files set the window themselves: `shock_model.gms` uses `(2020, 2050)` and `shock_CO2_tax.gms` uses `(2021, 2050)`, which holds 2020 fixed as the pre-shock year. These years belong to the Danish benchmark calibration and move with the data set.

6. The mechanism is the group `G_flat_after_last_data_year`, declared in `base_model.gms` and implemented in `calibration.gms` as `{name}{sets} =E= {name}{sets}{$}[<t>t1];` over `t1.val < t.val and t.val <= tEnd.val`. Modules add their own parameters to the group with `$GROUP+ G_flat_after_last_data_year`. A companion set group, `SG_flat_after_last_data_year`, does the same for dummies.

7. In `model/modules/02_ramsey_household.gms` the Euler equation carries the condition `$(not tEnd[t])` and is replaced in the terminal year by `vNetFinAssets['Hh',t] =E= vNetFinAssets['Hh',t-1]`. The accumulation identity divides the lagged term by f^{v} and this condition does not, which is what makes it a statement about the detrended level. The same split

appears for `pK_k_i` in `model/modules/03_factor_demand.gms` and for the Phillips curve, which is paired with `nL`, in `model/modules/06_labor_market.gms`. The levelised cost truncation is `uTKexp&_tEnd` in `model/modules/03_energy_technology.gms` and is active only when `d1switch_energy_technology` is on.

8. `model/modules/04_government.gms` places it in `government_calibration_equations`, which is added to the calibration model and not to `main`; the adjusting instrument is the lump-sum transfer `vLumpsum`.

9. Both CES and CET algebra are derived in Appendix A.

10. `model/modules/03_financial_accounts.gms` sets `rInterests` to a flat four per cent. The wage `pW[t]` is the variable paired with the Phillips curve equation in `model/modules/06_labor_market.gms`, the competing export price `pRoW_x[x,t]` is exogenous in `model/modules/05_exports.gms`, and the energy supply prices are taken out of `main_endogenous` when the `exogenous_supply_prices` switch is on, which is the default.

2 Households

Households choose total consumption versus savings, and the composition of the consumption bundle. Those are the only two margins they optimise over. Two quantities that a fuller household problem would also choose are present in the model but settled outside it. Labour supply is one: employment is determined in the labour market module by a Phillips curve around an exogenous path for structural employment, so the household does not trade off work against leisure. Portfolio allocation is the other: the composition of household wealth is fixed from outside and enters here only through a single average return on net financial assets.[1] The optimal choice problem can be solved separately in partial equilibrium. It takes prices as given and returns economy-wide totals that feed into the rest of the general equilibrium model.

When reading the code mapped to this chapter, it is helpful to view the four constituent modules as a single, combined "household block".[2] Doing so draws a boundary around the household sector of the model. The combined household module takes prices and macroeconomic income as given and returns the household's decisions on spending, saving, and product demand.

The following variables are determined inside the household block:

- **Total disposable income.** Compensation of employees plus interest and dividend income, minus net payments to the government.
- **Aggregate consumption and savings.** The budget allocated to current spending versus the amount saved for the future.
- **Disaggregated goods demand.** The quantities demanded for the twelve private consumption purposes of the set $[c]$ (five food categories, other goods, services, tourism, cars, housing and the two energy nodes) implied by the CES consumption tree.
- **Energy demand.** The quantities demanded for individual energy goods of the set $[e]$ inside the two energy nodes of the tree.

Household wealth is a household variable but it is not accumulated inside these four files. The stock of household net financial assets, and therefore the household budget constraint, sits in the financial accounts module `03_financial_accounts.gms`, which holds the net financial position of the five national accounts sectors of the set `sector`.[3] There is no household housing stock and no life-cycle structure, so there is no housing wealth on the household balance sheet; the dwelling stock is the building capital of the housing-services industry (Section 2.5).

The following variables are determined outside the block and act as the signals sent from the rest of the economy:

- **Prices.** Consumer prices for the twelve consumption purposes, determined by market clearing in the input-output module, and prices of individual energy

goods including taxes and distribution margins, determined in the energy market and energy tax modules.

- **Return on wealth and its taxation.** The return on household net financial assets is determined in the financial accounts module as the ratio of interest and dividend income to the lagged stock of assets; the marginal tax rate on that return is set in the government module.
- **Labour income.** Total compensation of employees, determined in the labour market module as the wage per efficiency unit times employment.
- **Government transfers and taxes.** Pensions, social benefits, contributions, income tax rates, VAT, energy duties and carbon taxes (determined in the government and energy tax modules).
- **Firm dividends and interest.** Distributed profits and net interest flowing to households as owners of private firms (determined in the financial accounts module).
- **Preference parameters.** The intertemporal elasticity, the habit weight, the marginal propensity to consume out of income, the discount rate, the survival rate, and the substitution elasticities of the consumption tree.

In the wider CGE model, the government, firm and labour market modules feed the exogenous signals (incomes, taxes, and prices) into the household block. The block processes them through the consumption function, or the Euler equation, and the consumption tree, and produces savings and the demand for specific goods and energy. These demands are then sent back to the market modules that clear the economy.

The indices used below come from two places. The demand components $[d]$, the private consumption purposes $[c] \subset [d]$, the industries $[i]$, the energy goods $[e]$ and the energy end purposes $[es]$ are all loaded from `data.gdx`. The nests of the consumption tree and every mapping between these sets are declared in `model/sets/households.sets.gms` and `model/sets/output.sets.gms`. The division of labour described in Section 1.4 applies here: how the tree branches—how many nests it has, and which goods compete with which—is written in the code and can be changed by editing it, but which goods sit at the bottom of the tree is settled by the data set the model is calibrated to.

2.1 Exogenous Savings and Income

This module, in `02_households.gms`, sets the accounting framework for the household sector: it aggregates the sources of household revenue—labour income, capital income, and net transfers from the government—and uses exogenous rules for total consumption and its composition, so that the module can be run before an Euler equation and a CES demand system are added. This rule-of-thumb description of

consumption keeps the module self-contained. It allows the household accounting to be run independently for troubleshooting, or dynamic interdependencies to be switched off in specific simulations.

There is a single representative household, with no heterogeneity across income or age groups. Nominal (aggregate) consumption is a fraction λ of total income plus a fraction ζ of assets.

$$p_t c_t = \lambda_t I_t + \zeta_t B_{t-1}$$

where B denotes the stock of household net financial assets and I is total household income (including interest earned on assets). The parameters (λ, ζ) are the reduced form summary of the endogenous optimal savings decision.[4] The marginal propensity to consume out of income is exogenous and set to 0.4; the propensity to consume out of wealth is calibrated in the first model year so that the consumption function reproduces observed aggregate consumption, and is then held flat after the last data year.[5]

Total income includes capital income, so there is no separate interest term in the consumption function:

$$I_t = w_t L_t - T_t + \Pi_t$$

Here $w_t L_t$ is total compensation of employees (`vWages[t]`, gross of tax, determined in the labour market module), T_t is net payments from households to the government (`vNetHh2Gov[t]`), and Π_t is net interest plus net dividends received by the household sector (`vNetInterests['Hh',t]` and `vNetDividends['Hh',t]`). The single object T_t carries the whole tax-and-transfer wedge: income tax on wages, tax on the return on wealth, capital taxes, contributions and accounting corrections enter with a positive sign, while transfers, the fiscal-closure lump sum and net acquisitions enter with a negative sign.

Assets have the standard law of motion—the household budget constraint—but it is stated in the financial accounts module rather than here:

$$B_t = B_{t-1} + I_t - p_t c_t + R_t$$

where R_t is the revaluation of the household portfolio (`vNetRevaluations['Hh',t]`), which is a valuation gain and therefore does not enter income. It should not be confused with interest earned. Portfolio choices are exogenous and are contained in the return $r_t \equiv \Pi_t / B_{t-1}$ which defines `rHh[t]` as the average return on household wealth. This can be implicit if that part of the model is not detailed and an exogenous average rate is used; in the current calibration the interest rate is set to four per cent for all sectors and revaluation rates are calibrated in the first model year.

The module also contains the placeholder for the composition of consumption. The expenditure share of each consumption purpose $c \in [c]$ is carried forward from the previous year up to an adjustment term,

$$\frac{v^D_{c,t}}{p_t c_t} = \frac{v^D_{c,t-1}}{p_{t-1} c_{t-1}} + j^D_{c,t}$$

with the real quantity $q^D_{c,t}$ as the endogenous variable. On its own, with $j^D_{c,t}$ exogenous and zero, this is a constant budget share model that can be calibrated to any data set. When the consumption tree of Section 2.3 is switched on, $j^D_{c,t}$ becomes endogenous and absorbs the difference between constant shares and CES demand.

Finally, the module carries a consistency variable. Aggregate consumption in the input-output module is the sum over $[c]$ of the twelve consumption purposes plus a residual `vC_WalrasLaw[t]`, and this module determines that residual from

$$p_t c_t = \lambda_t I_t + \zeta_t B_{t-1} + j^{C,ramsey}_t$$

so that the residual measures the failure of Walras' law in private consumption and should be zero. The term $j^{C,ramsey}_t$ is zero unless the module of the next section is active.

2.2 Endogenous Savings (Ramsey)

When the model is run dynamically, the Ramsey module (`02_ramsey_household.gms`) is added and takes over the determination of aggregate consumption. The household is now forward-looking: it maximises utility over the whole horizon and chooses how much to consume today against how much to save for the future. There are no cohorts, no initial wealth heterogeneity and no bequests, but there is a survival rate that acts as an additional discount factor, and there is habit formation and a direct income term in the utility argument.

The module does not delete the consumption function of the previous section. Instead it adds the link

$$c_t = c^{ramsey}_t \qquad t > t_1$$

with $j^{C,ramsey}_t$ as the endogenous variable. The residual $j^{C,ramsey}_t$ therefore absorbs whatever gap opens between the constant-propensity function and the optimal path, and aggregate consumption is effectively set by the Euler equation.

The core of the module is the Euler equation, which balances the marginal utility of present consumption against future consumption, adjusting for the after-tax return on wealth, consumption price changes, the survival rate and time preference:

$$\frac{U'(\hat{c}_t)}{p_t} = \frac{\sigma_t}{1+\rho_{t+1}} \left[1 + r_{t+1}\left(1-\tau^r_{t+1}\right)\right] \frac{U'(\hat{c}_{t+1})}{p_{t+1}}$$

where r is the return on household wealth `rHh[t]`, τ^r is the marginal tax rate on that return (`trHh[t]`, set to 0.25 in the government module), σ is the survival rate (`rSurvival[t]`, set to 0.993), ρ is the household discount rate (`rHhDiscount[t]`, initialised at $1.04/f^p - 1$ and calibrated in the calibration year).[6]

The budget constraint does not appear in this file either. It is the law of motion for household net financial assets stated in `03_financial_accounts.gms` and reproduced in the previous section.

The argument of the utility function is not consumption itself but consumption net of habits and of a Keynesian income term,

$$\hat{c}_t = c_t - h_t c_{t-1} - \lambda_t \frac{I_t}{p_t}$$

where h is the habit weight (`rCHabit[t]`) and λ is the same marginal propensity to consume out of income as in the previous section (`rMPC[t]`), which allows current consumption to respond to current income shocks. Marginal utility is defined normalised to the base year,

$$U'(\hat{c}_t) = \frac{\hat{c}_t^{-\eta}}{\hat{c}_{t_{Base}}^{-\eta}}$$

so that it equals one in the base year by construction. In the code this object is `qmuC[t]` and t_{Base} is the year in which prices are normalised to one, currently 2020.

A forward-looking savings decision needs a terminal condition. In the terminal year the Euler equation is switched off and replaced by the requirement that household net financial assets are unchanged from the year before,

$$B_{t_{End}} = B_{t_{End}-1}$$

so that the household stops accumulating at the end of the time horizon.[7] In the calibration the Euler equation is complemented by the requirement that Ramsey consumption equals observed consumption in the calibration year, with the discount rate of that year as the calibrated parameter.

2.3 The Consumption Tree

Once the aggregate consumption budget is determined, the module `02_consumption_disaggregated.gms` allocates it across categories of goods and services. It uses a nested Constant Elasticity of Substitution (CES) utility tree to model how households substitute between products when relative prices change.

At the top of the tree, households allocate spending between a housing bundle and a non-housing consumption bundle. The housing bundle splits into housing energy and housing itself; the non-housing bundle splits into a goods, tourism and services bundle and a car bundle, and the car bundle splits into car energy and cars. As the tree descends, the goods bundle disaggregates into food and other goods, food into five product groups, and services into services and tourism. At the bottom, each node is one of the twelve consumption purposes of the set $[c]$, and is sourced from specific industries by the input-output module. The demand quantities determined here feed into the goods market clearing conditions.

Total consumption is a CES aggregation of many different goods.[8] The tree is defined by the mapping `cf_mapping[cfNest,cf]` in `model/sets/households.sets.gms`. The index $[cf]$ is the union of the twelve bottom nodes, which are the elements of $[c]$, and the eight nests, which are the elements of

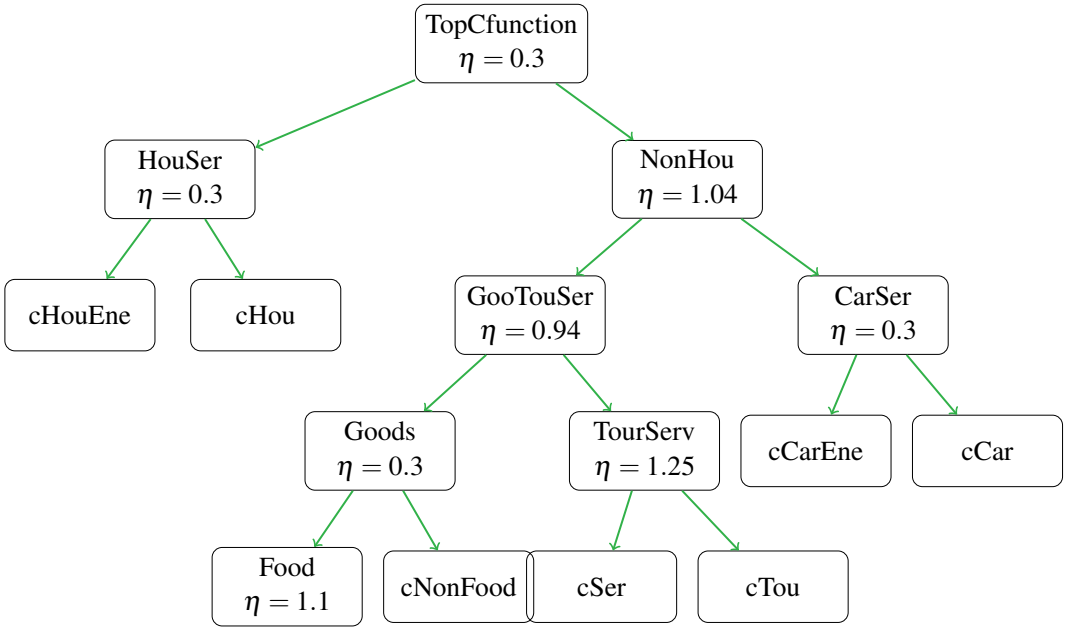


Figure 2.3.1: The household consumption tree. Nest names are the elements of `consumption_nests`; the leaves are elements of the set $[c]$ of consumption purposes. Each elasticity is the value of `eCHh` assigned to the nest.

`consumption_nests`.[9] The current structure, with the substitution elasticity of each nest, is shown in Figure 2.3.1.

The GAMS code is implemented by *smart indexing*, the device that lets an entire nesting tree be written as one group of equations rather than one group per level; it is set out in Appendix A.3. The equations themselves are the zero profit condition at every nest, $PQ = \sum_i p_i x_i$, and the demand curves $x_i = \mu_i Q (P/p_i)^{\eta}$. In the code they are written with an extra index which marks the tree node the equation refers to. Reading them takes some interpretation, and Appendix A.3 works through it.

There are two boundary conditions in the entire tree. At the top,

$$p_{top,t} q_{top,t} = p_t c_t$$

so that the value of the top node equals aggregate private consumption. At the bottom, the price of each node is the input-output price of the matching consumption purpose, $p_{c,t} = p^D_{c,t}$, and the value of the node returns to the input-output module as $v^D_{c,t} = p_{c,t} q_{c,t}$, which is the equation that endogenises the share adjustment $j^D_{c,t}$ of the previous section. The scale parameters—the μ_i (`uCHh`)—are calibrated in the first model year for every node except the top, and held flat after the last data year.

2.3.1 The tree under Food

There is additional detail here because different products are associated with different emissions. Food splits into five product groups, all of which are bottom nodes of the tree, as shown in Figure 2.3.2. A further split of meat into pork, beef and poultry is a

natural extension of this nest and is used in the Danish model, but it is not part of the current GreenREFORM-EU set definitions: `cFoodMeat` is a leaf.

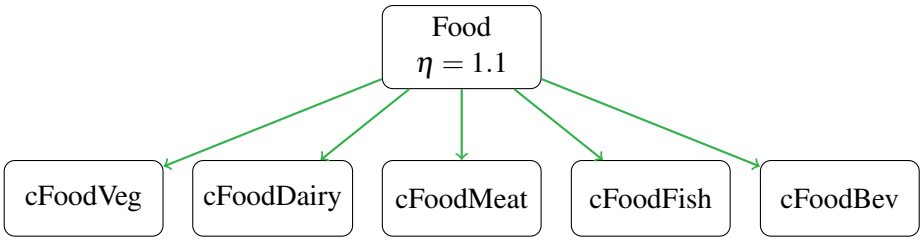


Figure 2.3.2: The food nest. All five children are bottom nodes of the consumption tree.

2.3.2 The bottom of the tree

There are many more production sectors indexed $[i]$ than there are bottom nodes in the consumption tree indexed $[c]$, and also in the general demand index $[d]$.[10] The reason is that each "consumption good" is an object sourced from different production sectors, with both domestic (y) and foreign (m) components. We can look at the household as purchasing directly from every industry, domestic and foreign. Or we can think of this as a number of zero profit intermediaries purchasing inputs from the different industries and assembling the consumption goods that households buy. These are equivalent views as long as we have a model with a single representative household. The moment we generalise the problem of the household to many different households, the intermediary becomes the more feasible way to model the problem.

This intermediation process is done through Leontief aggregation and is contained in a pair of equations in the `06_input_output.gms` code file.[11]

$$y^d_{i,t} = (1-\mu^{m,d}_{i,t})\mu^{y,d}_{i,t} D^d_t$$

$$m^d_{i,t} = \mu^{m,d}_{i,t}\mu^{y,d}_{i,t} D^d_t$$

where $0 \leq \mu_{i,t} \leq 1$ and

$$\sum_i \mu^{y,d}_{i,t} = 1$$

so that there are imported quantities in the demand objects consumed by households.[12] This is a Leontief construction where the total output generated is D and each input is determined by a ratio like $D = y_i/((1-\mu^m_i)\mu^y_i) = m_i/(\mu^m_i\mu^y_i)$. This implies the price of the good D is $p = \sum_i[(1-\mu^m_i)\mu^y_i p_i + \mu^m_i\mu^y_i p^f_i]$. In the code $\mu^{y,d}_{i,t}$ is `rYM[i,d,t]`, which is the industry composition of demand, and $\mu^{m,d}_{i,t}$ is `rM[i,d,t]`, which is the import share. Both are exogenous in the input-output module and are calibrated in the calibration year; submodels can endogenise them, and the energy market module does so for the energy demand components.

This applies to every bottom node except for energy.

2.4 Energy Consumption

The module in `02_consumption_disaggregated_energy.gms` is a sub-branch of the consumption tree, dedicated to the household's demand for energy. Because energy is rarely consumed for its own sake, but rather to run a vehicle or heat a home, the accounting of the underlying data distinguishes the energy end purpose, listed in $[es]$, from the energy good that serves it, listed in $[e]$.[13]

The equations in this module determine the demand for distinct energy goods based on their relative prices.[14] Household demand thus responds to energy and carbon taxation, though it uses the average rather than the marginal tax wedge.[15] Keeping energy in its own module allows the framework to interact with the bottom-up energy technology models used in policy simulations. Note that the energy technology model of `03_energy_technology.gms` is linked to the CGE model over industries only, so household energy demand is not routed through a household technology or durable stock.

The two energy nodes of the consumption tree, `cHouEne` in housing and `cCarEne` in transport, collect the demand for energy end purposes, which are themselves served by different energy goods. These energy goods, like any other goods, are produced by industries $[i]$. Getting from a consumption purpose down to a demand on named industries therefore takes three steps, from $[c]$ to $[es]$ to $[e]$ to $[i]$, with two intermediaries in between.

Which pairs (es,e) actually appear under a household energy node is not decided in this module. It follows from the dummy `d1pEpj[es,e,d,t]`, which is switched on when the energy balance gives that demand component a base price for the pair, when the pair is taxed, or when the energy comes from own production. The branch contents described below are therefore a property of the data set, not of the model structure.

In the case of **transport** the demand item is $cCarEne \in [c] \in [d]$. From this item there is a one-to-one (single element) mapping to the energy end purpose $transport \in [es]$. This intermediation step is therefore trivial (one to one), captured in the mapping `es2cf2d(es,cf,d)`.[16] This then branches into the energy goods that carry transport in the data. This is the second intermediation step, and it is a horizontal CES aggregation,

$$q^{es} = \left[\sum_{j \in S \in [e]} \left(q^{e}_{j}\right)^{(\eta-1)/\eta} \right]^{\eta/(\eta-1)}$$

where, in the Danish benchmark data set, the set S contains five elements of the set $[e]$: Liquid biofuels, Gasoline for transport, Diesel for transport, Natural gas incl. biongas, and Electricity. From here there is one remaining step between each of these energy goods (for example Gasoline for transport) and the industries associated with producing these goods.[17] The link from $[e]$ to $[i]$ only exists after aggregation, as discussed below. Before that we look at housing energy.

In housing the demand item is $cHouEne \in [c] \in [d]$. It has a one-to-two mapping to the energy end purposes $(heating, appliances) \in [es]$. This is the first (zero profit)

intermediation step, from $[c]$ to $[es]$. The second is from $[es]$ to $[e]$. Appliances has only one energy good $[e]$ associated with it, which is Electricity. Heating has several, as there are different ways (pellets, electricity) to heat a house.[18] So we should have the structure of Figure 2.4.1.

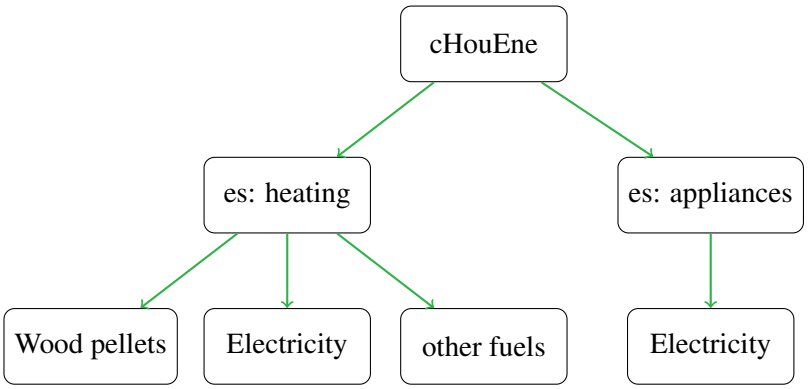


Figure 2.4.1: Housing energy as a two-level tree: consumption purpose, energy end purpose, energy good.

But instead the code implements Figure 2.4.2.

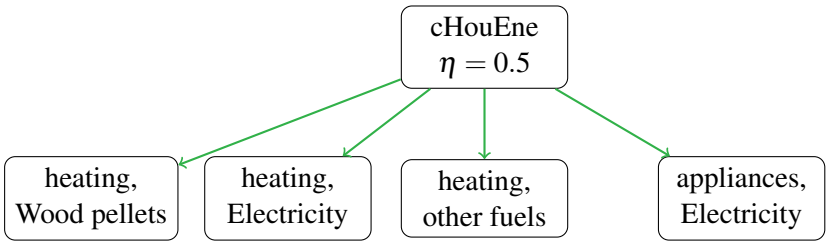


Figure 2.4.2: Housing energy as implemented: a flat CES over the (energy end purpose, energy good) pairs that exist in the data.

The energy end purpose level is bypassed, and the demand for each (es, e) pair depends directly on the quantity and price of the energy node of the consumption tree,

$$q^{Epj}_{es,e,c,t} = \mu^{Epj}_{es,e,c,t} \sum_{cf \ni (es,c)} \left(\frac{p^{Epj}_{es,e,c,t}}{p^{CHh}_{cf,t}} \right)^{-\eta^{ene}_{cf}} q^{CHh}_{cf,t}$$

so the aggregation is flat (single level) with a common elasticity for all pairs belonging to the same node. The elasticity η^{ene} (`eChh_cEne`) is 0.5 for both `cHouEne` and `cCarEne`, and the scale parameters μ^{Epj} (`uChh_Epj`) are calibrated in the first model year and held flat after the last data year. The demand variable in the code carries three indices, (es, e, c): the energy end purpose, the energy good, and the consumption purpose. The subset `cf_ene`, which contains the two energy nodes `cHouEne` and `cCarEne` of the household consumption tree, appears only as the summation index that selects the node being calculated, and for households it coincides with c.

Now we again reach the bottom link from $[e]$ to $[i]$. The link is in `06_energy_markets.gms`. Total demand (in the code q^e = `qEtot[e,t]`) for the energy good e

is computed by summing all demand elements (es, d) that carry a base price, plus the transmission losses of the demand subset `tl`,

$$q^e = \sum_d \sum_{es} q^{e,es,d}$$

This total demand is then allocated to different industries depending on a few cases. Case 1: when there is only one supplier of this energy good $e \in [e]$, either domestic or foreign, supply from that node $i \in [i]$ is set equal to demand. Case 2: exogenous domestic suppliers, with imports taking the slack (which means that imports could in principle go negative). Case 3: several endogenous suppliers, in which case the allocation of demand across producing industries and imports is done through a CES weights structure, with weights `sY_Dist` and `sM_Dist` calibrated in the calibration year and a distributor elasticity of five.[19] Whichever case applies, the resulting matrix of energy flows from industries to demand components has to balance: what each industry supplies of an energy good must equal what all demand components take of it.

2.5 Housing

Housing enters the model in two places, and it is worth separating them. The energy used in the home is the bottom node `cHouEne`, covered in the previous section. Everything else about housing is the other bottom node under the `HouSer` nest, `cHou`. This section is about `cHou`.

The short statement is that `cHou` is an ordinary consumption flow. It is a leaf of the CES tree like food or services, it is sourced from industries by the same fixed input-output coefficients as any other demand component, and it carries no special equation of its own.[20] The elasticity between housing and housing energy in the `HouSer` nest is 0.3, the same value as at the top of the tree, so a change in the price of home energy moves the household towards or away from the rest of the housing bundle only weakly.

The consequence is that, on the household side, housing is a purchase and not an asset. There is no household housing stock, no housing wealth in the household balance sheet and no housing investment by households. That is the ESA 2010 industry-account convention, not an omission of dwellings. National accounts record housing services as output of real-estate activity (NACE 68): actual rents from the letting of dwellings and imputed rents from owner-occupied dwellings. Households consume that output as the purpose `cHou`; the dwelling stock (AN.111) is the fixed capital used to produce it. In GREU that capital is the building stock `iB` of the housing-services industry `68203`, documented with the rest of firm capital in Chapter 3.[21]

There is no owner-occupier and no imputed rent as a separate object, so the model cannot distinguish owning from renting: both are inside `cHou` and inside `68203`. And because `cHou` is a flow and not a household user cost of a stock, its price is a consumer price, even though the price variable in the code is named `pCHh` and labelled a user cost.[22] Taxation follows the same line: the energy consumed in

the home is subject to energy duties, the carbon tax and the ETS prices through cHouEne, while cHou itself faces only the general duty and VAT structure that every demand component faces. There is no property tax, no land tax on dwellings and no housing-specific VAT rate.[23]

Nothing in the household modules is waiting to be switched on. Putting a dwelling stock on the household would be new module work, and it would leave the ESA industry layout rather than complete it. A national-accounts-consistent extension would keep cHou as the service flow and would split dwellings out of iB and/or split 68203 back into renting and owner-occupation. Until then, a policy that works through the cost of owning a home, rather than through the consumer price of cHou or the user cost of buildings in 68203, has no extra household channel. Results for the housing purpose are expenditure on housing services; the associated dwellings are the building stock of 68203.

Notes

1. Employment nL[t] is set by the Phillips curve in 06_labor_market.gms against structural employment snL[t], which is held at its first-model-year value by $(not t1[t]).. snL[t] =E= snL[t1];. The average return on household wealth is rHh[t], discussed in Section 2.1.

2. The four files are 02_households.gms, 02_ramsey_household.gms, 02_consumption_disaggregated.gms and 02_consumption_disaggregated_energy.gms. As explained in Chapter 1, the numerical prefix on a module file is the number of the chapter that documents it.

3. The elements are FinCorp, NonFinCorp, Gov, Hh and RoW. These positions must sum to zero. The module enforces this by making the rest of the world the residual sector, and its tests abort if the sum departs from zero.

4. In the code $p_t c_t$ is vC_CMP[t], λ is rMPC[t], ζ is rMPCW[t], I is vHhIncome[t] and B is vNetFinAssets['Hh',t].

5. rMPC.l[t] = 0.4; in 02_households.gms. rMPCW[t1] is swapped into calibration_endogenous against the equation vC_CMP[t] =E= vC[t] at t1, and the variable is registered in G_flat_after_last_data_year, which is the group that carries the equation holding it at its last data-year value in every later year. Section 1.4 explains that construction.

6. The (utility) coefficient of relative risk aversion η is (eCRRA, set to $1/0.8$).

7. The two equations are qC_ramsey[t]$(not tEnd[t]) and qC_ramsey&_tEnd[t]$(tEnd[t] and not t1[t]) in 02_ramsey_household.gms. This is a condition on the level of vNetFinAssets['Hh',t]: the accumulation identity divides the lagged stock by f^v and this condition does not, which is what makes it a statement about the detrended variable. Section 1.4 lists the other terminal conditions in the model.

8. CES algebra and tree structure are discussed in Appendix A.

9. The subsets that separate the two are cf_bottom for the leaves, cfNest for the nests and cf_top for TopCfunction. Since cf_bottom is defined as all of $[c]$ and every leaf must appear in cf_mapping

to receive a demand equation, the twelve leaves in Figure 2.3.1 and Figure 2.3.2 are the complete set $[c]$.

10. In the Danish benchmark data set there are 57 industries and 12 consumption purposes; the 8 consumption nests are a modelling choice and do not come from the data.

11. `qY_i_d[i,d,t] =E= (1-rM[i,d,t]) * rYM[i,d,t] * qD[d,t];` and `qM_i_d[i,d,t] =E= rM[i,d,t] * rYM[i,d,t] * qD[d,t];`

12. The composition of this mapping can be found in `data.gdx`, in the parameters `vIO_y` and `vIO_m`.

13. Only three elements of $[es]$ reach households: `heating`, `appliances` and `transport`. The set also holds the industry purposes `process_normal`, `process_special`, `in_ETS` and `unspecified`. The energy goods used below are named as in `these_e` in `model/sets/energy_outputs.sets.gms`, which enumerates the 25 energy products; note that the code spells bio natural gas *biongas*.

14. The relevant price is the average price of energy per petajoule, `pEpj`, which is built in `06_energy_markets.gms` and `04_energy_and_emissions_taxes.gms` from a base price plus VAT, energy duties, the national carbon tax, the ETS1 and ETS2 allowance prices, and retail and wholesale distribution margins.

15. The code flags this as a difference from the treatment of industries that should be revisited.

16. The intermediate index `cf` is redundant here. In `model/sets/households.sets.gms` the mapping reads `(heating,appliances).cHouEne.cHouEne` and `transport.cCarEne.cCarEne`, so the second and third indices always coincide for households.

17. In the benchmark data set the suppliers are the $[i]$ elements 19000 (refineries, named `i_refineries` in the code), 35002 (gas distribution, `i_gasdistribution`), 35011 and 38393. The supply side can be read from `qY_CET[out,i,t]` for domestic production and `qM_CET[out,i,t]` for imports.

18. In the Danish benchmark data set these elements of $[e]$ are Firewood and woodchips, Wood pellets, Other oil products, Diesel for transport, Gasoline for transport, Natural gas incl. biongas, District heat and Electricity, supplied by the $[i]$ elements 02000, 16000, 19000, 35002, 35011 and 38393. The fuel names are categories of the energy balance rather than statements about use, which is why transport fuels appear under heating.

19. The switch between the cases is the dummy `d1OneSX[e,t]`, which is currently set for every energy good except Straw for energy purposes, so Case 3 applies to straw only.

20. There is no equation in `02_consumption_disaggregated.gms` that treats `cHou` or `HouSer` differently from any other leaf or nest. The industry composition comes from `rYM[i,'cHou',t]` and the import share from `rM[i,'cHou',t]`, both exogenous in `06_input_output.gms` and calibrated in the calibration year, with the underlying cells in `data.gdx` in the parameters `vIO_y` and `vIO_m`, column `d=cHou`.

21. Danish national accounts keep two housing-service industries: `68203` (renting of residential buildings, HFCE 04100) and `68204` (owner-occupied dwellings, HFCE 04200). The January 2026 GreenREFORM industry file maps both IO117 activities (`680023` and `680024`) onto the single model industry `68203`, labelled as the housing sector including the household dwelling stock. GREU's set

[*i*] has 68203 and no 68204; the reporting aggregation in `model/sets/report.sets.gms` places it among the service industries. The national-accounts asset N111 (dwellings) is folded into buildings `iB` together with other buildings and structures in `data/preprocessing/read_data.py`.

22. `pCHh[cf,t]` carries the description "Usercost of private consumption of households" for *every* node of the tree, and at the bottom it is set equal to the input-output consumer price by `pCHh[cf_bottom,t] =E= sum(c$c2cf_bottom_mapping[c,cf_bottom], pD[c,t])`. The name is a leftover and does not signal a capital-service price. The genuine user cost of the dwellings that produce `cHou` is `pK_k_i['iB','68203',t]` in `03_factor_demand.gms`.

23. `cHouEne` is a member of `c_co2_tax[d]` in `model/sets/energy_taxes_and_emissions.sets.gms`; `cHou` is not. The household capital tax in `04_government.gms` is levied on financial assets, `vtCap[t] = tCap[t] * vNetFinAssets['Hh',t]`, not on a dwelling stock.

3 Firms (Private Production)

The production side of the model is the most detailed part of it, and most of that detail comes from the data rather than from the GAMS source. The sets and parameters that carry it are loaded from `data.gdx` and are used in, among others, `input_output.sets.gms` and `production.sets.gms`. The one to look at first is the set $[i]$ of industries, which has 57 elements in the Danish benchmark data set.

The modules listed below are best read as one block. Taken together they draw a boundary around private production: the block takes demand, base factor prices and tax rates as given, and returns output, prices, factor demands and the choice of energy technology.[1]

What is determined inside the block:

- **Total output and pricing.** The quantity of goods produced and the market price set by the firm, including industry-specific markups.
- **Factor demand.** The optimal quantities of labour, machinery, structures, vehicles and intermediate materials required to minimise production costs, together with the investment flows that build the capital stocks.
- **Energy and technology demand.** The specific mix of energy goods demanded based on the levelised cost of energy.
- **Corporate financials.** Earnings before interest, taxes, depreciation and amortisation (EBITDA), debt issuance, and dividend distributions to shareholders.

What the block takes from the rest of the model:

- **Aggregate demand.** The total market size and demand for the firm's products originating from households, government and exports, determined in the market clearing of the input-output system.
- **Base factor prices.** The base wage rate, determined in the labour market, and the hurdle rate the firm must clear to satisfy its investors.
- **Taxes and subsidies.** Production taxes and subsidies, carbon taxes on emissions, energy duties, and the allowance prices of the emission trading systems, all determined in the government and energy tax modules.

How this chapter is organised

The chapter follows the firm from the top down. Section 3.1 states what the firm is actually solving: output is handed to it by the market, and its problem is to deliver that output at least cost and to price it. Section 3.2 describes the nested CES tree that performs the cost minimisation, and Section 3.3 the user cost measures that sit at

the bottom of that tree. Section 3.4 explains how capital goods and materials are assembled out of the input-output system, and Section 3.5 reaches the energy leaves of the tree.

Section 3.6 then describes the energy technology model, which chooses between named technologies on cost grounds instead of substituting between energy goods along an elasticity. It is the longest section of the chapter, and its length is out of proportion to its reach: it changes what happens at the nodes where an energy end purpose is resolved into energy goods, and nothing else in the tree of Section 3.2. What it buys is the ability to bring engineering evidence into a CGE model and to analyse energy policy with it.

The last two sections deal with production structures that depart from the standard case. Section 3.7 covers the multi-output firm. Most industries map to a single output, but some, such as oil refineries, produce several products from one production process, and a Constant Elasticity of Transformation (CET) allocates capacity across them as relative prices move. Section 3.8 tracks corporate profits and the financing of investment, and connects to the financial accounts of `03_financial_accounts.gms`.

3.1 Output, Cost and Price

This section covers `03_production.gms` and `03_pricing.gms`. It sets out the chain that runs from the total cost of production to the output price, before the later sections describe how individual inputs are chosen.

The firm does not choose how much to produce. Output is handed to it by the market, and its problem is to deliver that output at least cost and then to price it. This is what makes the rest of the chapter a cost minimisation problem rather than a profit maximisation problem.[2]

Output sold and output produced are not the same. Installing new capital consumes resources, so the quantity leaving the top node of the tree, Q_i, equals output sold plus the installation costs $\Phi_{k,i}$ incurred on each of the capital types k,

$$Q_i = Y_i + \sum_k \Phi_{k,i}$$

The installation cost function is described in Section 3.3. The unit cost of output, p_i^0, then follows from an accounting identity: the value of output at unit cost equals the value of the top nest of the tree plus the production costs that are not part of the CES nesting,

$$p_i^0 Y_i = P_i Q_i + v_i^{other}$$

where P_i is the CES price index of the top nest.[3] The residual term v_i^{other} collects net production taxes and subsidies, less the bottom-up deductions on energy use, plus the energy costs that are handled outside the nesting tree (Section 3.5), less net government transfers to corporations.

Finally, the output price is a markup over unit cost,

$$p_i^y = (1 + m_i)\, p_i^0$$

This single equation is the whole of `03_pricing.gms`. The markup m_i (`rMarkup_i[i,t]`) is exogenous in the model: it is calibrated in the first model year so that the price identity reproduces the observed output price, and is then held flat after the last data year. The interpretation is monopolistic competition with a constant desired markup. There is no price adjustment cost and no sticky-price mechanism in the current version, so the markup does not respond to demand or to marginal cost.[4]

3.2 The Production Function

Here we describe the structure of the CES tree used to model the production function with the code contained in the module `03_production.gms`. The algebra of a CES nest and of a nesting tree is set out in Appendix A. The device that lets a whole tree be written as one group of equations rather than one group per level is *smart indexing*, and reading the code requires it; it is worked through in Appendix A.3. The tree is identical across private sectors, although parameter values will differ, and stacks up the different inputs as shown in Figure 3.2.1.[5]

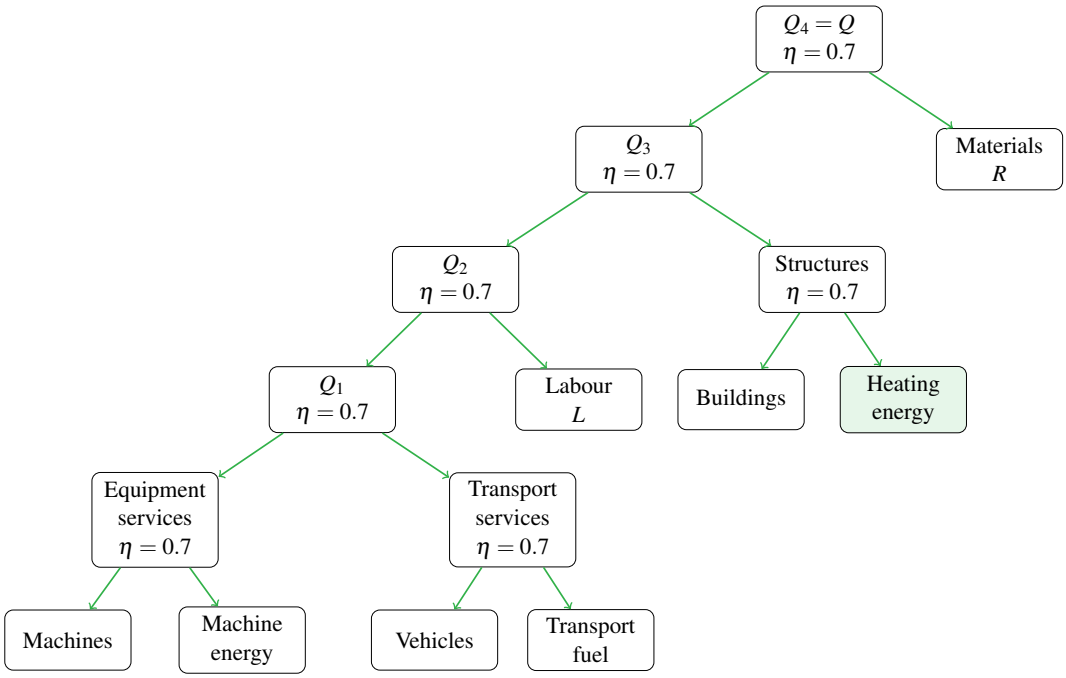


Figure 3.2.1: The production tree of a private industry. Every nest is a CES aggregator with the same substitution elasticity, $\eta = 0.7$. The code names of the nodes are listed in Table 3.2.1. The shaded Heating energy node is the one used as the example in Section 3.6.

Here Q stands for gross output, R for non-energy intermediate inputs (materials consumed in production), *structures* for the pairing of buildings and the heating

energy used in them, L for labour services, *equipment services* for the pairing of machines and the energy that runs them, and *transport services* for the pairing of vehicles and the fuel they burn. The nests Q_1, Q_2 and Q_3 have no separate economic interpretation; they exist to order the substitution possibilities. Buildings, machines and vehicles are stock variables, and they are the three capital types of the set $[k]$.

Two features of the figure are worth pausing on. First, the tree pairs each type of capital with the energy that this capital consumes, one level above the leaves. That pairing is what allows a carbon price to change the desired capital stock and not only the fuel mix. Second, the pairing is not decorative: the three capital-and-energy nests correspond exactly to the three capital types, and the mapping `es2k` in `model/sets/energy_technology.sets.gms` assigns every energy end purpose to one of them. Heating is assigned to buildings, transport to vehicles, and every other end purpose to machines. The energy branch of Section 3.5 and the technology model of Section 3.6 both hang off these three leaves.

All nests share a single substitution elasticity, set to 0.7 for every nest and every industry in `03_production.gms`. This is a placeholder rather than an estimate, and it is the natural first parameter to vary when investigating the sensitivity of a simulation.

Label in Figure 3.2.1	**Element of $[pf]$**	**Role**
$Q_4 = Q$	`TopPfunction`	top nest
Materials R	`RxE`	leaf, non-energy intermediates
Q_3	`KETELBE`	nest
Structures	`BE`	nest
Buildings	`iB`	leaf, capital stock
Heating energy	`heating_energy`	leaf, energy
Q_2	`KETEL`	nest
Labour L	`labor`	leaf
Q_1	`KETE`	nest
Equipment services	`KE`	nest
Machines	`iM`	leaf, capital stock
Machine energy	`machine_energy`	leaf, energy
Transport services	`TE`	nest
Vehicles	`iT`	leaf, capital stock
Transport fuel	`transport_energy`	leaf, energy

Table 3.2.1: The nodes of the production tree and their names in the code. The tree is declared by the mapping `pf_mapping[pfNest,pf,i]` in `model/sets/production.sets.gms`; the nests are the elements of `production_nests` and the leaves are the elements of `factors_of_production`, which is loaded from `data.gdx`.

3.3 Factor Demand

The CES structure aggregates all inputs into the final output object using a sequence of cost minimisation problems. At all levels except the bottom all prices are CES aggregation prices. At the bottom of the tree lie all the actual input prices, which generally are user cost measures derived from optimal decisions. This way of structuring the problem through a CES tree avoids calculating a complex expression for the marginal product of any given production factor. Instead it is implicit in the user cost measure. This can be seen in the optimal choice of a capital stock.

Capital stocks have law of motion $k_t = (1-\delta)k_{t-1} + I_t$, and the frictionless case has the user cost of capital given by the optimal condition

$$\beta\left[p_{t+1}F_k(k_t, l_{t+1}, \text{etc})\right] = p_t^k - \beta(1-\delta_k)p_{t+1}^k$$

where $\beta = 1/(1+r)$ is the firm's discount factor. Our CES tree avoids an explicit derivation of F_k. As we see in the `03_factor_demand.gms` code all we need is to define the user cost at time t, the time period when the decision is made,

$$p_t^{uck} = p_t^k - \beta\,(1-\delta_k)\,p_{t+1}^k$$

which then feeds into the CES tree. The discount rate inside β is a "hurdle rate" the firm must achieve to satisfy its investors. It is an object exogenous to the model, and it is set to 20 per cent for every industry.[6]

Capital adjustment is not frictionless. Installing capital consumes output according to the quadratic installation cost function

$$\Phi_{k,t} \;=\; \phi_k \frac{I_{k,t}^2}{k_{t-1}}$$

with ϕ_k fixed at 0.5 for every capital type and industry. Because a larger stock today both raises today's installation bill and lowers tomorrow's, two extra terms enter the user cost,

$$p_t^{uck} = p_t^k - \beta\,(1-\delta_k)\,p_{t+1}^k \;+\; P_t\,\frac{\partial \Phi_{k,t}}{\partial k_t} \;+\; \beta\,P_{t+1}\,\frac{\partial \Phi_{k,t+1}}{\partial k_t}$$

where P_t is the price of the top nest of the production tree, since it is output that is being consumed. The third term is positive and the fourth is negative: raising the stock today is expensive now but cheap later, and it is the balance of the two that slows rapid capital adjustment. Both derivatives are carried as their own variables in the code, `dInstCost2dK_k_i` and `dInstCost2dKLag_k_i`, and setting $\phi_k = 0$ recovers the frictionless user cost above.

The user cost of labour is simpler. It is the wage plus an additive wedge, $p_{i,t}^L = w_t + \omega_{i,t}$, and the wedge is zero, so the price of the labour leaf of the tree is the wage the labour market delivers.[7]

3.4 Capital Goods and Intermediate Inputs

Capital goods are stocks, subject to depreciation and accumulated through investment. This investment is then sourced from different sectors to create an investment object that then adds to the capital stock. Each investment, either in equipment or structures or transport, has its own tree. One characteristic of this level of the tree is that it can be interpreted either as part of the firm or as an intermediary object.

In the current model, it is easier to interpret it as a zero profit intermediary. The reason is that capital is assumed to be homogenous across industries. When different firms buy equipment to add to their capital stock, they all buy the same equipment capital good. This good is "produced" by putting together quantities of inputs purchased from potentially all different industries, domestic and foreign. The set $[k]$ lists the three types of capital, buildings, machines and vehicles, and $[i]$ indexes the industry. The k-intermediary uses a Leontief function, $I^k = min_{i\in[i]}(x_i^k/\alpha_i^k)$. We then have $p^k I^k = \sum_{[i]} x_i^k p_i$ and $x_i^k = \alpha_i^k I^k$ with $p^k = \sum_{[i]} \alpha_i^k p_i$, where some of the α_i^k could be zero.[8] Because every industry buys the same investment good, aggregate demand for investment of type k is a sum of quantities demanded across industries,[9]

$$D_t^k = \sum_{i\in[i]} I_{i,t}^k$$

where D_t^k is the aggregate quantity demanded and I_t^k the quantity supplied by the intermediary. They are the same object. It is only this total investment demand that is sourced from different sectors and from abroad, which is why there is no subtree under investment within each industry.

Materials are sourced the same way, from many sectors and possibly from abroad, so there is a tree under the node R as well, and it too can be read as a zero profit intermediary. Materials are not a stock, and there is a second difference from the investment case that matters when reading the code. Investment of type k is aggregated across industries *before* it is sourced, so there is one common capital good that everyone buys. Materials are not aggregated first. The composition of R depends on the buyer: the materials node of industry i is its own demand component, so its composition is the column of the input-output matrix belonging to industry i and it differs across industries.[10]

3.5 Energy End Purposes

Another feature that sets this production structure apart from other CGE models is the fact that under each energy leaf of Figure 3.2.1 there is a further subtree. This subtree resolves an energy leaf first into energy *end purposes* and then into energy *goods*. It is coded in `03_production_CES_energydemand.gms`, and its shape is shown in Figure 3.5.1.

The end purposes deserve a word before the figure, because the particular list below is not a modelling primitive and is not meant to be permanent. It comes from the

Danish energy and tax accounts, where these distinctions exist because energy duties and the carbon tax fall differently on heating, on transport and on process use, and because the boundary of the emission trading system cuts across all three. A different country brings different distinctions, and the model is built to take them: the set $[es]$ is loaded from `data.gdx` rather than written into the GAMS source, so adapting the model means supplying another list of end purposes and another mapping from those purposes to energy goods, not rewriting the production function.[11]

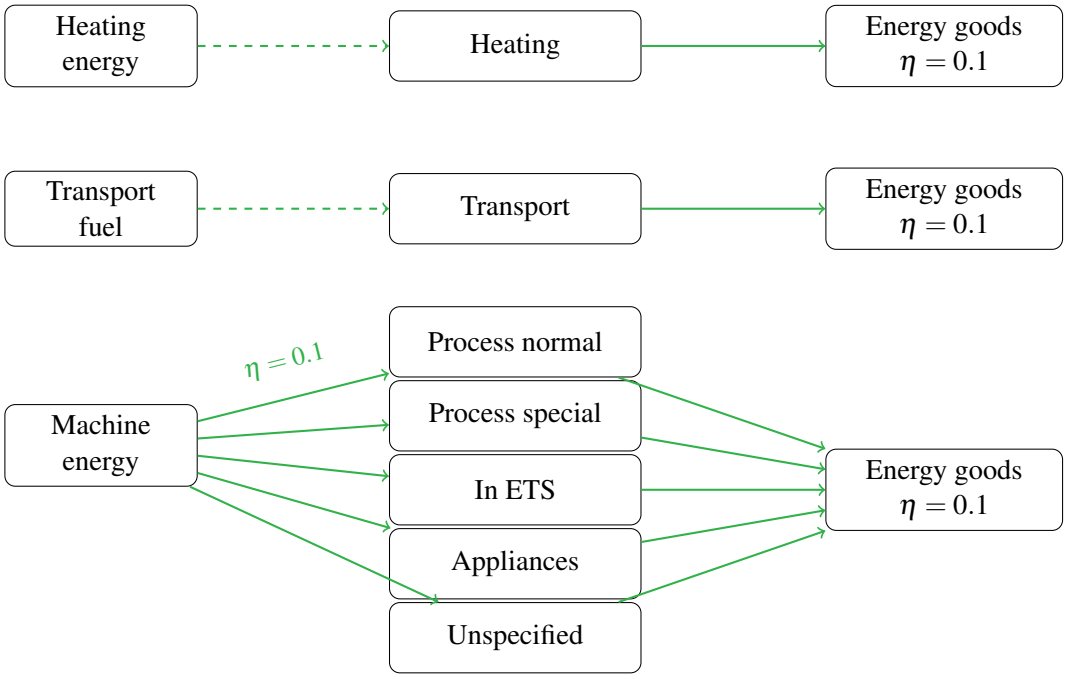


Figure 3.5.1: The energy branch below the three energy leaves of Figure 3.2.1. The middle column is the set of energy end purposes $[es]$; the right column is the set of energy goods $[e]$. Dashed arrows are identities, not aggregators. It is the last step, from an end purpose to energy goods, that the energy technology model of Section 3.6 replaces.

The heating and transport leaves are the simple cases. Each is identical to a single energy end purpose, so no aggregation happens and the leaf inherits the price of that end purpose directly. The machine energy leaf is a CES aggregate over all the remaining end purposes, with a substitution elasticity of 0.1. The three leaves are distinct all the way down the production tree, but they do not stay distinct once the tree meets the input-output table: at that point they are summed into a single intermediate energy input. So the firm chooses between heating, transport and machine energy, and substitutes within each of them, while the input-output table records only one energy column per industry. The split of that column across supplying industries is therefore common to the three leaves rather than specific to each.[12] Each end purpose is in turn a CES aggregate over energy goods, again with an elasticity of 0.1.[13]

Two details of the aggregation deserve mention. First, the price entering the CES demand functions is the *marginal* price of the energy good, that is the price including

the marginal tax rate before any deductions. This is why the deductions appear separately in v_i^{other} in Section 3.1 rather than in the tree. Second, the zero profit condition of an end purpose node is not quite the usual one: it includes a term for the cost of the machinery that delivers the service,

$$p^{es} q^{es} = \sum_e p^{e,marg} q^e + v^{\kappa,es}$$

The term $v^{\kappa,es}$ is zero when the energy technology model is switched off, and is the value of technology capital when it is on. It is the point at which the energy technology model of the next section attaches to the production tree.

A word on vocabulary before that section, because it calls this node something else. The set $[es]$ has two names in the source. It is declared as an *end purpose*, which is what it is in the data: a category of the energy and tax accounts. The energy technology model calls the same node an *energy service*, which is what it is in that model: the thing a technology is bought to deliver. There is one set, not two. This document says *end purpose* for the node of the tree, and keeps *energy service* in Section 3.6, where the algebra follows variables that are named after it.[14]

Not all energy use passes through the tree. Four uses are treated as Leontief in gross output at the top of the production function instead: crude oil in the refineries, biogas and extracted natural gas in gas distribution, and electricity in data centres. Their total cost is collected in `vEnergycostsnotinnesting[i,t]` and enters v_i^{other}. These are inputs whose quantity is fixed by a physical process or by external data rather than chosen by substitution.

This energy subtree eventually has bottom nodes consisting of energy goods. The production of energy goods requires modelling specificities detailed in Section 3.7, and the markets on which they are bought and sold are the subject of Section 6.4.

3.6 The Energy Technology Model

This is the longest section of the chapter and the one that carries the feature that distinguishes GreenREFORM-EU from a conventional CGE model. Where a CES nest lets a firm substitute between energy goods according to an elasticity, this section lets it choose between named technologies—a pellet burner against an electric heater, and between variants of each—on the basis of what they cost to buy and to run. That is what makes a carbon price act on the technology a firm installs rather than only on the quantity of fuel it burns, and it is what allows the model to speak to bottom-up engineering evidence about the green transition.

That evidence enters as what this chapter calls the *technology data*: a table with one row for each technology, in each end purpose, in each industry, in each year. A row carries five numbers—the share of the end purpose the technology could supply if all of its variants were running, how much of each energy good it consumes per unit it delivers, what it costs to build, what it costs to run, and how many years it lasts. All five are used below: the first is the capacity $\bar{q}_j$, the second gives the energy coefficients μ_j^e, and the last three are what the levelised capital requirement $\bar{\kappa}_j$ is

computed from. They are engineering quantities, measured in energy units and in currency per energy unit rather than as shares of a value flow, and that is what lets evidence from outside the national accounts into the model.[15]

Two points about its place in the model are worth making before the algebra, because the section is long enough to read as though it stood on its own. It does not.

First, it does not replace the CES nest below an end purpose. That nest stays where Section 3.5 put it, and the two structures are tied together by three links. Demand passes down: the CES quantity q^{es} fixes how much of the energy service the technology model has to deliver. Composition passes back up: the technology model's demand for each energy good is imposed on the CES nest, and the share parameters of that nest—ordinary calibrated constants when the technology model is off—become endogenous, solved so that the nest reproduces the fuel mix the technologies imply. Capital passes across: the machinery the technologies require is added to the firm's investment in the capital types of Section 3.3, and its cost is the term $v^{\kappa,es}$ that Section 3.5 left open. So the CES nest survives as the demand and accounting structure of the energy branch, while the technology model supplies the substitution that the elasticity used to supply. The two are solved as one system, in the sense set out in Section 1.2.[16]

Second, all of this can be switched off, and in the ordinary run sequence it starts off: the CGE model is calibrated first with the plain CES nests, and the technology model is brought in afterwards. The subsection "Turning the model on" below sets out that sequence.[17]

The firm buys inputs, including energy, to minimise total production costs.[18] Energy is supplied through a variety of technologies, specific to the energy service being provided. Consider a firm that needs to heat a factory building. Suppose this involves two technologies. One uses pellets and the other uses electricity (these are energy *goods*). Both technologies use machines to boil water to heat up the building. There are different pellet burners and electric heaters available. We call these *variants* of each technology j. The heated building is called a structure. We have the situation of Figure 3.6.1.

The problem is to choose which technologies and variants to use, and how much to demand from each. Along the decision tree and until the energy service nodes we use the standard nested CES construction. This results in a desired quantity of the energy service, heating in the example, which we label q^D. From the specific energy service node downwards, the choice of technologies used to satisfy q^D requires a more complex model than the standard CES aggregator. The energy service price p that matches q^D is not a standard CES price. We now explain how this is calculated. The nodes involved are the end purposes of Figure 3.5.1: what would otherwise be a CES aggregator over energy goods is not removed, but its shares are handed over to the technology data.

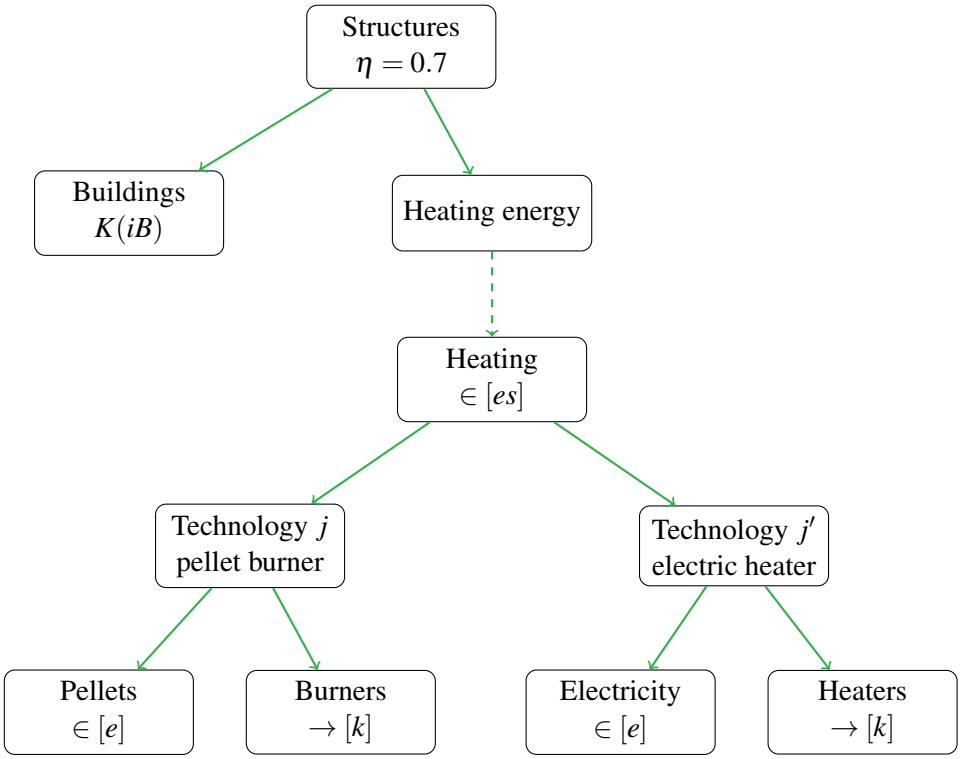


Figure 3.6.1: The energy technology model sits under a single energy end purpose. Above the *Heating* node everything is the standard nested CES construction of Figure 3.2.1. Below it, the choice of technologies is not a CES aggregation.

The energy service supply curve

We have a quantity demanded q^D and we will make use of a zero profit condition (pq^D = total cost), just like in the CES aggregation problem, and it is this zero profit condition that determines the price. What we now need to do is calculate the total cost of providing the energy service quantity q^D.

A technology variant is used if the cost c of doing so is lower than a threshold value c^*. The cost c is specific to each technology and each variant. As demand increases less efficient technologies are activated and satisfy demand up to their capacity $\bar{q}$.[19] Now, since there is a finite number of technologies and variants, the marginal cost curve jumps discretely as the next, less efficient, technology and variant is activated. However, discrete jumps create a computational problem which we solve with a smooth approximation. This smooth approximation assumes there are infinite variants of each technology, of which we observe only a finite number in the data.

We start by making the unit cost a function of capital *intensity*, $c(\kappa)$.[20] This κ is a draw from a distribution.[21] For technology type j and a given variant we use

$$c(\kappa) = p^{\kappa}\,\kappa + \sum_{e}\left(\mu_j^e\, p^e\right)$$

and it must be that $c(\kappa) < c^*$ for the variant to be used. At the margin $c(\kappa_j^*) = c^*$.

This unit cost function describes the cost of combining machinery capital and different energy goods to provide the energy service. From this function the marginal value κ^* depends on c^* as it satisfies

$$\kappa_j^* = \frac{c^* - \sum_e \left(\mu_j^e p^e\right)}{p^{\kappa}}$$

It is useful to add detail to the marginal relationship $c^* = p^{\kappa}\,\kappa_j^* + \sum_e (\mu_j^e\, p^e)$, which is the form in which the code writes it. Apart from the price of capital, all objects in this equation have a demand index d. So they are industry specific.[22] And all of them except p^{κ} have an energy service index es. The price p^{κ} is the price of buying machinery, not a user cost, and the capital it prices is a flow rather than a stock. The subsection "Capital in the technology model" below says why.

Now define a potential output (capacity), $\bar{q}_j$, for each technology j. This potential is reached if all variants are active. The realised *energy service supply* of a given technology j is then $q_j = \bar{q}_j \times F_j(\kappa_j^*)$ where F_j is the cumulative density function.[23]

Capital in the technology model

Capital is treated differently here than anywhere else in the model. The difference is worth setting out before the next subsection, which is where it comes from. In the CGE model of Section 3.3 capital is a stock: it accumulates, it depreciates, and investment is whatever is needed to carry the stock from one year to the next. The

firm pays for it through a user cost. In the technology model none of that happens. Capital enters as an annual flow, and it is priced at what the machine costs to buy.

The ideal would be the CGE treatment applied technology by technology: each technology, in each end purpose, in each industry, carrying its own stock, its own depreciation and its own vintages. The model does not do this. It levelises instead. The capital requirement $\bar{\kappa}_j$ of the next subsection is an *annuitised* figure: the cost of buying and running the technology over its whole life, spread evenly across the years of that life. Once the requirement is written that way the intertemporal problem has already been solved inside $\bar{\kappa}_j$, and what is left at the model's annual frequency is a pure flow. The machine is in effect bought new each year and used up within it, which is why p^{κ} is the price of investment goods rather than a user cost.[24]

What this buys is tractability; what it costs is history. There are no vintages, so the installed base carries no memory of what was built when, and a technology that becomes uneconomic is not stranded—it simply stops being bought. A change in the price of investment goods reaches the cost of every technology at once rather than only the cost of new installations.

That leaves a question of double counting, because the benchmark data contains one stock of machinery and the technology model now charges for some of the same machines. The model settles it by carving them out at calibration. In the first model year the CGE capital stock is set to its baseline value *minus* the value of the technology model's capital, so the machines the technology model pays for annually are taken out of the stock the CGE model accumulates. Each year's technology capital requirement is then added to investment demand without ever entering that stock: it generates demand for the output of the machinery industries, but it never depreciates.[25]

Two details of the bookkeeping are worth knowing when reading run output. The first is an asymmetry. The *price* of technology capital is always the price of machinery, but the *quantity* is added to whichever capital type the mapping `es2k` assigns to the end purpose, so the burners and boilers that serve heating are booked as investment in buildings.[26] The second is a pair of safeguards. The calibration requires that the value of the capital-and-energy nest of Figure 3.2.1 is unchanged when the technology model is switched on, and the tests abort if it is not; a further test aborts if the investment the technology model implies exceeds the investment recorded in the CGE data for the same capital type and industry. Both checks are in the `tests` stage of `03_energy_technology.gms`, and they run only if the optional tests are requested after the calibration solve.[27]

Where the distribution comes from: the levelised cost of energy

The mean of the distribution of κ for technology j is not a free parameter. It is the levelised capital requirement of that technology, and it is where the technology data enter the model. Let $\bar{\kappa}_j$ denote that mean, I_j the investment cost per unit of capacity, $C_{j,s}$ the variable capital cost in year s, n_j the life span of the technology in years, and

ρ the discount rate used for technologies. Then

$$\bar{\kappa}_{j,t} = \frac{I_{j,t} + \sum_{s=t}^{t+n_j-1} \left(\frac{1}{1+\rho}\right)^{s-t} C_{j,s}}{\sum_{s=t}^{t+n_j-1} \left(\frac{1}{1+\rho}\right)^{s-t}}$$

This is the levelised cost of energy of the technology, expressed in units of capital per unit of energy service delivered. The numerator is the present value of building and running the technology over its life; the denominator annuitises it. The discount rate is set to 5 per cent, and it is deliberately not the firm's hurdle rate of Section 3.3: it is the rate at which the technology data are levelised.[28]

The dispersion of κ around $\bar{\kappa}_j$ controls how smooth the approximation to the step function is. It is set proportional to the ratio of the technology's full unit cost at capacity to its capital requirement,

$$s_j = 0.05 \times \frac{\bar{c}_j}{\bar{\kappa}_j}, \qquad \bar{c}_j = p^{\kappa}\bar{\kappa}_j + \sum_e \mu_j^e p^e$$

so that technologies whose cost is dominated by fuel rather than by the machine are given a wider distribution of variants.[29]

Equilibrium

We have the technology specific supply quantities $q_j(c^*) = \bar{q}_j \times F_j(\kappa_j^*(c^*))$. We have the desired energy service q^D. Now define c^* as the margin where that demand is satisfied:

$$q^D = \sum_j q_j(c^*) \quad \Rightarrow \quad 1 = \frac{1}{q^D}\sum_j q_j(c^*) \quad \Rightarrow \quad 1 = \sum_j \tilde{q}_j(c^*)$$

This is one of the key equations in this part of the model as it determines c^*.[30]

The price of the energy service

This price is determined in the zero profit condition which we construct now. The energy service provided by a raw energy input e is given by

$$q_e = \sum_j \left(q_j \mu_j^e\right)$$

The average capital intensity of an active variant of technology j is the conditional expectation of κ when $\kappa < \kappa_j^*$. Label this $E_j^* = E_j\left[\kappa | \kappa \leq \kappa_j^*\right]$. The demand for energy machinery capital (burners, boilers, panels, batteries, etc.) is then given by

$$q_\kappa = \sum_j \left(q_j E_j^*\right)$$

Note that the weight here is realised supply q_j and not potential $\bar{q}_j$: only active variants own machines.[31] Finally we can write the total cost of this energy service

$$v^D = \sum_e q_e \, p^e \; + \; p^{uc\kappa} \, q_\kappa$$

and then, to get the price of the energy service p, use the zero profit condition $pq^D \equiv v^D$. These equations can be found in `03_energy_technology.gms`. Note that, while we have used many steps to calculate the price of the energy service, this is still an aggregator price similar to the CES prices we find elsewhere in the model.

Symbol	Code	Meaning
j	$[l]$	technology, up to 1000 slots
es	$[es]$	energy end purpose
d	$[d]$	demanding agent, an industry when linked
q^D	`qES[es,d,t]`	desired quantity of the energy service
$\bar{q}_j$	`sqTPotential[l,es,d,t]`	capacity, as a share of q^D
q_j	`sqT[l,es,d,t]`	realised supply, as a share of q^D
c^*	`pESmarg[es,d,t]`	marginal unit cost of the service
κ_j^*	`uTKmarg[l,es,d,t]`	capital intensity at the margin
$\bar{\kappa}_j$	`uTKexp[l,es,d,t]`	levelised capital requirement, the mean
s_j	`eP[l,es,d,t]`	log standard deviation of κ
μ_j^e	`uTE[l,es,e,d,t]`	energy input per unit of capacity
p^e	`pEpj_marg[es,e,d,t]`	marginal price of energy good e
$p^\kappa, p^{uc\kappa}$	`pTK[d,t]`	price of technology capital
q_e	`qESE[es,e,d,t]`	energy used to deliver the service
q_κ	`qESK[es,d,t]`	machinery capital used
v^D, p	`vES`, `pES`	value and price of the service

Table 3.6.1: Notation of this section and the corresponding names in `03_energy_technology.gms`.

Discussion

Our supply curve ranks technologies by efficiency levels. This ranking reproduces an environment where technologies are perfect substitutes but have capacity limits. The firm demands full capacity from the cheapest technology first, then from the second cheapest, and so on until fulfilling its needs. The original step-function supply curve is the model of this discrete choice available to the firm.

The smoothed supply curve takes the observed finite set of variants of each technology as realisations of a continuous process. When demand increases, c^* and κ_j^* increase, and so does the cumulative density of profitable variants for all technologies. Because all κ have the same support on the non negative real line, the supply function which contains all technologies is a continuous function with positive density everywhere in this domain. And, because it sums different j specific distributions on

the same support, it is a smooth approximation of the step function representing the discrete choice the firm faces. Efficient technologies have most of their density mass towards zero cost, and inefficient ones have most of it away from zero, but both have positive density everywhere.

Turning the model on

The technology model is not solved together with the CGE model on the first pass. In `base_model.gms` the switch `d1switch_energy_technology` is set to zero, so the CGE model is calibrated with the ordinary CES energy nests of Section 3.5. `base_model_energy_technology.gms` then saves that solution as a baseline, solves a partial energy price model to obtain the energy prices the technology data need, sets the dispersion parameters, solves the technology model on its own, and only then solves the integrated calibration. The link that closes the loop makes the CES share parameters `uEpj` endogenous, so the technology model dictates the energy mix and the CES shares adjust to reproduce it. A calibration residual absorbs any technology energy input that has no counterpart in the CGE data, and it is phased out over the five years following the first model year.

Where the technology model stops

The technology model ends at energy goods and machinery capital. Below that point nothing in this section applies: an energy good has to be produced or imported by some industry, and machinery has to be assembled out of the input-output system in the way Section 3.4 describes. That bottom layer is the same whichever agent is demanding, so it does not carry the index $[d]$, and it is not part of the firm's problem at all. Electricity, for instance, reaches a firm through a distributor that buys it from the industries that generate it; that market, and the price the firm ends up facing, are the subject of Section 6.4.

3.7 The Multi-Output Firm

In a typical problem the firm uses various inputs, like capital and labour, to produce quantities of one type of output. We would solve that problem by maximising profit through the choice of inputs where profits are $\pi = pF(x_1, x_2) - p_1x_1 - p_2x_2$ and where $F(\cdot)$ is a production technology, typically CES. When the firm produces several types of output from the same production process we need to change the way we structure the problem. This approach is known as a CET transformation and here we detail its derivation.[32] The code is contained in `03_production_CET.gms`.

The problem of the firm

Consider a firm that produces two types of output, (y_1, y_2) from the same production process. Production uses two different types of inputs, (x_1, x_2). Profits are given by

$$\pi = p_1^y y_1 + p_2^y y_2 - p_1^x x_1 - p_2^x x_2$$

Linking the revenue and the cost part in a way that allows the problem to be solved can be done in two ways. Either we transform the cost part into a (cost) function of (y_1, y_2), or we transform the revenue part into a (production) function of (x_1, x_2). Here we generate a cost function since that allows us to explore different demand functions for the different outputs.

The process takes two steps. First, we merge both inputs into one composite using a standard cost minimisation problem where total cost, $p^x x = p_1^x x_1 + p_2^x x_2$, is minimised subject to $x = CES(x_1, x_2)$ with $\alpha > 0$. This yields the demand functions x_j and price index p^x

$$x_j = \gamma_j \, x \, \left(p_j^x\right)^{-\alpha} \left(p^x\right)^{\alpha} \qquad \left(p^x\right)^{1-\alpha} = \sum_j \gamma_j \left(p_j^x\right)^{1-\alpha}$$

In the model this first step is the entire tree of Section 3.2, and the composite x is the top of that tree. Our profit expression now looks as follows

$$\pi = p_1^y y_1 + p_2^y y_2 - p^x x$$

The second step rewrites the cost function by linking the composite input x with the two outputs produced by the firm. This is done by postulating the relationship $y(y_1, y_2) = \phi \, x$ where y is again a CES function.

$$y(y_1, y_2) = \left(\sum_i (\mu_i)^{\frac{1}{\eta}} \, (y_i)^{\frac{\eta-1}{\eta}} \right)^{\frac{\eta}{\eta-1}}$$

with $\partial y / \partial y_i = (\mu_i \, y / y_i)^{1/\eta}$. One difference from an ordinary CES function matters, and the subsection "More on the function y" below returns to it: here $\eta < 0$.

Our profit expression now looks as follows

$$\pi = p_1^y y_1 + p_2^y y_2 - \frac{p^x}{\phi} y(y_1, y_2)$$

When the firm is a price taker the first order conditions are

$$p_i^y = \frac{p^x}{\phi} \frac{\partial y}{\partial y_i}$$

When the firm is not a price taker we treat the output price p_i^y as an isoelastic function, $p_i^y = Z_i y_i^{-\theta_i}$, $0 < \theta_i < 1$. The first order conditions are now

$$p_i^y + y_i \frac{\partial p_i^y}{\partial y_i} = \frac{p^x}{\phi} \frac{\partial y}{\partial y_i} \Longrightarrow p_i^y (1 - \theta_i) = \frac{p^x}{\phi} \left(\mu_i \frac{y}{y_i} \right)^{1/\eta}$$

where the right hand side is the **marginal cost** of y_i and $1/(1-\theta_i)$ is the markup. For this equation in the code inside `03_production.gms` we define $y =$`qYO_i[i,t]` and $p^x =$`pYO_i[i,t]`, which are the quantity and unit cost of Section 3.1. These objects then appear inside `03_production_CET.gms`. We do not see the parameter ϕ because it is non separable from the markup:

$$p_i^y = \left[\frac{1}{(1-\theta_i)}\frac{1}{\phi}\right] p^x \left(\mu_i \frac{y}{y_i}\right)^{1/\eta}$$

The code solves the same condition inverted, as a supply function for each output,

$$y_i = \mu_i \left(\frac{p_i^y(1-\theta_i)}{p^x}\right)^{-\eta} y$$

with $-\eta$ entered as a positive number, so that a rise in the relative price of one output raises the share of capacity devoted to it. The price appearing in the code is the marginal cost price $p_i^y(1-\theta_i)$, and the markup is applied to it in a separate equation.[33]

Closing the problem

In both cases we solve for (y_1, y_2) as functions of all prices and parameters, which then allows us to compute y followed by x and the final demand for inputs (x_1, x_2). In the first case the firm is a price taker so prices are the exogenous objects. In the second case the firm faces a downward-sloping demand curve, so the Z_i are the exogenous objects instead.

More on the function y

It is useful to look again at $y(y_1, y_2)$. First note that if $\eta < 0$ then $(\eta - 1)/\eta \equiv \delta > 1$. We can then write the function y without any negative parameters

$$y = \left(\sum_i \left(\frac{1}{\mu_i}\right)^{\delta-1} (y_i)^{\delta}\right)^{\frac{1}{\delta}}$$

where we note that all $\mu_i > 0$. The key point is that this function is **convex** in y_i

$$\frac{\partial y}{\partial y_i} = \left(\frac{\mu_i y}{y_i}\right)^{\frac{1}{\eta}} = \left(\frac{\mu_i y}{y_i}\right)^{1-\delta} = \left(\frac{y_i}{\mu_i y}\right)^{\delta-1}$$

since $\delta - 1 > 0$, and that is the reason we can view it as a cost function. Regarding the code the elasticity is defined as its negative, $-\eta =$`eCET[i]`$= 5$, inside `03_production_CET.gms`, uniformly across industries.

The bottom mapping

This production structure is implemented by a mapping from the input-output industry index $[i]$ to the collection of outputs indexed $[out]$. In most cases there is a one to one correspondence as a given industry will produce a single output, and the entire mathematical structure of this section does not apply. But in a few selected industries this mapping is non degenerate.

Two things need saying about how the code handles this distinction, because it is not handled with a switch. The equations are written over every pair (out, i) for which the dummy `d1pY_CET[out,i,t]` is true, and that dummy is set directly from the data: it is true wherever the loaded output price `pY_CET[out,i,t]` is non-zero.[34]

Therefore the CET equations are always present. Where an industry has a single output, the transformation degenerates: the supply function together with the zero profit condition force the output price to equal the industry's unit cost and the share parameter to equal one, and the elasticity has nothing to act on. Nothing needs to be switched off.

Which industries have more than one output is therefore a property of the data rather than of the code, and it can only be read off `data.gdx`. The reader can examine the object `pY_CET` there, where for example oil refineries ($i = 19000$) are associated with jet petroleum, refinery gas, diesel for transport, gasoline for transport, and with other oil products.

The structural reason why the non-degenerate cases are the energy producers is worth stating: the set $[out]$ contains both the energy goods $[e]$ and the ordinary industry products, and an energy producer typically delivers several distinct energy goods. Refineries are the clearest case, splitting one crude input across five refined products. It is not that every energy producer also carries a non-energy product; for refineries, gas distribution and 38393 the code zeroes the non-energy input-output rows outright, so their multiple outputs are all energy goods.[35]

One further point regarding the units of measurement. For energy outputs the quantities `qY_CET` and `qM_CET` are measured in petajoules, while non-energy outputs are measured in billions of 2019 currency units. The CET transformation - the $y(y_1, y_2)$ function - therefore mixes units within one industry, and the share parameters `uY_CET` absorb the conversion. There is also a distinction between gross and net production: `qY_CETgross` is what the transformation determines, and it splits into the part sold on the market, `qY_CET`, and the part consumed by the producing industry itself, `qY_CETown`, which is not recorded in the national accounts.

3.8 Corporate Finance

Operating closely with the overarching financial system, detailed in `03_financial_accounts.gms`, this section tracks the flow of corporate profits and the financing of capital investments. It details the modelling of corporate debt-to-equity ratios and dividend policies, and how corporate balance sheets connect to the rest of the

macroeconomy.

The link between production and the balance sheet is earnings before interest, taxes, depreciation and amortisation, formed for each industry as output value less the wage bill, less the cost of non-energy intermediates, less the cost of energy, less net production taxes and subsidies, plus net government transfers to corporations,

$$EBITDA_i \;=\; v_i^{y} - v_i^{w} - v_i^{R} - v_i^{E} - v_i^{tax} + v_i^{gov}$$

where v^R is non-energy intermediate consumption, v^E the energy bill and v^{tax} net production taxes and subsidies.[36] Note what is not subtracted: neither depreciation nor investment, which is the point of the measure. Industries are then grouped into the two corporate sectors of the national accounts. The financial corporation is the single industry 64000; every other private industry is a non-financial corporation; the public industry is `off`.[37] For each corporate sector, net financial assets accumulate as EBITDA plus net interest, dividends and revaluations, less capital and inventory investment.

Two behavioural rules close the corporate side. Debt is a fixed multiple of net equity,

$$D_t^{corp} \;=\; \lambda_t \, E_t^{corp}$$

with the ratio λ calibrated in the first model year and held flat thereafter, separately for the financial and non-financial sectors. And corporations pay out any excess cash as dividends: the dividend rate ρ_t^{div} is the variable that makes the payout absorb EBITDA net of investment and interest and net of any change in debt outstanding. Dividends received by each sector are then that common rate applied to the equity each sector holds. The clearing equation is written on the non-financial sector alone; the financial sector's cash flow does not enter it, but it receives dividends at the same rate on the equity it holds.[38]

It is important to be clear about what this does *not* include, because the distinction matters for interpreting simulations. There is no costly external finance in the current version: the debt-to-equity ratio is exogenous, so the firm's cost of funds does not depend on its balance sheet, and the hurdle rate of Section 3.3 is exogenous as well. The value of the firm is not computed, so revaluations are a rate applied to lagged assets, calibrated in the first model year and carried forward, rather than the consequence of an endogenous asset price. Together these mean that there is no financial accelerator in the model. The code says so explicitly, and the structure is in place should it be added.[39] A test in the same module aborts the run if net interests, dividends, revaluations, debt, equity or net financial assets fail to sum to zero across all five national accounts sectors. The rest of the world is the residual holder of equity and absorbs the residual in the interest account, which is what makes those two sums close.

Notes

1. The files are `03_production.gms`, `03_pricing.gms`, `03_production_CET.gms`, `03_factor_demand.gms`, `03_factor_demand_energy.gms`, `03_production_CES_energydemand.gms`

and, for the energy technology model, `03_energy_technology.gms`. As explained in Chapter 1, the numerical prefix on a module file is the number of the chapter that documents it.

2. The code carries a scale factor between the top of the production tree and Y_i, `qPFtop2qY[i,t]`. It is calibrated so that the price of the top nest is one in the base year, and it becomes endogenous when the multi-output module of Section 3.7 is active. We suppress it here.

3. In the code p_i^0 is given by `pY0_i[i,t]`, Q_i by `qProd['TopPfunction',i,t]`, P_i by `pProd['TopPfunction',i,t]` and v_i^{other} by `vProdOtherProductionCosts[i,t]`.

4. The one place where a markup is derived from a demand elasticity rather than calibrated is the multi-output problem of Section 3.7, and even there it is not separately identified.

5. Earlier versions excluded refineries (19000) and gas distribution (35002) from the tree. Those exclusions are commented out in `model/sets/production.sets.gms`, where the subset `i_standard` equals all of $[i]$, so the tree is at present the same for every industry. The special treatment of energy supply happens in the CET split of Section 3.7, and not in this tree.

6. `rHurdleRate_i[i,t]` in `03_factor_demand.gms`. Note that this is the rate the firm discounts at, and it is far above the interest rate of the financial accounts, which is 4 per cent. The depreciation rates are 4 per cent for buildings and 15 per cent for machines and vehicles.

7. `pL_i[i,t] =E= pW[t] + pLWedge_i[i,t]` in `06_labor_market.gms`, entering the tree through `pProd[labor,i,t] =E= pL_i[i,t]` in `03_production.gms`. The wedge is industry-specific in its dimensions, which is what it exists for, but it carries no value in this version. Chapter 6 documents the wage itself.

8. In the code we use only the demand function $x_i^k = \alpha_i^k I^k$ and the constraint $p^k I^k = \sum_{[i]} x_i^k p_i$. The coefficients α_i^k are the input-output shares `rYM[i,d]` of `06_input_output.gms` evaluated at the demand component $d = k$, split between domestic and imported deliveries by `rM[i,d]`.

9. Equation in `03_factor_demand.gms`. The k set is loaded from `data.gdx` in `input_output.sets.gms`, and its three elements are `iB`, `iM` and `iT`. The national-accounts assets behind those three elements are not one-for-one. Buildings `iB` receive dwellings (N111) as well as other buildings and structures (N1121, N1122_3), from `fixed_assets_dict` in `data/preprocessing/read_data.py`. For most industries that mix is non-residential. For the housing-services industry 68203 the building stock is the dwelling stock that produces the housing services households buy as `cHou` (Section 2.5). Danish accounts split that activity into 68203 (renting of residential buildings) and 68204 (owner-occupied dwellings); GREU follows the Danish GreenREFORM aggregation into the single industry 68203.

10. In `03_factor_demand.gms` the link is `qD[i,t] = qR2qY_i[i,t] * qY_i[i,t]`, where the demand component index is the industry index itself. This works because $[i]$ is declared as a subset of the demand components $[d]$ in `input_output.sets.gms`. The materials-to-output ratio `qR2qY_i` is calibrated in the first model year and held flat after the last data year, so materials are Leontief in output.

11. `model/sets/output.sets.gms` declares `set es "End-purpose of ergy";` and fills it with `$load out, e, es, eBunkering` from `data.gdx`. The pairs (es, e) that actually carry demand are governed by dummies rather than by a hard-coded list, as described below. What the code does fix is a handful of element *names*: `output.sets.gms` declares `transport`, `heating`, `process_normal`, `process_special` and `in_ETS` as named singleton subsets of $[es]$ and refers to

them by name, and `households.sets.gms` does the same for `appliances`. An adaptation either keeps those names or changes the references with the data.

12. The leaves `machine_energy`, `transport_energy` and `heating_energy` are elements of `pf_bottom_e` in `model/sets/production.sets.gms`, nested in `KE`, `TE` and `BE` by `pf_mapping`, with their own CES links in `03_production_CES_energydemand.gms`. The set of intermediate energy inputs `re`, by contrast, holds the single element `energy`: the data builder sets `re_records=['energy']` in `data/preprocessing/read_data.py` and `data/data_from_GR.gms` sums the leaves, `qE_re_i['energy',i,t] = qProd['machine_energy',i,t] + qProd['transport_energy',i,t] + qProd['heating_energy',i,t]`. The collapse happens in the mappings `es2re` in `model/sets/output.sets.gms` and `pf_bottom_e2re` in `model/sets/production.sets.gms`, both of which send every leaf to `energy`. Three-element versions of both mappings sit commented out immediately above, marked `#£Temp until IO-split`, so a per-leaf input-output split is intended but not implemented. There is no switch: restoring it means editing the data builder and both set files.

13. `eREes` and `eEpj` in `03_production_CES_energydemand.gms`, both set to 0.1. The relevant variables are `qREmachine[i,t]` for the machine energy aggregate, `qREes[es,i,t]` for an end purpose, and `qEpj[es,e,i,t]` for an energy good used for a given end purpose. Which pairs (es, e) actually appear is governed by the dummies set in `04_energy_and_emissions_taxes.gms`.

14. `model/sets/output.sets.gms` declares `set es "End-purpose of ergy";` and `model/sets/energy_technology.sets.gms` only aliases it, so there is one set. But `03_production_CES_energydemand.gms` documents `qREes` as "CES-Quantity of energy-services", and `03_energy_technology.gms` declares `qES` as "Energy service, quantity." The two words are used interchangeably in the code.

15. The data are read from `data/Energy_technology_data/Excel_data/Energy_technology_data.xlsx` and converted to GDX by `Import_energy_technology_data.py` in the same directory. The five are `sqTPotential`, `uTE`, `uTI`, `uTC` and `LifeSpan`, indexed by technology $[l]$, end purpose $[es]$, industry and year. The workbook also supplies a capital cost index and the energy service quantity `qES`. The switch `generic_energy_technology_data` in `settings.gms` chooses between this data and a generic set held in `data/Energy_technology_data/Generic_dummy_data`.

16. The link equations are the block `energy_technology_equations_links` in `03_energy_technology.gms`. Demand down is `qES[es,i,t] =E= jES[es,i,t]*qREes[es,i,t]`, with `jES` exogenous. Composition up is `qESE[es,e,i,t] + jqESE[es,e,i,t] =E= qEpj[es,e,i,t]`, an equation named after `uEpj`: in shocks `jqESE` is exogenous and the CES share parameter `uEpj` is endogenous. Capital across is `qI_k_i_energy_tech` and `vI_k_i_energy_tech`, mapped to capital types by `es2k` in `energy_technology.sets.gms`, alongside `pTK[i,t] =E= pD['iM',t]*jpTK[i,t]`. The residual `jqESE` is not dropped at once: it is phased out linearly over the first five model years, so the two structures agree exactly only from $t1+5$.

17. There are two switches, and the link equations above require both: `d1switch_energy_technology` turns the technology model on, and `d1switch_integrate_energy_technology` connects it to the CGE model. The shock files set them as a pair. The reader should also know that this part of the model goes by two names. The source document calls it the *abatement model*; the code and this documentation call it the *energy technology model*. They are the same thing.

18. For detailed algebra on this section read "Documentation of the Abatement Model of the Green Reform EU Model" by Louis Stewart, Peter Stephensen, and Jens Sand Kirk, June 2025. The code is `model/modules/03_energy_technology.gms`, with its sets in `model/sets/energy_`

`technology.sets.gms`.

19. The ranking of technologies by efficiency has a specific meaning which we discuss below.

20. κ is not a quantity. It is a measure per unit of output and a technological property of the machine.

21. κ is not a random variable. It is an ex-post *realisation* of a log normal random variable. The density function is used for aggregation. There is no randomness in the model.

22. The variables of the technology model are declared over the full set of demand components $[d]$, which would in principle admit the household energy nodes. In practice they are not covered: the technology data are supplied for industries only, both in the Excel workbook, where the energy service quantity is `qES[es,i,t]`, and in the generic alternative, where `these_i_es` lists industry codes. The link equations are likewise written over industries, `qES[es,i,t]` and `jqESE[es,e,i,t]`. Household energy demand is handled entirely by `02_consumption_disaggregated_energy.gms` through `cHouEne` and `cCarEne`, with no connection to the technology model.

23. In the code $q_j \equiv$`sqT[l,es,d,t]` and $\bar{q}_j \equiv$`sqTPotential[l,es,d,t]`, with the extra (es,d,t) indices, and with the technology indexed by the set $[l]$ rather than by j. Both are measured as *ratios* of the energy service quantity q^D rather than in levels, which is why the equilibrium condition below reads $1 = \sum_j \tilde{q}_j$. On the distribution itself: each technology variant follows $\log(\kappa) \sim N(m_j, s_j)$, and all κ have support $]0, +\infty[$. The code parameterises the distribution by its *mean* rather than by m_j, which is the subject of the next subsection.

24. In the code $p^{\kappa} \equiv$`pTK[d,t]`, set by `pTK[i,t] =E= pD['iM',t]*jpTK[i,t]` in the links block, where `pD['iM',t]` is the purchase price of machinery. The declaration of `pTK` calls it a "user cost of capital in technologies for energy services", and a comment beside the equation refers to `pK_k_i['iM',i,t]`. Both are leftovers: the equation uses the purchase price, and `jpTK` is endogenous during calibration, so it is calibrated against the same price the equation applies it to.

25. The carve-out is the calibration equation named after `qK_k_i` in `03_energy_technology.gms`, which sets `pK_k_i*qK_k_i =E= pK_k_i_baseline*qK_k_i_baseline - vI_k_i_energy_tech` at `t1`, with `jI_k_i` taken out of the calibration endogenous group and `qK_k_i` at `t0` put in to absorb it. The flow itself is `qI_k_i_energy_tech`, which `03_factor_demand.gms` adds to `qI_k_i` outside the accumulation term, so it never reaches `qK_k_i`.

26. `es2k` in `model/sets/energy_technology.sets.gms` maps `heating . iB`, `transport . iT`, and the remaining end purposes to `iM`.

27. The three `ABORT` statements are in `energy_technology.gms`; `production_CES_energydemand.gms` contains none. They are invoked from `base_model_energy_technology.gms`, alongside the zero-shock test.

28. $\bar{\kappa}_j$ is `uTKexp[l,es,d,t]`, I_j is `uTI`, C_j is `uTC`, n_j is `LifeSpan` and ρ is `DiscountRate`, all in `03_energy_technology.gms`. A second version of the equation applies in the last years of the horizon, where the sums run past *tEnd* and costs are held constant thereafter. The technology data are read either from `data.gdx` or from `data/Energy_technology_data/Excel_data/Energy_technology_data.gdx`, according to the switch `generic_energy_technology_data` in `settings.gms`.

29. `eP[l,es,d,t]` and `smooth_factor` in `03_energy_technology.gms`. The dispersion is

recalculated in `base_model_energy_technology.gms` after a partial energy price model has been solved, because some of the energy prices needed for $\bar{c}_j$ do not exist in the CGE model data. There is no estimate behind the value 0.05; it is chosen to make the supply curve smooth enough to solve.

30. In the code this is the equation named after `pESmarg[es,d,t]`, which is the variable it determines: $c^* \equiv$`pESmarg`. The quantity of the energy service q^D is `qES[es,d,t]`, and it is taken from the CES quantity `qREes[es,i,t]` of Section 3.5 up to a calibrated share parameter.

31. The code writes the same thing as $q_\kappa = \sum_j \bar{q}_j P_j(\kappa_j^*)$, where $P_j(\kappa^*) = \int_0^{\kappa^*} \kappa dF_j(\kappa)$ is the *partial* expectation. The two are identical because $P_j = F_j \cdot E_j^*$ and $q_j = \bar{q}_j F_j$. The GAMS functions are `@cdfLogNorm` and `@PartExpLogNorm` in `model/functions.gms`.

32. Details in "Strategy for integrating the energy sector model and the CGE model in Green Reform" by Jens Sand Kirk, Peter Stephensen, and Rasmus Berg (May 26, 2020), appendix B.

33. In `03_production_CET.gms` the equation reads `qY_CETgross[out,i,t] = uY_CET[out,i,t] * (pY0_CET[out,i,t]/pY0_i[i,t])**eCET[i] * qY0_i[i,t]` and is closed by `pY0_i[i,t]*qY0_i[i,t] = sum(out, pY0_CET[out,i,t]*qY_CETgross[out,i,t])`, the zero profit condition of the transformation. The markup is applied afterwards, in `pY_CET[out,i,t] = pY0_CET[out,i,t]*(1+rMarkup_out_i[out,i,t])`. The markup is an endogenous variable, but it is pinned to a calibrated constant by `rMarkup_out_i[out,i,t] =E= rMarkup_out_i_calib[i,t1]`, so every output of a given industry carries the same markup and it does not move over time. An alternative link from the CET block to the pricing module, which would let the industry markup absorb the value of CET output, sits commented out in `03_production_CET.gms`.

34. The assignment is in the market clearing part of `06_energy_markets.gms`. A companion dummy `d1pM_CET[out,i,t]` does the same job for imports.

35. The zeroing is in `06_input_output.gms` and `data/data_from_GR.gms`. The industries that produce both energy and non-energy output are flagged by the dummy `d1Y_i_nepnei[i,t]` in `06_input_output.gms`. The candidates are the industries of the `power_and_utility` aggregate of `report.sets.gms`, that is extraction (0600a), refineries (19000), electricity and district heat (35011), gas distribution (35002), water and waste (36000, 37000, 38391–38395), plus the agricultural industries that deliver straw and biogas. Refineries (19000) and gas distribution (35002) are the two the code names explicitly, the latter through `qY_CET['Natural gas incl. biongas','35002',t]` in `03_production_CES_energydemand.gms`.

36. In the code the three are `vD[i,t]`, `jvE_i[i,t]` and `vtY_i_NetTaxSub[i,t]`, in `03_financial_accounts.gms`. Note that `jvE_i` defaults to zero when the energy modules are switched off, so in a partial run the energy bill silently drops out of EBITDA.

37. The subsets are `i_private_fin`, `i_private_nonfin` and `i_public` in `model/sets/input_output.sets.gms`.

38. The equation determines `vNetDividends['NonFinCorp',t]` in `03_financial_accounts.gms`. A corporate tax term `vCorpTaxes[t]` is present in the expression but commented out.

39. The comment in `03_financial_accounts.gms` reads: "Fow now we assume that corporations pay out any excess cash as dividends (issue stocks) And we do not calculate value of the firm for endogenous revaluations", with the typo in the source. The revaluation rate is `rRevaluations_s[sector,t]`, which enters `vNetRevaluations[sector,t] =E= rRevaluations_s[sector,t] * vNetFinAssets[sector,t-1]/fv`. It is calibrated at `t1` and is not listed in `G_`

`flat_after_last_data_year`, so it is the catch-all extension in `calibration.gms` that carries it forward. The interest rate is a single economy-wide rate of 4 per cent plus a calibrated sector-specific deviation.

4 Government and Public Finances

The modules listed below are best read as one block, drawing a boundary around the public sector. The block takes market prices, tax bases and trend growth as given, and returns the public balance sheet, tax revenue, public demand and the fiscal closure.[1]

The government described here is not an optimising agent but rather an accounting structure. There is no public sector objective function, no optimisation problem and no reaction function. Every revenue item is a rate or a share applied to a base that is determined somewhere else in the model, and every expenditure item is either a rate applied to such a base or an exogenous path. What the block delivers is (i) a complete and internally consistent public budget, (ii) the demand that the public sector places on the input-output system, and (iii) a closure rule that keeps the government's net financial position from drifting. This is a deliberate design choice, and it is what makes the fiscal side robust for counterfactuals: the modeller, and not the model, decides fiscal policy.

That said, a mechanical rule is not the same thing as an arbitrary rule. Several of the rules in this chapter can be viewed as reduced forms for decisions that an elected government could take. Public consumption is a constant share of GDP, which is what a government that wants to hold the public service level constant relative to the size of the economy would deliver. Transfers to households are indexed to the wage with a lag, which is what statutory indexation of benefits produces. The lump-sum item of Section 4.5 returns revenue windfalls to households and finances revenue losses, which is the reduced form of a government that neither runs down nor accumulates public wealth in response to a policy experiment. Reading them this way matters for interpretation: the rules are not behavioural equations that would survive a change in the policy regime, but they are also not innocuous accounting conventions, because they determine who ultimately bears the incidence of a shock.

What is determined inside the block:

- **Public revenue.** The total collection of direct taxes on wages, corporate profits and returns on wealth; indirect taxes on the input-output flows; the specialised carbon and energy taxes, including the revenue of the emission trading systems; and social contributions.
- **Public expenditure.** Public consumption, public investment, subsidies and transfers to corporations, and income transfers to households.
- **Fiscal balances.** The primary balance and the period-by-period evolution of the government's net financial assets.
- **Fiscal closure.** The lump-sum transfer required to stabilise the public net asset position over the long run, or to hold the primary balance at its baseline path in a counterfactual.

- **Net flows to the other sectors.** The consolidated net transfer from government to corporations, to households and to the rest of the world, which the firm, household and foreign accounts use as inputs.

What the block takes from the rest of the model, and from the policy modeller:

- **Macroeconomic tax bases.** The total wage bill, industry gross value added, industry EBITDA and depreciation, household net financial assets and the return on them, and the physical use of energy and the associated emissions. These are determined in the labour market, firm, household and energy modules.
- **Policy rates.** Statutory tax rates, carbon prices, ETS allowance prices and free allocations, subsidy rates, and the target ratio of public consumption to GDP, all set explicitly as policy inputs.
- **Scale.** The number of transfer recipients and the exogenous level of social contributions, both of which grow with the trend of the model rather than with a demographic projection.[2]

The block does not only receive and record. The taxes, duties and subsidies computed here are price wedges for firms and households, so they change behaviour directly. The tax rate on the return to household wealth, `trHh[t]`, enters the household Euler equation in `02_ramsey_household.gms` and therefore changes the intertemporal price of consumption. Likewise, the net transfer from government to corporations, `vNetGov2Corp_xI0[i,t]`, enters industry EBITDA and the residual cost term of the production function in `03_production.gms`, and therefore changes the output price.

At the top level the government is summarised by three aggregates. The primary balance is revenue less expenditure,

$$\text{PrimaryBalance}_t = \text{Revenue}_t - \text{Expenditure}_t$$

and this is the first equation in `04_government.gms`. The primary balance is then the only fiscal item that enters the government's balance sheet, which accumulates in `03_financial_accounts.gms` as

$$A^{gov}_t = A^{gov}_{t-1} + \text{PrimaryBalance}_t + \text{Interests}_t + \text{Dividends}_t + \text{Revaluations}_t$$

where A^{gov} is the government's net financial assets. Everything else in this chapter is the unpacking of the primary balance into a detailed list of flows, and the construction of a rule that pins down the path of A^{gov}.

How this chapter is organised

Section 4.1 takes the revenue side apart into indirect taxes, direct taxes, the carbon pricing instruments and a residual category of other revenues. Section 4.2 does the

same for the expenditure side and explains how public consumption reaches the input-output system. Section 4.3 deals with public production, which is the one place where the public sector appears in the model as a producer rather than as a budget. Section 4.4 collects the consolidated net flows that the other sectoral accounts consume, and is where carbon pricing shows up in the external accounts. Section 4.5 describes the closure, both in the baseline calibration and in a counterfactual, and closes the chapter by listing what the accounting framework does not contain.

4.1 Public Revenue

Government revenue is the sum of indirect taxes, direct taxes, and a residual category of other revenues,

$$\text{Revenue}_t = T_t^{ind} + T_t^{dir} + O_t^{rev}$$

Table 4.1.1 lists the items and where each of them is computed. Two conventions apply throughout. First, the block follows the national accounts in recording taxes gross: production subsidies are not netted off the revenue side but appear as an expenditure item in Section 4.2. Second, wherever an item cannot be tied to a modelled base, it is written as a share of a base that can be modelled, and that share is calibrated in the first model year so that the aggregate reproduces the data. Those residual shares are then held constant after the last data year. They are the price of having a complete budget without a full tax code.

Indirect taxes

Indirect taxes are dominated by taxes that are already embedded in the input-output pricing and demand system. Conceptually, these are the taxes that wedge purchaser prices from producer and import prices: net duties, product taxes and VAT-like components. In the code, the indirect tax aggregate combines four items,

$$T_t^{ind} = \underbrace{T_t^{Y} + T_t^{M}}_{\text{duties}} + \underbrace{T_t^{prod} - T_t^{ETS}}_{\text{production taxes}} + \underbrace{\sum_i T_{i,t}^{CO_2}}_{\text{carbon tax}} + \underbrace{\sum_i \sigma_t^{ind}\,\text{GVA}_{i,t}}_{\text{residual}}$$

The first two terms are the duties on domestic deliveries and on imports collected through the input-output module. They are the sum over the whole input-output table of a rate times a base, one rate for every combination of supplying industry and demand component. The same rates appear in the purchaser price of that cell, so the incidence and the base stay consistent with how goods are priced and demanded in the rest of the model.[3]

The third and fourth terms are production taxes and subsidies. Production taxes are levied per unit of output, at an average rate per industry taken from the national accounts, and the corresponding subsidies are computed in the same place but spent in Section 4.2. Both enter the firm's residual cost term in `03_production.gms` and therefore its output price.[4]

The subtraction of the ETS term deserves an explanation and is taken up below. The last term is the residual block of other indirect taxes, modelled as a single share of industry gross value added, common across industries. The share is calibrated so that the total indirect tax revenue matches the data in the first model year. Because it is written industry by industry, this residual also enters the net transfer from government to corporations of Section 4.4 at the right industry level.

Direct taxes

Direct taxes are split into three main bases plus a residual component,

$$T_t^{dir} = \underbrace{\tau_t^r \, r_t^{hh} A_{t-1}^{hh}}_{\text{returns on wealth}} + \underbrace{\tau_t^w W_t}_{\text{wages}} + \underbrace{\tau_t^c \sum_i (\text{EBITDA}_{i,t} - D_{i,t})}_{\text{corporations}} + \underbrace{\sigma_t^{dir} \left(\tau_t^r \, r_t^{hh} A_{t-1}^{hh} + \tau_t^w W_t \right)}_{\text{residual}}$$

The first term is a marginal rate times the realised return rate on household wealth times lagged household net financial assets. The return rate r^{hh} is not a parameter: it is the realised interest and dividend income of the household sector divided by its lagged assets, $r_t^{hh} \equiv \Pi_t / B_{t-1}$, and is determined in `03_financial_accounts.gms`. The second term is the wage tax, levied on the total compensation of employees from `06_labor_market.gms`. The third is corporate taxation, with the profit-like base implemented as the sum over all industries of EBITDA minus depreciation. The fourth is a residual, implemented as a share of the household direct-tax base, so that it moves with wage and return taxation without needing a separate modelled base.

The intention is not to represent a full microsimulation of the tax code, but to ensure that the big macro bases and their cyclicality are represented in a transparent way. Two consequences follow. First, the marginal rate on returns to wealth and the marginal rate on wages are not calibrated but instead set directly in the code at 25 and 40 per cent respectively.[5] Second, the residual share σ^{dir} then absorbs whatever gap remains between observed total direct taxes and what those three rates applied to the three modelled bases deliver. So the level of direct tax revenue is right by construction, and the split between the rate and the residual is an assumption. Only the part that runs through the rates has an incentive effect: the wage tax does not enter labour supply, since employment follows a Phillips curve, but the tax on the return to wealth does enter the household Euler equation.

Carbon pricing and the emission trading systems

This is where environmental policy enters the public budget. Three instruments are computed in `04_energy_and_emissions_taxes.gms` and handed to the government block, and they are treated differently from one another.

Energy duties and energy VAT are treated exactly like any other indirect tax. They are computed bottom-up, for every combination of energy end purpose, energy good and demand component, and then written into the input-output duty variables so that

they appear in the purchaser price of energy and in the duty aggregates T^Y and T^M above. Nothing further is needed in the government block.

The national carbon tax on emissions that are not related to energy use enters the indirect tax aggregate as a separate item, industry by industry,

$$T^{CO_2}_{i,t} = \tau^{CO_2}_{i,t} \, \text{Emissions}^{xE}_{i,t}$$

where the rate is a price per tonne and the base is the industry's non-energy emissions of CO_2 equivalent. In the current parameterisation the rate is non-zero only for two industries, the mineral-products codes `23001` and `23002`.[6]

The two emission trading systems are the interesting case. The allowance price is a policy input, and firms and households pay it on their emissions exactly as they would pay a tax, so the marginal incentive to abate is the same as under a carbon tax of the same level.[7] What differs is where the money goes. Free allocation is subtracted from ETS1 revenue and returned to the firm as an infra-marginal deduction, so it lowers the firm's tax bill without blunting its incentive to abate,

$$T^{ETS1}_{i,t} = \tau^{ETS1}_{t} \left(\text{Emissions}^{ETS1}_{i,t} - \text{FreeAllowances}_{i,t} \right)$$

and the same free allocation term appears with a positive sign in the firm's bottom deductions in `04_energy_and_emissions_taxes.gms`. The total of ETS1 and ETS2 revenue, net of free allocation, is then *subtracted* from the indirect tax aggregate, as shown in the expression for T^{ind} above, and added to the net transfer from government to the rest of the world in Section 4.4. The reason is that the allowances are traded in an EU-wide market: the national accounts record the payments inside production taxes, which is where the calibrated rate `tY_i_tax` picks them up, but the receipts do not accrue to the national budget. Deducting the endogenous ETS revenue from the production tax aggregate removes the double count, and routing it abroad records where it goes. Any part that comes back, whether as a share of auction revenue or as any other transfer, arrives through the exogenous item "transfers from foreign countries" below. The practical implication for policy analysis is important: in GreenREFORM-EU, raising the ETS price tightens the private sector's budget without loosening the government's, whereas raising the national carbon tax does both.

Other revenues

The "other revenues" category collects items that are important in national accounts but not naturally expressed as a wedge on input-output prices,

$$O^{rev}_t = \underbrace{S^{gov}_t + D^{gov}_t}_{\text{public production}} + \underbrace{w_t \, \text{Cont}^{exo}_t}_{\text{contributions}} + \underbrace{\left(\sigma^{cap}_t + \sigma^{ncap}_t\right) \sum_i \text{GVA}_{i,t}}_{\text{from corporations}} + \underbrace{F^{gov}_t}_{\text{from abroad}} + \underbrace{\tau^{\kappa}_t A^{hh}_t}_{\text{capital tax}}$$

Social contributions are modelled as a wage-linked payment: an exogenous contribution level indexed to the current wage per efficiency unit, w_t. Revenue from public

production, S^{gov}, and the depreciation of public capital, D^{gov}, are both computed from the public industries and are the subject of Section 4.3. Capital transfers from corporations and other transfers from corporations are parameterised as shares of total gross value added. Transfers received from abroad are an exogenous path. The capital tax is a rate on household net financial assets, and unlike the tax on returns to wealth it is levied on the contemporaneous stock rather than the lagged one.

Item	Code	Computed in
Total revenue	`vGovRevenue[t]`	`04_government.gms`
Indirect taxes	`vtIndirect[t]`	`04_government.gms`
Duties, domestic	`vtY[t]`	`06_input_output.gms`
Duties, imports	`vtM[t]`	`06_input_output.gms`
Production taxes	`vtY_Tax[t]`	`06_input_output.gms`
ETS revenue, deducted	`jvtCO2_ETS_tot[t]`	`04_energy_and_emissions_taxes.gms`
Carbon tax, non-energy	`jvtCO2_xE[i,t]`	`04_energy_and_emissions_taxes.gms`
Residual, share of GVA	`vtIndirect_other[i,t]`	`04_government.gms`
Direct taxes	`vtDirect[t]`	`04_government.gms`
Returns on wealth	`vtHhReturn[t]`	`04_government.gms`
Wages	`vtHhWages[t]`	`04_government.gms`
Corporations	`vtCorp[t]`	`04_government.gms`
Residual	`vtDirect_other[t]`	`04_government.gms`
Other revenues	`vGovRevOther[t]`	`04_government.gms`
Public production	`vGovRevGovCorp[t]`	`04_government.gms`
Depreciation, public capital	`vGovDepr[t]`	`04_government.gms`
Social contributions	`vCont[t]`	`04_government.gms`
Capital transfers, corp.	`vGovReceiveCorp[t]`	`04_government.gms`
Other transfers, corp.	`vGovReceiveCorpNonCap[t]`	`04_government.gms`
Transfers from abroad	`vGovReceiveF[t]`	exogenous, from data
Capital taxes	`vtCap[t]`	`04_government.gms`

Table 4.1.1: The revenue side of the government block. The items loaded from `data.gdx` for calibration are collected in the group `government_data_variables`; the calibration swaps that turn each observed total into a rate or a share are listed in the calibration stage of `04_government.gms`.

4.2 Public Expenditure and Transfers

Government expenditure is the sum of five components: public consumption, transfers to households, public investment, other expenditures, and a closing lump-sum item,

$$\text{Expenditure}_t = G_t + Z_t + I_t^{pub} + O_t^{exp} + L_t$$

Table 4.2.1 lists the items. The first four are described here. The lump-sum item L_t is the closure device and is the subject of Section 4.5; it is zero in the base year by construction.

Public consumption and its allocation

Government consumption G is an aggregate quantity computed as a proportion of GDP,

$$G_t = \gamma_t \, \mathrm{GDP}_t$$

where the ratio γ_t is calibrated in the first model year and held flat afterwards. This quantity is then allocated to the different industries in $[i]$ through an exogenous mapping. The allocation happens in `06_input_output.gms` through `rYM[i,d,t]` and `rM[i,d,t]`, evaluated at the demand component $d = g$.[8]

The set $[g]$ of government consumption types has a single element, also called `g`, so public consumption is a single demand component of the input-output system and there is at present no split by purpose.[9]

The GDP-share rule is the clearest example of a mechanical rule with a reduced-form reading. A government that wants to keep the volume of public services growing with the economy, and that faces costs which grow with the economy, will spend a roughly constant share of GDP. The rule is also the reason why public consumption in GreenREFORM-EU has a mild automatic-stabiliser property of the wrong sign: because it is tied to GDP rather than to trend GDP, a shock that lowers GDP lowers public consumption in the same period. In counterfactual runs this is usually not what the modeller wants, and the closure of Section 4.5 switches it off by holding real public consumption at its baseline path.

Transfers to households

Transfers to households and non-profit institutions are the product of a wage index and the number of recipients,

$$Z_t = w_{t-3} \, N_t$$

where w is the wage per efficiency unit of labour from `06_labor_market.gms` and N is the number of transfer recipients. The three-year lag is the model's representation of statutory indexation: benefits are adjusted to wage developments, but with administrative and legislative delay. In a steady state the lag has no effect on the level, because the wage per efficiency unit grows at the trend inflation rate and the lag is carried forward at that same rate.[10] Off the steady state the lag makes transfers a slow-moving expenditure item and dampens the response of household income to a wage shock.

The number of recipients N_t is exogenous, calibrated in the first model year and then flat in growth-adjusted terms, so in levels it grows at the trend growth rate of the model. Since there is no demographic module, an ageing scenario has to be imposed on this variable directly.

Public investment and other expenditures

Public investment is not a policy variable in the current version. It is the investment of the public industries, aggregated in `03_financial_accounts.gms` from exactly the same factor demand equations that govern private investment,

$$I_t^{pub} = \sum_{i \in i_public} \left(\sum_k I_{k,i,t} + \text{Inventories}_{i,t} \right)$$

so it is determined by the capital stock that the public industries need in order to deliver the output that public consumption demands, and by the installation cost function of Chapter 3. This is worth flagging because it means that a shock to public consumption automatically carries a public investment response with it, at the depreciation and adjustment-cost dynamics of the capital types involved.

The "other expenditures" bundle collects four items,

$$O_t^{exp} = \underbrace{T_t^{sub} + \sigma_t^{sub} \sum_i \text{GVA}_{i,t}}_{\text{subsidies}} + \underbrace{\sigma_t^{corp} \sum_i \text{GVA}_{i,t}}_{\text{transfers to corporations}} + \underbrace{F_t^{exp}}_{\text{transfers abroad}} + \underbrace{K_t^{acq}}_{\text{non-produced assets}}$$

namely subsidies, consisting of the input-output linked production subsidies of Section 4.1 plus a residual component parameterised as a share of gross value added; transfers to corporations, also a share of gross value added; transfers to the rest of the world; and net acquisitions of non-produced non-financial assets. The last two are exogenous paths. The two share parameters and the residual subsidy share are calibrated in the first model year and held flat afterwards.

Item	Code	Determined by
Total expenditure	`vGovExpenditure[t]`	sum of the items below
Public consumption	`vG[t]`	share of GDP, `vG2vGDP[t]`
by demand component	`vD[g,t]`	`rG_g[g,t] * vG[t]`, single element `g`
by industry	`rYM[i,g,t]`	exogenous, `06_input_output.gms`
domestic–import split	`rM[i,g,t]`	endogenous, `05_imports.gms`
Transfers to households	`vHhTransfers[t]`	`pW[t-3] * qPopTransfers[t]`
Public investment	`vI_public[t]`	factor demand of `i_public`
Other expenditures	`vGovExpOther[t]`	sum of the four items below
Subsidies	`vGovSub[t]`	`vtY_Sub[t]` plus a share of GVA
Transfers to corporations	`vGov2Corp[t]`	share of GVA
Transfers abroad	`vGov2Foreign[t]`	exogenous, from data
Non-produced assets	`vGovNetAcquisitions[t]`	exogenous, from data
Lump sum	`vLumpsum[t]`	closure, Section 4.5

Table 4.2.1: The expenditure side of the government block. All equations are in `04_government.gms` except the public investment aggregate, which is in `03_financial_accounts.gms`, the allocation of public consumption across industries, which is in `06_input_output.gms`, and the domestic–import split within each industry, which is in `05_imports.gms`.

4.3 Public Production

The public sector appears twice in the model: as a budget, which is the subject of the rest of this chapter, and as a producer. The producer is the subset `i_public` of the industries, which in the current data contains the single industry `off`, public administration and services.[11]

What is special about this industry is worth stating precisely, because less is special than the label suggests. The public industry uses the same nested CES production tree as every other industry, hires labour and capital through the same factor demand equations, and prices its output as a markup over unit cost through the same equation in `03_pricing.gms`.[12] Nothing in the production side of the model knows that this industry is public.

What is special is the accounting, and it is special in two ways. First, the demand for the industry's output does not come from a market. It comes from the public consumption rule of Section 4.2 through the exogenous industry composition of the demand component `g`. The industry therefore produces what the fiscal rule tells it to produce, and its price is a cost-plus price. This is what is meant by saying that public production is modelled by cost accounting: the quantity is set by the budget, and the price records what it cost to deliver, rather than both being determined by the intersection of a demand curve and a supply curve.

Second, the operating surplus of the public industry is not the income of a corporate sector. It is government revenue, and it enters the budget in two pieces,

$$S_t^{gov} = \sum_{i \in i_public} (\text{EBITDA}_{i,t} - D_{i,t}) + c_t^S, \qquad D_t^{gov} = \sum_{i \in i_public} D_{i,t} + c_t^D$$

where $D_{i,t}$ is depreciation, computed in `03_factor_demand.gms` as the user cost of capital times the depreciation rate times the lagged capital stock. Splitting the gross operating surplus into a net surplus and a depreciation component is the national accounts treatment of non-market output: the consumption of fixed capital is part of the value of the services the public sector delivers, so it must appear as revenue for the budget to balance against the expenditure that produced it. Adding the two lines gives back the gross operating surplus of the public industries plus the corrections.

The two correction terms c^S and c^D are there to reconcile the model's measures with the data. Each is calibrated in the first model year against the observed items for revenue from quasi-corporations, rent, and the depreciation of public capital, through the two swaps

```
-vGovRevGovCorp[t1],   vGovRevGovCorpCorrection[t1]
-vGovDepr[t1],         vGovDeprCorrection[t1]
```

in the calibration stage of the module.[13] Both corrections also appear in the net transfer from households to government in Section 4.4, which is where the residual is ultimately absorbed.

4.4 Net Flows to the Other Sectors

For consistency with the sectoral accounts, the government module also computes net flow aggregates that are used elsewhere. There are three of them, and each is consumed by a different part of the model.

The net flow between government and corporations is computed industry by industry, because the firm block needs it at that level,

$$\begin{aligned}\text{Gov2Corp}_{i,t} = &\left(\sigma_t^{corp} + \sigma_t^{sub} - \sigma_t^{cap} - \sigma_t^{ncap}\right)\text{GVA}_{i,t} \\ &- T_{i,t}^{CO_2} - \sigma_t^{ind}\text{GVA}_{i,t} - \tau_t^{c}\left(\text{EBITDA}_{i,t} - D_{i,t}\right)\end{aligned}$$

It collects everything that flows between the government and the corporate sector outside the input-output table: the four share-of-GVA items of transfers and residual subsidies received, less transfers paid to government, less the carbon tax on non-energy emissions, less the residual indirect taxes not covered by the input-output module, less corporate taxation. This variable enters industry EBITDA and the residual cost term of the production function in `03_production.gms`, so a change in any of these fiscal items changes the firm's costs and its output price.

The net flow between households and government collects the household-side taxes and transfers,

$$\text{Hh2Gov}_t = T_t^{r} + T_t^{w} + T_t^{dir,o} + \text{Cont}_t + \tau_t^{\kappa} A_t^{hh} + c_t^{S} + c_t^{D} - Z_t - K_t^{acq} - L_t$$

that is, wage taxes, wealth-return taxes, residual direct taxes, contributions and capital taxes, minus household transfers and the lump-sum item, and with the two public-production corrections of Section 4.3 added in. This variable is the fiscal term in the household budget constraint in `03_financial_accounts.gms`, so it is the single channel through which fiscal policy reaches household wealth accumulation, and hence consumption.

The net flow between government and foreign countries is

$$\text{Gov2Foreign}_t = F_t^{exp} + T_t^{ETS} - F_t^{gov}$$

that is, government transfers abroad plus the ETS revenue that was deducted from indirect taxation in Section 4.1, less the transfers received from abroad. This variable enters the accumulation of the rest of the world's net financial assets, alongside imports and exports. This is where carbon pricing shows up in the external accounts: emissions-related tax revenues are produced by the energy and emissions tax module, and then enter the government accounts as part of indirect taxation and, for the trading systems, as part of the foreign-flow reconciliation.

The three flows are not independent of the identities above; they are re-groupings of items that already appear in revenue and expenditure. Their purpose is to hand each counterparty sector exactly the number it needs, and the consistency of the re-grouping is checked. The report module `IncomeFlow.Report.gms` builds a full sector-by-flow matrix from these variables, and the test in `03_financial_`

`accounts.gms` aborts the run if net interests, dividends, revaluations, debt, equity or net financial assets fail to sum to zero across all five national accounts sectors. The rest of the world is the residual holder of equity and absorbs the residual in the interest account.

4.5 Fiscal Closure and Sustainability

In calibration and simulations, not every detailed item is meant to be solved endogenously in every period. The government block therefore uses a simple closure device: a lump-sum transfer between government and households is adjusted so that the public accounts have a well-defined long-run anchor. The device is used differently in the baseline and in a counterfactual, and it is worth separating the two because it is the counterfactual use that determines how the results of a policy experiment should be read.

The government balance sheet

The object being anchored is the government's net financial assets. Its law of motion was given in the introduction, and the composition of the stock is decided in `03_financial_accounts.gms`: the government holds a constant amount of equity relative to the size of the economy, and its net debt instruments are the residual,

$$A_t^{gov} = E_t^{gov} + B_t^{gov}, \qquad E_t^{gov} = E_{t-1}^{gov}$$

in growth-adjusted terms, so that all fiscal adjustment happens through the debt instruments B^{gov}. Interest is paid or received on the lagged stock of debt instruments, at a single economy-wide rate of 4 per cent plus a calibrated sector-specific deviation, and revaluations are a rate applied to lagged assets, calibrated in the first model year and carried forward rather than set from an endogenous asset price. The interest rate does not depend on the level of public debt.

Closure in the baseline

In the baseline calibration, the closure pins down the terminal condition and then smooths the adjustment backwards to avoid an abrupt jump. Two equations do this, both in the calibration block of `04_government.gms`,

$$A_{tEnd}^{gov} = A_{tEnd-1}^{gov}, \qquad L_t = 0.9\, L_{t+1} \quad \text{for } t_1 < t < tEnd$$

The first is the terminal condition. What the code pins is a growth-adjusted level: the net asset position in the final model year equals that of the year before. In levels this means public net wealth grows at the trend rate of the model, so along a balanced path it holds a constant ratio to the size of the economy, though no ratio to GDP is imposed as such. The lump-sum item in the final year is what adjusts to deliver it. The second equation is the geometric smoothing rule: working backwards from the

terminal year, the lump sum shrinks by ten per cent per period, so it is close to zero in the early projection years and reaches its full size only at the end of the horizon. In the base year it is set to zero outright.[14]

The effect is that the baseline is fiscally sustainable by construction, and that the sustainability is achieved by a transfer that is small in the near term. It is not a forecast of fiscal policy. It is a device that prevents the baseline from containing an explosive or imploding debt path that would contaminate every counterfactual run against it.

Closure in a counterfactual

The closure used in a shock is different, and it is set in the shock file rather than in the module. Two swaps are made before the solve,

`vG2vGDP` endogenous,	`qG` exogenous
`vLumpsum` endogenous,	`vGovPrimaryBalance` exogenous

The first swap holds real public consumption at its baseline path and lets the ratio of public consumption to GDP adjust residually. This switches off the pro-cyclical GDP-share rule discussed in Section 4.2, so the public service level is unchanged by the experiment.

The second swap holds the primary balance at its baseline path and lets the lump-sum transfer adjust to deliver it. This is the substantive closure assumption of the policy experiments that use it: the fiscal cost or windfall of the policy is returned to households as a lump sum in the same period in which it arises. A carbon tax that raises revenue is therefore automatically recycled, and a policy that costs revenue is automatically financed, by a transfer that has no incentive effect. The alternative closures, financing through a distorting tax or through debt, are not the default and would require replacing this swap.

The swap is written in the shock file, not in the module, so it is not automatic. The carbon tax template and the stepwise shock driver both set it, but the energy duty shock does not: it changes the duties and solves with the default endogenous group, so the primary balance moves and the lump sum stays at its baseline path. Which closure a given experiment uses therefore has to be read off the shock file.[15]

The reporting is built around the same logic. In `Public_finances.py`, in the `model/Report/` folder, the primary balance in the shock is reported with the change in the lump sum added back,

$$\frac{\text{PrimaryBalance}_t^{shock} + L_t^{shock} - L_t^{base}}{\text{GDP}_t^{shock}}$$

so that the reader can see the budgetary effect of the policy itself rather than the zero that the closure enforces, and a second series reports the balance excluding the lump sum for the same reason. The welfare decomposition in `Welfare.Report.gms` treats

the change in the lump sum as one of the components of the change in household income, alongside wages, transfers and household taxes, which is the accounting counterpart of the same recycling assumption.

What the accounting framework does not contain

There is no public sector objective function and no optimisation. There is no fiscal reaction function in the solved model: no tax rate and no expenditure share responds to the level of debt or to the output gap. The debt-stabilising terminal condition lives in the calibration model only, and in a counterfactual it is replaced by the fixed primary balance described above. There is no risk premium: the interest rate on public debt is exogenous and does not depend on the debt level, so there is no channel through which fiscal deterioration raises the cost of capital in the economy. There is no demographic projection driving expenditure. And there is no distinction between the statutory and the effective incidence of the residual items: whatever cannot be assigned to a modelled base is assigned to a share of gross value added or of the household tax base, and moves with it.

None of this prevents the block from doing its job, which is to deliver a consistent budget and a clean closure. But it does mean that GreenREFORM-EU is not a model of fiscal policy in the sense of debt sustainability analysis. It is a model in which fiscal policy is an input, the fiscal accounts are an output, and the two are reconciled by a lump-sum transfer whose size the reader should always look at before interpreting a simulation.

Notes

1. Consult `04_government.gms` for everything in this chapter except: the public balance sheet and the investment aggregate of the public industries are in `03_financial_accounts.gms`, the duties and production taxes that the government collects are computed in `06_input_output.gms`, and the carbon prices are computed in `04_energy_and_emissions_taxes.gms`.

2. There is at present no demographic module in GreenREFORM-EU. The age sets `a`, `a0_`, `a18_` and `aEnd` are declared in `model/sets/age.sets.gms`, but that file is not among the set files imported by `base_model.gms`, so the sets are not part of the running model at all. The number of transfer recipients is the single time series `qPopTransfers[t]`, calibrated in the first model year and then held flat in growth-adjusted terms. Bringing a demographic projection into the block is a natural extension and the transfer equation is already written in a form that would accept one.

3. In `06_input_output.gms` the revenue is `vtY_i_d[i,d,t] = tY_i_d[i,d,t] * vY_i_d_base[i,d,t]` and the matching price is `pY_i_d[i,d,t] = (1+tY_i_d[i,d,t]) * pY_i_d_base[i,d,t]`, with the analogous pair for imports. The base price is normalised by the duty rate of the base year, `tY_i_d[i,d,tBase]`, so that the base-year purchaser price is the industry price. The duty rates are calibrated cell by cell in the first model year. Energy duties and energy VAT reach this aggregate the same way: the links block of `04_energy_and_emissions_taxes.gms` writes the bottom-up energy tax revenue on the national accounts definition, `vtE_NAS`, into `vtY_i_d` and `vtM_i_d` for the energy demand components.

4. `vtY_i_tax[i,t] = tY_i_tax[i,t] * qY_i[i,t]` and `vtY_i_sub[i,t] = tY_i_sub[i,t] * qY_i[i,t]`, with the difference `vtY_i_NetTaxSub` entering `vProdOtherProductionCosts[i,t]`. The two aggregates `vtY_Tax[t]` and `vtY_Sub[t]` carry the same description string in the code, "Net production taxes and subsidies, total", but the first is the sum of taxes and the second the sum of subsidies.

5. `trHh.l[t] = 0.25` and `tW.l[t] = 0.4` in the exogenous values stage of `04_government.gms`. The corporate rate `tCorp[t]` *is* calibrated, in the swap `-vtCorp[t1], tCorp[t1]`.

6. `vtCO2_xE[d,t] = tCO2_xEmarg[d,t]/10**6 * qEmmxE['CO2e',d,t]`, with `tCO2_xEmarg.l['23001',t] = 125` and `tCO2_xEmarg.l['23002',t] = 125`. The division by 10^6 converts a price per tonne into the model's value units. The government block reads this through the term `jvtCO2_xE[i,t]`, one of two j-terms that stand in for the energy tax module so that the government block can be solved when the energy modules are switched off.

7. The default allowance prices are `tCO2_ETS.l[t] = 750` for ETS1 and `tCO2_ETS2.l[t] = 375` for ETS2, in currency units per tonne. The coverage split is by energy end purpose: ETS1 applies to the end purposes in the subset `in_ETS` and is defined over industries only, while ETS2 applies to the end purposes outside that subset and is defined over all demand components, so that it reaches households as well.

8. `rYM[i,d,t]` is the industry composition of demand component *d* and `rM[i,d,t]` the import share within industry *i* of that component. The two behave differently. `rYM` is exogenous, calibrated in the first model year from the observed input-output table and then held flat, so the split of public consumption across supplying industries does not respond to relative prices. `rM` is endogenous, determined by the Armington equation of `05_imports.gms`; what is calibrated and held flat there is the share parameter `rM0`. So the industry mix of public consumption is fixed, but the split between domestic and imported supply within each industry does respond to prices. The underlying data are `vIO_y` and `vIO_m` in `data.gdx`, the domestically produced and the imported part of the table.

9. The set is declared as `Set g[d<] "Government consumption types."` in `model/sets/input_output.sets.gms` with its elements loaded from `data.gdx`, and the data builder assigns it the single record `g`, in `data/preprocessing/read_data.py`. The share variable `rG_g[g,t]`, "Share of total government consumption expenditure by purpose", is therefore degenerate and solves to one. It is scaffolding for a future split of public consumption by function, and it is what the GDP-share equation is formally attached to in the code: `rG_g[g,t]$(first(g)).. vG[t] =E= vG2vGDP[t] * vGDP[t]`. The substance of the pair of equations is that the level of public consumption is a fixed share of GDP and its composition is exogenous.

10. The model is solved in growth- and inflation-adjusted variables, as described in Chapter 1 and implemented in `model/growth_adjustments.gms`. Written in levels, the equation `vHhTransfers[t] = pW[t-3] * qPopTransfers[t]` states that transfers equal the nominal wage of three years earlier, indexed forward by three years of trend inflation, times the number of recipients. Because the wage per efficiency unit grows at the trend inflation rate in steady state, the two effects cancel and transfers do not drift relative to the wage bill. Off the steady state the lag matters: a wage shock reaches transfers with a three-year delay, which is the point of writing it this way.

11. `Set i_public[i] "Public industries." / off /` in `model/sets/input_output.sets.gms`. The complement `i_private` is everything else, and is split further into the financial industry `64000` and the non-financial rest, because the financial accounts distinguish the two corporate sectors.

12. The subset `i_standard` in `model/sets/production.sets.gms` equals all of $[i]$, so the tree of Chapter 3 is the same for every industry including `off`. Its markup `rMarkup_i['off',t]` is calibrated in the first model year like any other and held flat afterwards.

13. The observed items are the data variables `vGovRevQuasi[t]`, `vGovRent[t]` and `vtGovDepr[t]`. They are loaded from `data.gdx` and used to initialise `vGovRevGovCorp` and `vGovDepr` before calibration; only the correction terms survive into the solved model.

14. Through the group `G_zero_t1_after_static_calibration`, applied in `calibration.gms` after the static calibration solve. This matters for reporting: the base year has no lump sum by construction, which is why `Report/Public_finances.py` overrides the first year when it reports the primary balance.

15. The swap above is in `shock_CO2_tax.gms` and `calc_shock_steps.gms`. `shock_model.gms`, which `run.py` also executes, doubles energy duties and solves the default `main_endogenous` with no closure swap. Note also that the carbon tax itself is phased in, through `phaseInTax`, while the closure swap is not phased: it applies in every projection period of the shock solve.

5 Rest of the World (Foreign Trade)

The modules listed below are best read as one block, drawing a boundary around foreign trade. The block takes domestic prices, foreign prices and global market sizes as given, and returns the volumes imported and exported.

The code files for the rest of the world as an agent specify the different export demand functions. There are two files, `05_exports.gms` and `05_exports_energy.gms`. One for goods and services other than energy, and one for energy. A third file, `05_imports.gms`, handles the substitution between domestic and imported goods.[1]

The three code files are small but their content is important. The rest of the world is where the model's prices are ultimately disciplined. The economy is a price taker on the import side: world prices are exogenous, so a domestic cost increase cannot be passed on to foreign suppliers. On the export side the economy is not a price taker: it faces a downward-sloping demand curve, so a domestic cost increase is passed on to foreign buyers, and they respond by buying less. The elasticity of that response is what converts a cost shock into a quantity shock, and it is therefore the parameter that determines how large the real effects of any policy experiment turn out to be.

What is determined inside the block:

- **Export quantities.** The volume of domestically produced goods and services demanded by foreign markets, for each of the export components of the input-output system.
- **Energy exports.** The volume of each energy good exported, indexed by energy good and end purpose rather than by export component.
- **Import shares and quantities.** The optimal mix and volume of foreign goods purchased to satisfy domestic demand, cell by cell in the input-output table.

What the block takes from the domestic economy and from the world market:

- **Domestic prices.** The prices of goods and services produced by domestic firms, determined in the firm and pricing modules of Chapter 3, and the duties levied on each input-output cell, determined in the government module of Chapter 4.
- **Foreign prices.** The exogenous world market prices for competing goods and for energy carriers. There are three of them, and they enter at different points: the price index of foreign goods competing with exports, the price of imports by industry, and the price of energy on the international market.
- **Global market size.** The exogenous scale of foreign demand targeted at the domestic economy, one series for each export component.

- **Total domestic demand.** The overall domestic demand for goods, from households, government and firms, that must be split between domestic and imported sources. The block does not determine the level of that demand, nor its composition across industries; it determines only how much of it is met from abroad.

Both sides of the block turn on one comparison: the domestic price against the foreign price. For exports, that ratio decides how much foreign demand reaches domestic firms. For imports, it decides the share of domestic demand met from abroad. Together the two give the trade balance, which accumulates into the foreign sector's claims on the domestic economy.

How this chapter is organised

Section 5.1 covers the export demand functions for goods and services, which is the heart of the block. Section 5.2 covers energy exports, which have the same structure but a finer index. Section 5.3 covers the import side, where the substitution between domestic and foreign supply happens cell by cell in the input-output table. Section 5.4 closes the chapter with the external balance, which is where the trade flows of the preceding three sections turn into a foreign asset position, and with the trade elasticities gathered in one place.

5.1 Export Demand

All export demand functions are of the type

$$q = q_0 \, p^{-\eta}$$

where q is a quantity exported, q_0 is a level object capturing market size, p is a home to foreign price ratio, and η is the price elasticity.[2] The elasticity η carries more of the model's behaviour than any other single parameter, for the reason set out at the end of this section.

These functions summarise the endogenous behaviour of the rest of the world following Armington (1969) and Anderson (1979). Through an appropriate specification of the market size variable q_0 and of the foreign price inside the price ratio variable, these equations capture the Armington assumption that countries look for goods from a variety of sources with a given substitution elasticity between them.

Note that the model does not contain an explicit Armington aggregator on the export side. A full Armington system would have foreign buyers minimising the cost of a bundle assembled from several sources, and the demand for the domestic variety would fall out of that problem with q_0 equal to the size of the bundle and p equal to the price of the domestic variety relative to the price index of the bundle. What GreenREFORM-EU writes down is the reduced form of that first-order condition: q_0 stands for the size of the foreign market and $p^{-\eta}$ for the share of it that the domestic variety wins. The advantage is that nothing needs to be known about the structure of foreign demand beyond one level and one elasticity. The cost is that the two cannot be identified separately from observation of the domestic economy alone, and the elasticity has to be imposed.

The export components

Regarding the code, exports are indexed on the demand set as $x \in [x]$. The set has two elements, `xOth` for exports other than energy and `xEne` for energy exports, and both are demand components of the input-output system in the same way that private consumption and investment are.[3]

Each export component draws on the industry set $[i]$ through the exogenous industry composition `rYM[i,d,t]` of `06_input_output.gms`, calibrated in the first model year from the observed table and then held flat. For non-energy exports the underlying data are the `xOth` column of `vIO_y` in `data.gdx`. Two consequences follow. The composition of exports across industries does not respond to relative prices, so a policy that raises the cost of one industry relative to another changes the level of exports but not their industry mix, except through the domestic-import split of Section 5.3. And exports can contain imported content: the cell `qM_i_d[i,x,t]` is not restricted to zero, so re-exports are represented.

Market size, the foreign price, and calibration

The two exogenous objects on the right-hand side are the market size q_0, coded `qXMarket[x,t]`, and the price index of competing foreign goods, coded `pRoW_x[x,t]`. Neither is observed, and the model does not pretend otherwise. The foreign price index is constant in growth-adjusted terms and never moves in the baseline, so the relative price is pinned by the domestic price alone. [4] The market size is then the residual: it is calibrated in the first model year so that the export demand function reproduces the observed level of exports, and it is held flat after the last data year.[5]

The practical reading of this is that the export block contains no information about the world economy. It contains a level that has been reverse-engineered from observed exports and an elasticity that has been assumed. What it does contain is the *mechanism*: a well-defined downward-sloping demand curve for domestic output, which is what the rest of the model needs in order to have a determinate price level.

Why the elasticity closes the model

Consider a policy that raises domestic production costs, for instance the carbon tax of Chapter 4. Domestic firms raise their output price by the markup rule of Chapter 3. Import prices do not move, because they are exogenous world prices. So the relative price p in the export function rises and the relative price in the import function of Section 5.3 moves against domestic goods. Exports fall by η times the price increase and the import share rises. That contraction in demand for domestic output is what forces the domestic factor prices down until the economy is consistent again, and in GreenREFORM-EU the adjustment runs through the wage, via the Phillips curve of the labour market module.

The larger η is, the smaller the price change the economy can sustain and the larger the quantity adjustment. In the limit of an infinite elasticity the economy is a price taker on both sides and domestic prices are pinned entirely by world prices. In the limit of a zero elasticity exports are a fixed quantity and the entire adjustment falls on prices. The default value is five, uniform across export components, which sits at the elastic end of the range usually estimated for aggregate trade.[6]

5.2 Energy Exports

Energy exports ($d = \texttt{xEne}$) have the same structure but more detailed indexing due to the specificities of energy goods. In the `05_exports_energy.gms` code the export demand equations appear with three indices, apart from time. There we find

$$q \equiv qEpj[es, e, \texttt{xEne}, t]$$

In the `data.gdx` file the energy balance item shows a one to one mapping between `xEne` and energy goods with an unspecified energy service. The extra indexing is included for generality, allowing for further more detailed modelling, as currently

these equations can effectively be written as $q \equiv qEpj[e,t]$ where they inherit the `input_output` mapping between energy goods $[e]$ and industries $[i]$.[7]

Written out, the equation is the same demand function as in Section 5.1,

$$q_e = u_e \left(\frac{p_e^{marg}}{p_e^{foreign}} \right)^{-\eta_e}$$

with three differences worth naming. First, the domestic price in the numerator is the *marginal* energy price `pEpj_marg`, which is the base energy price plus the marginal tax wedge computed in the energy and emissions tax module. Energy exports therefore respond directly to carbon and energy taxation to the extent that those taxes fall on exported energy. Second, the level term u_e (`uEpj_xEne[es,e,t]`) is described in the code as an Armington share but plays the role that market size plays in Section 5.1: it is calibrated in the first model year so that the function reproduces observed energy exports, and it is held flat afterwards.[8] Third, the elasticity is indexed by energy good, `eX_ene[e]`, although it too is set uniformly to five.

Energy exports are held fixed in the baseline

The calibration model contains an extra equation that sets the foreign energy price equal to the domestic marginal energy price in every year,

$$p_{e,t}^{foreign} = p_{e,\texttt{xEne},t}^{marg}$$

so that the price ratio is one and thus u_e equals energy exports exactly. The bottom-up energy exports are therefore frozen at their data values along the whole baseline path, and the export demand function has no work to do at that level. Note that this freezes the bottom-up block only: the aggregate export component `qD['xEne',t]` continues to follow the function of Section 5.1, with `qXMarket` flat in growth-adjusted terms, so the aggregate energy export volume still grows at the trend rate in levels. The code comment says as much and invites the modeller to change it.[9] The distinction matters: the baseline contains no independent projection of foreign energy demand, but a shock run does produce an energy export response.

How the two export functions fit together

Since the export demand function of Section 5.1 is written over the whole set $[x]$, the energy component `xEne` is described twice: once as an aggregate quantity index $qD[\texttt{xEne},t]$ and once good by good as $qEpj[es,e,\texttt{xEne},t]$. The two are not in conflict, because they determine different objects and the input-output layer reconciles them. The bottom-up function fixes the value of each energy good exported. The aggregate function fixes the quantity index of the export component. The value of the component is the sum over industries of the domestic and imported cells, so the price index $pD[\texttt{xEne},t]$ is the residual that makes value equal price times quantity, and the industry composition $rYM[i,\texttt{xEne},t]$ is endogenised by `06_energy_markets.gms`

to distribute the bottom-up quantities across supplying industries. The aggregate function is thus not redundant: it feeds back on the bottom-up level through the price index, which enters the marginal energy price.

5.3 Import Demand

The import side is where the model handles the substitution between domestically produced goods and imported goods. Using a constant elasticity of substitution framework, the import share adjusts on the relative price of imports to domestic output, so that domestic agents shift towards foreign goods if domestic production becomes relatively more expensive.

The place where this happens is not the aggregate. It is the individual cell of the input-output table. Every demand component d, whether it is a household consumption type, an investment good, an intermediate input or an export, draws on every supplying industry i, and for each such cell the model decides how much comes from domestic production and how much from abroad. The allocation is the pair of identities in `06_input_output.gms`,

$$qY_{i,d} = \left(1 - r^{M}_{i,d}\right) r^{YM}_{i,d}\, q^{D}_{d}, \qquad qM_{i,d} = r^{M}_{i,d}\, r^{YM}_{i,d}\, q^{D}_{d}$$

where q^{D}_{d} is the real demand of the component, determined by whichever module owns it, $r^{YM}_{i,d}$ is the exogenous industry composition of that demand, and $r^{M}_{i,d}$ is the import share. The whole of `05_imports.gms` is the equation that determines r^{M}.

The substitution rule

The rule is the first-order condition of a CES aggregate of the domestic and the imported variety. Writing σ for the elasticity of substitution and r^{M}_{0} for the share parameter, it says that the ratio of imported to domestic quantity in the cell moves with the relative price,

$$\frac{qM_{i,d}}{qY_{i,d}} = \frac{r^{M}_{0,i,d}}{1 - r^{M}_{0,i,d}} \left(\frac{p^{Y}_{i,d}}{p^{M}_{i,d}}\right)^{\sigma}$$

which is how the code writes it, cleared of denominators.[10]

The share parameter r^{M}_{0} is, as its description in the code says, the import share that would obtain if the relative price were one. It is calibrated cell by cell in the first model year so that the model reproduces the observed domestic and imported values, and it is then held flat after the last data year. The elasticity is set to two, uniformly across all cells.[11]

Two boundary cases are handled outside the equation. Where a cell has imports but no domestic supply, the import share is set to one; where it has domestic supply but no imports, it is set to zero. In both cases the substitution equation is switched off by its domain condition, which requires both dummies to be present. There is therefore

no extensive margin in the model: a cell that has never imported will never import, however expensive domestic production becomes.

Import prices are world prices

The import price of a cell is built up from a single exogenous world price per supplying industry,

$$p^M_{i,d} = \left(1 + t^M_{i,d}\right) \frac{1 + jf^M_{i,d}}{1 + t^M_{i,d,base}} \, p^M_i$$

where p^M_i is the price of imports from industry i, $t^M_{i,d}$ the duty on that cell, and $jf^M_{i,d}$ a calibrated deviation of the cell price from the industry average. The object p^M_i appears on the left-hand side of no equation in the model: it is exogenous.[12] This is the small open economy assumption, and it is the counterpart of the downward-sloping export demand curve: the economy cannot influence what it pays for imports, but it can influence what it receives for exports, and it must then bear the quantity consequences.

Energy imports

For energy goods the picture is more involved, because the quantities are determined bottom-up in physical units by the energy modules rather than by the input-output identities. The reconciliation works as follows. The substitution rule of `05_imports.gms` continues to determine r^M for energy cells exactly as for any other cell. What changes is that the share parameter r^M_0 becomes *endogenous* for those cells: `06_energy_markets.gms` adds an equation that requires the resulting imported quantity in each energy cell to equal the imported part of the bottom-up energy allocation, and r^M_0 adjusts to deliver it. The industry composition r^{YM} is endogenised in the same way, against the domestic part.[13] The effect is that for energy the domestic-import split is decided by the energy market model of Chapter 3, and the input-output share parameters follow along.

The world price of energy gets a separate treatment, and it is the most consequential exogenous foreign price in the model. In the default configuration the average supply price of every energy good that has both domestic production and imports is exogenous, as is the domestic producer price of that good. The producer's markup and the import price of the good are endogenised to make the accounting consistent.[14] So domestic energy prices in GreenREFORM-EU do not respond to domestic conditions at all: they are given from outside, and the market clearing of Chapter 3 works on quantities and markups. For a small economy in an integrated European energy market this makes sense: a domestic carbon price changes the price energy *users* face through the tax wedge, but not the price energy *producers* receive.

5.4 The External Balance and the Trade Elasticities

The three preceding sections determine trade flows. They do not determine what happens to the claims those flows create. That is done in `03_financial_accounts.gms`, where the rest of the world is one of the five national accounts sectors and its net financial assets accumulate as

$$A_t^{RoW} = A_{t-1}^{RoW} + M_t - X_t + \text{Gov2Foreign}_t + \text{Interests}_t + \text{Dividends}_t + \text{Revaluations}_t$$

The first three terms are the current account seen from abroad: a trade deficit and net transfers to foreign countries, the latter including the ETS payments of Chapter 4, add to the foreign sector's claims on the domestic economy. The last three are the return on the stock already held.

The foreign sector plays two residual roles in the financial accounts. It is the residual holder of equity, so that net equity sums to zero across the five sectors, and it absorbs the residual in the interest account, so that net interest payments sum to zero as well. A test in the same module aborts the run if net interests, dividends, revaluations, debt, equity or net financial assets fail to sum to zero across the sectors.

Two features are absent, and they follow from the same design choice as in Chapter 4. There is no exchange rate. All prices are in a single currency and the relative price ratios of this chapter are the model's real exchange rate, so a nominal devaluation cannot be simulated. And there is no closure rule on the external position: nothing in the model prevents the foreign asset stock from drifting, because the interest rate paid on it is the same exogenous four per cent that every sector faces and there is no risk premium that would respond to the level of external debt. The external position is a residual that should be inspected after a simulation, not an anchor that disciplines it.

The trade elasticities in one place

Because the elasticities of this chapter carry so much of the model's behaviour, we collect them in Table 5.4.1.

Elasticity	Code	Value	Governs
Export demand	`eX[x]`	5	foreign demand for the two export components
Energy export demand	`eX_ene[e]`	5	foreign demand for each energy good
Domestic–import substitution	`eM_i_d[i,d]`	2	the import share of every IO cell
Energy distributor	`eDist[out]`	5	the sourcing of energy across suppliers

Table 5.4.1: The trade elasticities. The first three are set in `05_exports.gms`, `05_exports_energy.gms` and `05_imports.gms`; the fourth is set in `06_energy_markets.gms` and is described in Chapter 3, where it allocates total demand for an energy good across domestic and importing suppliers. None of them has a time index, none is calibrated, and none varies across the set it is indexed by.

Table 5.4.2 lists the variables of the block and where each is determined.

Object	Code	Status
Export quantity, by component	qD[x,t]	endogenous, 05_exports.gms
Export price ratio	pX2pRoW_x[x,t]	endogenous, 05_exports.gms
Foreign competing price	pRoW_x[x,t]	exogenous; initialised to one
Export market size	qXMarket[x,t]	calibrated in t_1, then flat
Energy exports, by good	qEpj[es,e,'xEne',t]	endogenous, 05_exports_energy.gms
Energy export level term	uEpj_xEne[es,e,t]	calibrated in t_1, then flat
Foreign energy price	pEpj_foreign[es,e,t]	exogenous; see the baseline caveat
Import share of a cell	rM[i,d,t]	endogenous, 05_imports.gms
Import share parameter	rM0[i,d,t]	calibrated in t_1; endogenous for energy
Relative price of a cell	pY2pM_i_d[i,d,t]	endogenous, 05_imports.gms
Import price, by industry	pM_i[i,t]	exogenous world price
Industry composition of demand	rYM[i,d,t]	calibrated in t_1; endogenous for energy
Total imports and exports	vM[t], vX[t]	endogenous, 06_input_output.gms
Foreign net financial assets	vNetFinAssets['RoW',t]	endogenous, 03_financial_accounts.gms

Table 5.4.2: The objects of the rest of the world block. Real and nominal aggregates of trade, and the chain-linked price deflators pX[t] and pM[t], are built in 06_input_output.gms; the sector-by-flow matrix that displays the external accounts is built in Report/IncomeFlow.Report.gms.

Notes

1. Two further files carry part of the block. The world prices of imports and the allocation of each demand component between domestic and foreign supply live in 06_input_output.gms, and the foreign sector's balance sheet lives in 03_financial_accounts.gms. The energy modules 06_energy_markets.gms and 04_energy_and_emissions_taxes.gms take over part of the import allocation for energy goods, as described in Section 5.3.

2. Inside 05_exports.gms the two equations are qD[x,t] =E= qXMarket[x,t] * pX2pRoW_x[x,t]**(-eX[x]) and pX2pRoW_x[x,t] =E= pD[x,t] / pRoW_x[x,t].

3. The set is declared as Set x[d<] "Export types." in model/sets/input_output.sets.gms with the elements loaded from data.gdx, and the data builder assigns the two records in data/preprocessing/read_data.py. Note that the export demand equation of this section is written over the whole set $[x]$ and so applies to the energy component as well; how it coexists with the bottom-up energy export function is explained at the end of Section 5.2.

4. The assignment is pRoW_x.l[x,t] = fpt[t] as per the convention for exogenous prices which are constant in growth-adjusted terms, as 05_exports_energy.gms does for the foreign energy price and 06_input_output.gms does for the import price.

5. The calibration swap is -qD[x,t1], qXMarket[x,t1], so observed exports in the calibration year are the data and market size is backed out from them. Because qXMarket is a quantity variable, a flat growth-adjusted path means it grows at the trend growth rate gq in levels, which is one per cent per year. Foreign demand thus expands with the trend of the domestic economy in the baseline; a different assumption about world growth has to be imposed on this variable directly. The module writes the group entry as qXMarket[t] although the variable is declared qXMarket[x,t]; the outcome is the same either way, because calibration.gms extends the level of every non-constant variable flat.

6. eX.l[x] = 5 in the exogenous values stage of 05_exports.gms. The elasticity has no time index

and is not calibrated, so it is the same in every year of every run. Because it is a single scalar per export component, and because the model has only two of them, it is the cheapest and most informative sensitivity check available.

7. The variables are declared under the condition `d1pEpj_base[es,e,'xEne',t]`, so the equation is generated only for the combinations of energy end purpose and energy good that carry a base price for the export component in the data. The mapping `es_d2d` in `model/sets/output.sets.gms` contains the entry `set.es . xEne . xEne`, which admits every end purpose in principle; the dummy is what reduces it to the combinations that exist.

8. The swap is `-qEpj[es,e,xEne,t1], uEpj_xEne[es,e,t1]`. Unlike `qXMarket`, the name does not begin with `q`, so under the prefix rule of `model/variable_groups.gms` it is not growth-adjusted. Held flat, it therefore keeps growth-adjusted energy exports constant, which means energy exports grow at the trend growth rate in levels, as non-energy exports do.

9. The comment in the calibration block of `05_exports_energy.gms` reads: "In this equation we fix energy-exports to the value in data-year. Consider changing if you want baseline-forecast to produce varying energy-exports." The equation is part of `exports_energy_calibration_equations` and therefore belongs to the calibration model only. In a counterfactual run it is not present, `pEpj_foreign` is fixed at the path the calibration produced, and energy exports respond to the domestic marginal energy price with elasticity five as the function specifies.

10. The equation is `rM[i,d,t]$(d1Y_i_d[i,d,t] and d1M_i_d[i,d,t]).. qM_i_d[i,d,t] * (1-rM0[i,d,t]) =E= qY_i_d[i,d,t] * rM0[i,d,t] * pY2pM_i_d[i,d,t] **eM_i_d[i,d]`, with `pY2pM_i_d[i,d,t] =E= pY_i_d[i,d,t] / pM_i_d[i,d,t]`. The equation is named for `rM`, so the import share is the variable it determines. Note that `pY2pM` is a price ratio, not an individual price.

11. The swap is `-rM[i,d,t1], rM0[i,d,t1]` and the elasticity is `eM_i_d.l[i,d] = 2`. The prices in the ratio are the purchaser prices of the cell, so they include the duties of Chapter 4: a duty levied on imports into a particular cell raises its p^M and shifts that cell towards domestic supply.

12. It is initialised as `pM_i.l[i,t] = fpt[t]`, which is the convention for an exogenous price that is constant in growth-adjusted terms, and it is held flat after the last data year by the general extension in `calibration.gms`. The set of industries that have imports at all is `Set m[i] "Industries with imports."`, loaded from `data.gdx`.

13. The code marks this explicitly, with the comment "NOTE THAT THIS IS RM0 (not RM), BECAUSE OF THE 'IMPORTS.GMS'-MODULE THAT TAKES OVER RM IN INPUT_OUTPUT" above the equation `rM0&_energy_imports`. The bottom-up split between domestic and imported energy is the share `sSupply_d_e_i_adj`, which comes from the energy distributor of Chapter 3. An alternative version of the equation, which endogenises `rM` directly for the case where the imports module is switched off, is present but commented out in the same block.

14. This is the switch `exogenous_supply_prices`, set to one in `model/settings.gms` and implemented in `model/conditional_logic_config.gms`. It exogenises `pE_avg[e,t]` and `pY_CET[e,i,t]` and endogenises `rMarkup_out_i[e,i,t]` and `pM_CET[e,i,t]` for the energy goods in the subset `these_e`, which in `model/sets/energy_outputs.sets.gms` holds twenty-five of the twenty-six energy goods; `Captured CO2` is commented out of it. The swap is further conditional on the good having both domestic CET production and imports in the data. Both exogenous prices are calibrated in the first model year and held flat afterwards, so they are constant in growth-adjusted terms in the baseline. A world energy price scenario is imposed by shocking them.

6 Markets and Equilibrium

The modules listed below are best read as one block. It takes the decisions of households, firms, government and the rest of the world, and imposes the conditions under which those decisions are mutually consistent. What comes out is the equilibrium prices and quantities, together with the national accounts and the environmental accounts that summarise them.

Six modules make up the block. `06_input_output.gms` is the accounting skeleton: it allocates every demand component across industries and across domestic and imported supply, and it builds the aggregates of the national accounts. `06_non_energy_markets.gms` and `06_energy_markets.gms` contain the clearing conditions themselves, the first for ordinary goods and services and the second for the twenty-six energy carriers of the set $[e]$, twenty-five of which are subject to the exogenous supply price of Chapter 5. `03_pricing.gms` sets the industry output price. `06_labor_market.gms` clears the labour market and carries the wage Phillips curve. `06_emissions.gms` maps physical energy flows into greenhouse gases.

What the block determines

What is determined inside the block:

- **Market-clearing quantities.** Output by industry and by product, imports cell by cell, and total supply of each energy carrier. In GreenREFORM-EU the clearing conditions are conditions on quantities.
- **The wage.** The one price in the model that is attached to a market clearing condition, together with wage inflation, employment, and the industry-specific user cost of labour.
- **The domestic price vector.** Industry and product prices, each a markup over unit cost, and therefore a simultaneous system rather than a chain of markups, since one industry's price is another industry's cost.
- **Aggregate macroeconomic indicators.** Nominal GDP and gross value added, private and public consumption, investment, exports and imports, each with a chain-linked volume index and an implicit deflator.
- **Emission totals.** Greenhouse gases in CO_2e equivalents, aggregated from physical energy flows and from non-energy activity, and reported on both the national accounts and the UNFCCC conventions.

What the block takes from the other chapters:

- **Sectoral supply and demand.** The demand of households, government and foreign buyers, and the supply decisions of firms. The block does not determine these; it determines the terms on which they meet.

- **Statutory policy.** Tax rates, carbon prices, ETS quota prices, subsidies and emission targets, all set in Chapter 4.
- **Foreign prices.** World prices of competing imports and of energy carriers, and the size of the export markets, all set in Chapter 5.
- **Structural labour market conditions.** Structural employment and labour-augmenting productivity, both exogenous.
- **Calibrated wedges.** The markups, the distribution margins on energy and the industry composition of each demand component, calibrated in the first model year. Most are then held constant, but not all: the industry composition `rYM` is endogenised for the energy columns, and the import share `rM` is endogenous wherever a cell has both domestic and imported supply. What is held flat there is the share parameter `rM0`, not the share itself.

One correction at the outset

It is tempting to describe the block as a mechanism in which excess demand pushes prices up until supply and demand balance. That description does not fit GreenREFORM-EU, and almost everything in the chapter follows from the difference.

There is exactly one price in the model attached to a market clearing condition, and it is the wage. Goods prices are set by firms as a markup over unit cost. Import prices are exogenous world prices. Energy supply prices are exogenous under the default switch. So when demand for the output of an industry rises, the response in the first instance is a quantity response: the industry produces more at the prevailing price. Prices do move, and they move for good economic reasons, but they move because *costs* move, not because a market is short. Section 6.1 takes this apart and shows that the textbook picture is nevertheless present: the supply schedule slopes up, through installation costs at the industry level and through the wage at the level of the economy, and the demand schedule facing an industry slopes down, through imports and exports.

The interaction loop is therefore a cost loop rather than a scarcity loop. The block receives quantity demands and supplies from the other blocks. Higher demand raises production, employment and the investment rate; that raises wages and installation costs; that raises unit costs and, through the calibrated markups, output prices; those prices are broadcast back to households, firms and foreign buyers, who re-optimise; and the system settles when every identity in this chapter holds simultaneously, in every year of the horizon at once.

How this chapter is organised

Section 6.1 covers the input-output system, which is where the goods market clears and where the national accounts are built. It is the longest section and it contains

the discussion of supply, demand and price determination that the rest of the chapter relies on. Section 6.2 covers price setting and inflation, and is short because the pricing module is short. Section 6.3 covers the labour market, which is where the one genuine market clearing price of the model lives. Section 6.4 covers the energy market, which is organised quite differently from the goods market and needs the most space. Section 6.5 closes the chapter with the environmental accounts.

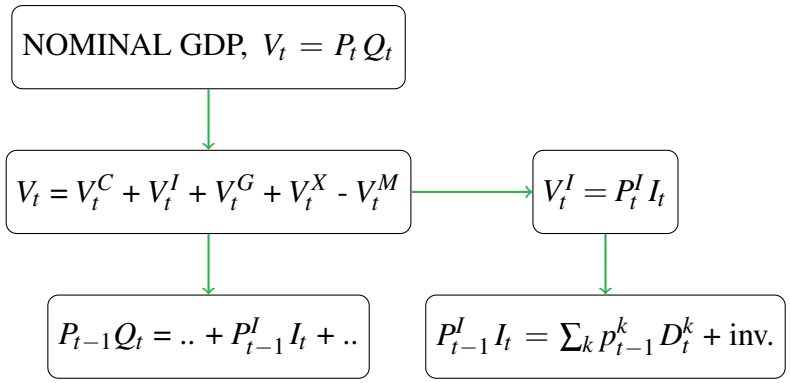
6.1 The Input-Output System (Goods Market)

The `06_input_output.gms` module binds together all objects in the model. It contains two fundamental elements. First, it works at the level of data aggregation used in the model. This is reflected in how fine a partition of the data is embodied in the sector index $[i]$.[1] And second, it contains the market clearing equations consistent with the data and the design of the model. Given these, the input-output module also contains all aggregation steps required to generate aggregate national accounts measures such as the general equilibrium expression of nominal GDP.[2]

Aggregate objects

Unlike the problems of the firm or the household where CES and Leontief aggregation is the working tool, at national accounting levels we need to use dynamic price indexing. Nominal objects can be summed, but aggregate price and quantity objects must be constructed with the index approach. And the need for the index structure arises not just for the top level of aggregation but also for intermediate levels.

The following diagram shows the indices used at the top levels of aggregation. The price of aggregate investment P^I is an index construction, as is the GDP deflator P.



The "inv." in the last node stands for the two inventory terms that the real investment index carries alongside the capital types. They are written out below, in the discussion of how the algebra closes, where they matter.

This algebra is contained in the `03_factor_demand.gms` and `06_input_output.gms` code files. The index constructions themselves are all in `06_input_output.gms`; `03_factor_demand.gms` supplies the objects that are aggregated, namely investment by capital type and industry and the intermediate input bundle of each industry.

Equilibrium

In order to understand the input-output system we must be clear about equilibrium, and about supply versus demand. Equilibrium is achieved at the market level, which

is the index $[i]$ level. This is where the industry prices are formed, and all other prices are derived from them. At this level, supply is defined in the production function of the firm. It is the quantity produced.[3] Demand, however, originates from all agents in the economy. Demand must be aggregated across agents, while supply originates from that single industry.

One point about the industry level needs care, because there is no equation anywhere in the model that reads "the price of industry i adjusts until supply equals demand". There are two separate equations. One is the market clearing identity above. The other is a pricing rule,

$$p_{i,t} = (1 + \theta_{i,t})\, p^0_{i,t}$$

in `03_pricing.gms`, where p^0 is the cost price delivered by the production nest of Chapter 3 and the markup θ is calibrated in the first model year and held constant thereafter. Firms are price setters with a constant markup, and the quantity produced is whatever is demanded at the resulting price. The code says as much in the comment above the first production equation: "Output is determined in the input-output system, to meet the demand at the prevailing price levels. Given the level of production, we determine the most cost-effective way to produce it in this module."[4]

It would be wrong to conclude from this that supply is horizontal and that prices respond only to shocks that hit costs from outside. Both schedules of the textbook picture are present, and it is worth being precise about where each of them comes from, because they come from unexpected places.

The supply schedule slopes up, for two separate reasons

The first reason is at the industry level and is easy to miss. Unit cost is not independent of the scale of production, because the cost price carries installation costs. Writing Q^0_i for output measured net of installation costs and $C^{inst}_{k,i}$ for the installation cost of capital type k, the code defines

$$p^0_{i,t} = p^{top}_{i,t}\left(1 + \frac{\sum_k C^{inst}_{k,i,t}}{Q^0_{i,t}}\right) + \frac{\text{other production costs}}{Q^0_{i,t}}$$

$$C^{inst}_{k,i,t} = \varphi\left(\frac{I_{k,i,t}}{K_{k,i,t-1}}\right)^2 K_{k,i,t-1}$$

where p^{top} is the marginal cost of the CES nest and φ is fixed at 0.5.[5] Producing more in period t requires more capital, and the lagged stock $K_{k,i,t-1}$ is predetermined, so the extra output has to be bought with investment. Installation costs are quadratic in the investment rate, so cost per unit of output rises with the rate of expansion. The same term appears a second time in the user cost of capital, through the marginal installation cost $\partial C^{inst}/\partial K$, which raises the price of the capital input to an expanding industry.[6]

Installation costs are the only source of scale dependence in unit cost, which makes the statement sharp. Everything in "other production costs" is proportional either to

output or to gross value added: net production taxes and subsidies are rates per unit of output, and the net flow between government and corporations is a set of shares of GVA_i.[7] The CES nest itself is constant returns to scale, so p^{top} does not depend on the level of output given input prices. Set $\varphi = 0$ and each industry's short-run supply schedule becomes horizontal at its markup price; at $\varphi = 0.5$ it slopes up, and φ is the parameter that sets the slope.

The second reason has nothing to do with installation costs and operates on the economy as a whole. The wage is the one price in GreenREFORM-EU that *is* attached to a market clearing condition: labour demand equals labour supply is the equation that carries p^W, and the Phillips curve is the equation that carries employment. Section 6.3 sets this out. Higher aggregate production requires more employment, employment above its structural level raises wage inflation, the wage enters every industry's cost price, and the markup passes it into every industry's output price. That is an upward-sloping aggregate supply schedule, and it survives even if installation costs are switched off. It is also the channel through which the export elasticity of Chapter 5 disciplines the domestic price level.

The price system is a fixed point, not a chain of markups

The second thing that "markup over unit cost" hides is that unit cost is itself a function of output prices. Each industry buys an intermediate bundle whose price is the input-output composite $p^r_{i,t} = p^D_{i,t}$, and that composite is a weighted average of the output prices of *all* industries and of import prices. Each industry also buys capital, whose user cost is built on the composite price of investment goods, again an average over all industries. So the pricing rule is not a recursion but a simultaneous system,

$$p^Y = (1+\theta) \odot c\left(p^Y, p^W, p^M, p^E\right)$$

in the vector of industry prices, which is the Leontief price system with markups. It has a solution, and the reason is instructive: each pass around the loop multiplies the domestic price weight by the domestic intermediate content of costs, which is strictly less than one in every industry because labour, imports and energy each take a positive share. Those three are what pin the level. The wage comes from the labour market, import prices are exogenous world prices, and energy supply prices are exogenous under the default switch of Chapter 5. Remove all three and the system would be homogeneous of degree one in prices and only relative prices would be determined.[8]

The fixed point also runs forward in time. The user cost of capital contains next period's investment good price and next period's marginal cost, discounted at the hurdle rate, so this period's price depends on next period's prices. That is why the model is solved as one system over the whole horizon rather than period by period, and it is the reason a shock announced for a later year moves prices today.

What the demand schedule facing an industry actually looks like

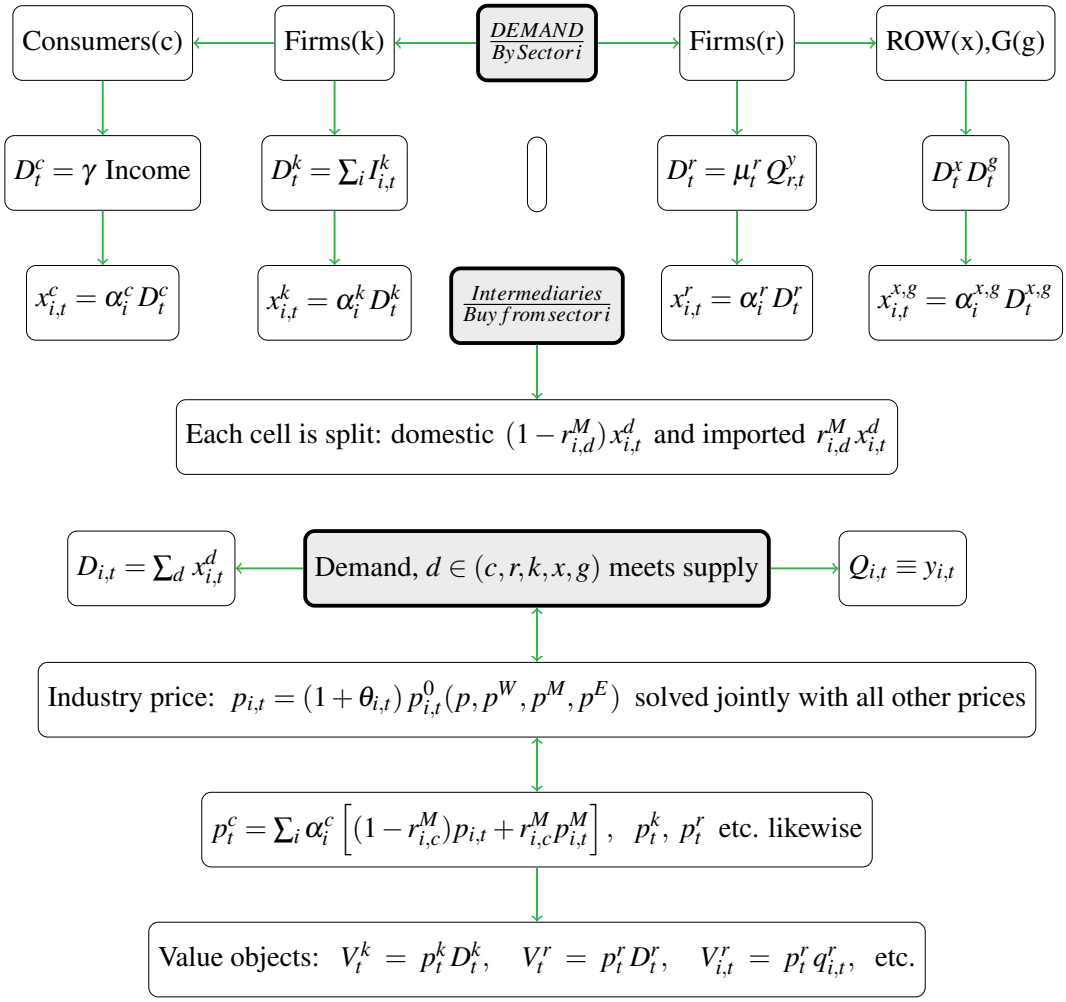
Demand slopes down, but not for the reason one would expect in a multi-sector model. Because the industry composition α_i^d of each demand component is exogenous, there is no substitution between domestic industries inside a cell: if industry i raises its price, no buyer shifts towards industry j. What remains are three channels. Within each cell the buyer substitutes towards imports, with elasticity two. On the export component the foreign buyer substitutes away, with elasticity five. And the level of the buyer's own demand falls, through the production nest of a purchasing industry, through the consumption nest of a household, or through the user cost of an investor. Where a cell has no imported twin, that cell's demand is inelastic with respect to the industry's own price, and the only response left is the level effect.

The two schedules therefore both exist, but the upward slope of supply is entirely a cost phenomenon rather than a scarcity phenomenon. Quantity is never rationed, the markup never widens under demand pressure, and the price of an industry moves only when one of its costs moves, whether that cost is the wage, an imported input, an energy price, an installation cost brought on by its own expansion, or the price of another domestic industry that supplies it. The one exception is energy, where the supply price is exogenous by construction and the markup, not the price, is what absorbs. This is taken up in Section 6.4.

A last word on how to read the equations. The model is a square system solved simultaneously, so the pairing of a variable with an equation in the endogenous groups is bookkeeping and not causality. Nothing is computed before anything else. What is substantive is which equations exist: a clearing identity and a markup rule for goods, a clearing condition and a Phillips curve for labour.

From demand components to industry output

In the next diagram firms in a given sector $i \in [i]$ spend $I_{i,t}^k$ on investment in capital type k. The total investment expenditure of type k across all industries is D_t^k. Intermediaries then require from a specific industry i the quantity $x_{i,t}^k$ to satisfy that aggregate demand D_t^k. But industry i meets demand from many sources, not just from "k intermediaries". Total demand aimed at industry i is given by $D_{i,t}$. In equilibrium this equals supply $Q_{i,t}$. The double arrow below that equality is a reminder that quantities and prices are settled together: the price $p_{i,t}$ enters all composite prices, and those composite prices come back into $p_{i,t}$ through cost.



Three features of the diagram deserve a note, because each is a place where the code is less symmetric than the picture.

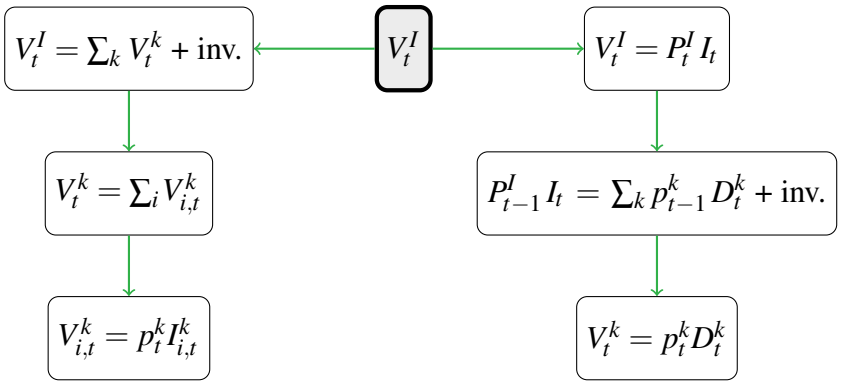
The coefficients α_i^d are the industry composition of demand component d, which is `rYM[i,d,t]` in the code. They are exogenous, calibrated in the first model year from the observed table and then held flat, so the industry mix of any demand component does not respond to relative prices. What does respond is the split of each cell between domestic and imported supply, $r_{i,d}^M$, which is the import share of Chapter 5 and is the extra row that the diagram carries below the intermediaries. The composite price of a demand component is then the exact weighted average of the domestic and the imported price in each cell, which is what the second to last node records.[9]

The intermediate branch is indexed differently from the investment branch. For capital there are three capital types and every industry invests in each of them, so D_t^k is a genuine sum across industries. For non-energy intermediate inputs there is one demand component per *purchasing* industry, because the set of input types $[rx]$ has been given the same elements as the set of industries $[i]$. The intermediate demand of industry r is therefore its own bundle, proportional to its output, and the sum across supplying industries happens one level down through the α coefficients.[10]

Finally, the equality $D_{i,t} = Q_{i,t}$ holds in values, and in quantities it carries a base-year

duty normalisation: the code writes `qY_i[i,t] =E= sum(d, qY_i_d[i,d,t] / (1+tY_i_d[i,d,tBase]))`, so each cell is converted to base-year producer prices before being added up.

At this point we want to relate our objects in the text with objects in the code. The reader should recall that quantity objects in the code have a prefix q while value objects (price times quantity) have a prefix v. Demand side quantities may vary in terminology but in the input-output system they will contain the letter D. The code uses the letter Y to define supply side objects since q is a prefix. The following diagram shows different ways of constructing value objects which can be found in the code.



Closing the algebra

Given equilibrium we can construct all price, quantity, and value measures within a given industry. But we cannot directly construct quantity or price measures for total aggregate consumption or investment. That must be done using the dynamic index construction detailed above. That structure uses lagged prices. Consider the aggregation of investment into the aggregate quantity object in the nominal GDP expression. It is created as follows

$$I_t = \frac{1}{P_{t-1}^I}\left[\sum_k p_{t-1}^k D_t^k + p_{t-1}^{inv} D_t^{inv} + p_{t-1}^{inv,e} D_t^{inv,e}\right]$$

where the two terms after the sum are inventories and energy inventories, which are elements of $[d]$ in their own right and not members of the capital set $[k]$.[11] The construction relies on the lagged aggregate price P_{t-1}^I. At calibration all prices are set (usually equal to one), and therefore this price will also be assigned an initial value.[12] Then, note that the value object on the left hand side of $V_t^I = P_t^I I_t$, is constructed separately as $\sum_k V_t^k$ and therefore the equation $V_t^I = P_t^I I_t$, determines the current aggregate price.

The nominal aggregate is not built over the same five components as the real index. It adds inventories to the three capital types but leaves energy inventories out, so the nominal and the real investment aggregate cover different parts of the demand side.

Where the clearing conditions actually sit

The identity $D_{i,t} = Q_{i,t}$ appears twice in the code, once in values and once in quantities, and the two live in different modules. The value identity `vY_i[i,t] + vtY_i[i,t] =E= sum(d, vY_i_d[i,d,t])` is in `06_input_output.gms`. The quantity condition is in `06_non_energy_markets.gms`, where output of the non-energy product of industry i is the sum of its deliveries across all non-energy demand components, plus the trade margins that the industry supplies on energy goods,

$$Q_{i,t}^{other} = \sum_{d \in D^{xe}} \frac{q_{i,d,t}^{Y}}{1 + t_{i,d,base}^{Y}} + q_{i,t}^{margins}$$

with the mirror condition for imports.[13] That this condition carries a *quantity* on its left-hand side is the concrete form of everything said above: the goods market clears in quantities. The margins term is there because wholesalers, retailers and car dealers deliver a service alongside every energy good, and that service has to be counted as part of their non-energy output. It is described in Section 6.4.

There is one more equilibrium object worth naming, because it is the model's own check on Walras' law. Aggregate private consumption is built in two ways: as the sum of the twelve consumption cells, and from the consumption function of Chapter 2. The difference between the two is carried by an explicit variable, and it should be zero, because the consumption nest is budget-exhausting.[14]

6.2 Price Setting and Inflation

`03_pricing.gms` is the shortest module in the model. It declares two variables and writes one equation,

$$p^{Y}_{i,t} = (1+\theta_{i,t})\, p^{0}_{i,t}$$

the industry output price as a markup over the cost price of the production nest. The markup is calibrated in the first model year, by swapping the observed price out and the markup in, and it is then held flat for the rest of the horizon.[15] Everything interesting about price formation is therefore not in this module but in the two objects it connects: the cost price p^0, which comes from the production nest of Chapter 3 and carries the whole fixed point of Section 6.1, and the markup, which is a number.

Two markups, and what they mean

There are in fact two markups in the model, at two levels of the supply side. The first is the industry markup above. The second is in the multi-output split of Chapter 3, where each product of an industry gets its own price,

$$p^{Y}_{o,i,t} = (1+\theta_{o,i,t})\, p^{0}_{o,i,t}$$

and where the calibration imposes a single markup per industry, common to all its products and constant over time.[16] The one place where this markup stops being a constant is energy, and there it becomes the model's shock absorber: under `exogenous_supply_prices` the energy supply price is fixed and the markup is endogenous, so a change in the cost of producing electricity shows up as a change in the producer's margin rather than in the price paid by users. Section 6.4 returns to this.

It is worth being clear about what the markup is not. A monopolistic competition markup is tied to the price elasticity of demand facing the firm, so that $\theta = 1/(\eta - 1)$, and it moves when that elasticity moves. Nothing in GreenREFORM-EU makes that link. The markup is a calibrated wedge, and the demand elasticity facing an industry — which Section 6.1 showed comes from imports and exports rather than from substitution between domestic industries — plays no part in setting it. The two are independent, which means the model reproduces the observed price-cost margins of the data year exactly and says nothing about where they come from.

The wedge between producer and purchaser prices

The price an industry receives and the price a buyer pays are not the same, and the two steps between them are both in `06_input_output.gms`. The first is a cell-specific deviation from the average industry price. The second is the duty on the cell:

$$p^{Y,base}_{i,d,t} = \frac{1+\phi^{Y}_{i,d,t}}{1+t^{Y}_{i,d,base}}\, p^{Y}_{i,t}, \qquad p^{Y}_{i,d,t} = \left(1+t^{Y}_{i,d,t}\right) p^{Y,base}_{i,d,t}$$

with the same pair for imports.[17] The division by the base-year duty is what makes the "base price" comparable across years: it strips out the duty rate of the base year so that the chain indices of Section 6.1 are built on a consistent valuation. The duties themselves are the product taxes and subsidies of Chapter 4, and they are the only part of the purchaser price that a policy maker sets directly.

There are no sticky prices

Nothing in the model makes prices costly to change. There is no price adjustment cost, no staggered contract, no partial adjustment of the price towards a target, and no lagged price in `03_pricing.gms`. A cost shock in year t is in the output price of year t in full.

Two frictions do exist, and it is easy to mistake either of them for price stickiness. The first is the installation cost of capital, which is a quadratic adjustment cost and which does slow the response of the economy. But it is a friction on *quantities*: it makes capital expensive to move quickly, and its effect on prices is the upward-sloping supply schedule of Section 6.1, not a delay in passing costs through. The second is the wage Phillips curve, which is a genuine nominal rigidity, and it is the only one in the model. In GreenREFORM-EU all inertia in the price level runs through the wage.

What inflation means in this model

The word inflation has to be handled carefully here, because GreenREFORM-EU has no money, no monetary policy rule and no nominal interest rate that responds to anything. What it has is the following.

Trend inflation is a setting, not an outcome. `settings.gms` fixes $g_p = 0.02$, and every price variable in the model is divided by the cumulative factor f_t^p before it is solved for, so a price that is constant in the solved variables is growing at two per cent a year in levels.[18] This is a normalisation: it fixes the units in which the baseline is reported and has no allocative content.

The one inflation rate that the model computes is wage inflation, and it is computed because the Phillips curve of Section 6.3 is written in terms of the change in wage inflation rather than its level. There is no corresponding equation for goods price inflation. The rate of change of any price index in the model — the GDP deflator, the consumption deflator, the export deflator — is whatever falls out of the identities that define the index, given the wage, the world prices and the calibrated wedges. This is the sense in which the block "establishes the macroeconomic equilibrium prices": it determines the whole vector of relative prices, and it determines the level too, but the level is pinned by the wage and by foreign prices rather than by a monetary authority.

The deflators themselves are all implicit. For every aggregate the module builds a nominal value by summation and a real volume by a chain index on lagged prices, and then defines the price as the ratio of the two, $P_t Q_t = V_t$. The GDP deflator is the

leading example, and it is defined exactly this way.[19]

6.3 The Labour Market

The current set-up is contained in `06_labor_market.gms`. It consists of the labour demand aggregated from the firm and government side of the model, a classical supply equals demand condition, and a Wage Phillips Curve (WPC).[20]

The aggregate labour demand originates in the optimal decision of the firm (where the marginal product value of labour equals its user cost)

$$L_t^d = \sum_j l_t^j$$

and sums labour demand from each firm j, l_t^j. The equilibrium condition is the standard supply equals demand expression $L_t^d = L_t^s$. The sum runs over all industries including the public sector, whose labour demand comes out of the same production nest as any private industry's, as described in Chapter 4.[21]

In our context the WPC generalises the household labour supply curve:

$$\mathscr{W}_t - \mathscr{W}_{t-1} = \mu^e \left[\mathscr{L}_t\right] + \mu^w \left[\mathscr{W}_{t+1} - \mathscr{W}_{t-1}\right]$$

$$\mathscr{W}_t = \frac{w_t \rho_t}{w_{t-1}\rho_{t-1}} - 1 \qquad \mathscr{L}_t = \frac{n_t}{n_t^*} - 1$$

where $l = \rho$, n is labour supply, w is the wage rate, ρ is a productivity term, and where n^* is structural employment which is exogenous.[22]

Unemployment is not modelled explicitly: the model has no unemployment rate or count of unemployed people. The employment gap, $\mathscr{L}_t$, measures employment relative to structural employment; it is not an unemployment rate and must not be interpreted as the share of the labour force without work. The model captures the wage response to employment moving above or below that benchmark, but does not model unemployment, labour-force participation or job search as separate mechanisms.

This structure allows us to add dynamics and rigidity to the labour market.[23]

The two coefficients

Both coefficients are set by hand, with no time variation and no calibration: $\mu^e = 5$ and $\mu^w = 0.3$.[24] The first is large. An employment gap of one per cent adds five percentage points to wage inflation, so the curve is steep, and the practical consequence is that employment stays close to its structural level while the wage does most of the adjusting. Anyone reading simulation output should expect small employment responses and large wage responses, and should recognise that this is a parameter choice rather than a result.

The second coefficient makes the curve forward-looking: expected future wage inflation feeds into today's. Together with the first this is what gives the labour market its dynamics, and it is why an announced future policy moves wages before it takes effect.

There is also an additive term, j_t^n, described in the code as "deviations from Phillips curve. Can be used to override the Phillips curve model". It is exogenous and zero in the baseline. Endogenising it against a fixed path for employment or for the wage is the standard way of running an experiment in which the labour market is not allowed to respond.

Where the nominal anchor is, and where it is not

The Phillips curve is written in differences. It determines the *change* in wage inflation, not its level, so the level of wage inflation is inherited from history and moved only by employment gaps. In the baseline it sits at the trend value implied by the growth adjustment, and the initial condition is what puts it there.

This has a consequence worth stating plainly, because it is the model's main adjustment mechanism and it is not obvious from the equations. Consider a permanent domestic cost increase. Prices rise, exports fall against the elasticity of five of Chapter 5, production and employment fall, the employment gap becomes negative, and wage inflation falls. It keeps falling for as long as the gap is open. Since nothing pins the level of wage inflation, the gap closes only when the wage *level* has fallen enough to restore competitiveness. The model has no exchange rate, so this permanent shift in the wage level is the real exchange rate adjustment. The Phillips curve is not a temporary nominal friction bolted onto a classical core; it is the mechanism through which the small open economy returns to external balance.

What the model does not have is a monetary policy rule, a money demand equation or a nominal interest rate that responds to inflation. The interest rate is exogenous and the same for every sector, as Chapter 4 and Chapter 5 note. So there is no channel by which a central bank could stabilise the employment gap, and no distinction between a nominal and a real shock in the usual monetary sense.

One wage, and no matching

Every industry pays the same wage. The user cost of labour is

$$p_{i,t}^L = p_t^W + \text{wedge}_{i,t}$$

and the wedge is identically zero: it is declared, never assigned, and therefore stays at its default of zero in every industry and every year. The wedge is also the whole of the model's labour market friction. [25] There are no compensating differentials, no industry wage premia, and no wedge between what the firm pays and what the worker receives other than the labour taxes of Chapter 4.

Labour supply separability

The labour market in our model is developed independently of the household problem. There is one aspect of this independence which is important to emphasise as it is

pervasive in general equilibrium models, and which is the removal of wealth effects from the labour market. See Greenwood, Hercowitz, and Huffman (1988).

In the code this separability is not an approximation but a construction: there is no labour supply equation derived from household preferences anywhere in the model. The household of Chapter 2 receives wage income and decides how much to consume and save; it does not decide how much to work. Employment is determined by the Phillips curve, and that is the precise content of the statement above that the WPC "generalises the household labour supply curve". A wealth effect on labour supply would require the household's intertemporal problem to include leisure, and it does not.

Calibration and the long run

Three things are calibrated in the first model year and then held fixed, and together they determine the long-run supply of labour.

Observed employment and observed compensation are data. The wage per efficiency unit is swapped against labour-augmenting productivity, so productivity is whatever makes the observed wage bill consistent with the observed number of employees. Productivity is then held flat in growth-adjusted terms, which means it grows at the trend rate $g_q = 0.01$.[26]

Structural employment is swapped against observed employment, so the employment gap is exactly zero in the calibration year, and a calibration equation then sets it equal to its first-year value in every subsequent year.[27] Structural employment is therefore a constant headcount over the whole horizon. There is no demographic projection behind it, which matches the finding of Chapter 4 that the age sets are declared but unused and that the transfer population is a single flat series.

The long-run supply of labour in efficiency units therefore grows at one per cent a year, entirely through productivity. A policy scenario that would change the size or composition of the workforce has to be imposed by shocking structural employment directly.

Object	Code	Status
Wage per efficiency unit	`pW[t]`	endogenous, carries the clearing condition
Employment, persons	`nL[t]`	endogenous, carries the Phillips curve
Labour in efficiency units	`qL[t]`	endogenous, sum of industry labour demand
Compensation per employee	`vW[t]`	endogenous; data up to the calibration year
Wage inflation	`rWageInflation[t]`	endogenous
User cost of labour by industry	`pL_i[i,t]`	endogenous, equal to `pW[t]`
Wage–user cost wedge	`pLWedge_i[i,t]`	exogenous, identically zero
Structural employment	`snL[t]`	calibrated in t_1, then constant
Labour-augmenting productivity	`qProductivity[t]`	calibrated in t_1, then flat
Phillips curve slope	`uPhillipsCurveEmpl[t]`	exogenous, 5
Forward weight	`uPhillipsCurveExpWage[t]`	exogenous, 0.3
Phillips curve override	`jnL[t]`	exogenous, zero in the baseline
Compensation by industry	`vWages_i[i,t]`	endogenous; data up to the calibration year

Table 6.3.1: The labour market block. All of it is in `06_labor_market.gms`. The industry labour demands `qL_i[i,t]` that are summed into `qL[t]` are determined in the production nest of Chapter 3.

6.4 Energy Markets and Distributors

The energy market is organised differently from the goods market, and the differences are worth listing before the equations arrive, because each of them explains a piece of machinery that has no counterpart in Section 6.1.

Energy is measured in physical units. Quantities are petajoules and prices are billions of kroner per petajoule, so every energy object has to be converted before it can enter the input-output table in kroner. Energy is demanded at a much finer resolution than it is supplied: a handful of industries produce each carrier, but demand is indexed by carrier, by end purpose and by demand component at once. And the price of energy is not formed the way the price of a good is formed. There is a single average supply price for each carrier, users pay that price plus a margin and a tax wedge that differ from user to user, and under the default switch the supply price itself is exogenous. All of this is in `06_energy_markets.gms`, with the tax wedges in `04_energy_and_emissions_taxes.gms`.

Energy in the input-output system

While energy has its own specificities it still links with the input-output structure the same way other inputs and goods do. Just like in the capital investment case we have $V^e_{i,t} = p^e_{i,t} q^e_{i,t}$, where $q^e_{i,t}$ is the quantity demanded of a particular energy type from sector $i \in [i]$. We have also $q^e_{i,t} = \mu^e_{i,t} Q^y_{i,t}$ and then

$$D^e_t = \sum_{es} \sum_{d} q^e_{es,d,t}$$

where the sum runs over the end purposes of energy and over all demand components that use energy, not over industries alone: households, exports, energy inventories and transmission losses are elements of $[d]$ in the same way industries are.[28] These

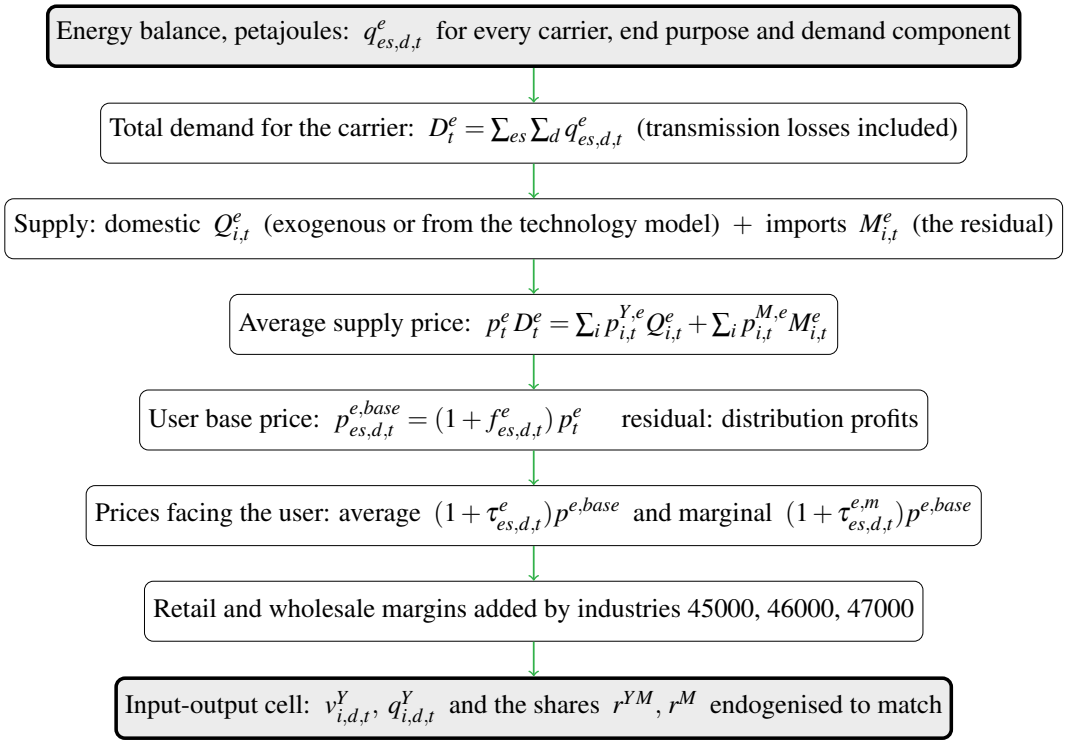
objects aggregate on the demand side for any energy type $e \in [e]$ without any difference relative to other inputs.[29] The crucial detail lies in the specification of $\mu^{e}_{i,t}$, which is the energy demand nest of Chapter 3 for industries and the consumption nest of Chapter 2 for households.

Total demand, total supply, and the two layers

The market for a carrier is a market in petajoules. Total demand D^{e}_{t} is the sum above. Total supply is the sum of what domestic industries produce and what is imported,

$$D^{e}_{t} = \sum_{i} Q^{e}_{i,t} + \sum_{i} M^{e}_{i,t}$$

where the industry index on imports records which industry the import is attributed to in the input-output table rather than who consumes it.[30] The diagram below traces the whole cascade from this market to the input-output cell.



The two ends of that diagram are two different accounts of the same energy. The top is the energy balance, in petajoules, disaggregated by end purpose. The bottom is the input-output table, in kroner, disaggregated by supplying industry. They come from different statistical sources and they do not agree cell by cell, so the module carries a set of calibration residuals that absorb the difference in the data year and are then forced to zero, and a set of tests that abort the run if the two accounts disagree in a model year.[31]

Clearing: who is the residual supplier

The clearing condition takes one of three forms, and which one applies to a carrier is decided by a dummy.[32] The three cases are:

- **Domestic supply only.** Domestic output equals total demand, $\sum_i Q^e_{i,t} = D^e_t$. Supply is demand-determined.
- **Imports only.** Imports equal total demand, so supply is once more determined by demand, this time at the exogenous world price.
- **Both.** Domestic supply is taken as given and *imports are the residual*, so that $M^e_t = D^e_t - \sum_i Q^e_{i,t}$. The code comment reads "for exogenous domestic suppliers imports are residual".

The third case is the one that matters, because it is the case that applies to almost every carrier, and it is a strong statement about how the energy market works.[33] For a carrier that is both produced at home and imported, domestic production is not determined by a supply decision inside this block. It is either extended from data or, when the energy technology model of Chapter 3 is switched on, determined there by the technology choice. Imports then absorb whatever demand is left over, at an exogenous world price. In effect the economy faces a perfectly elastic import fringe, and the domestic quantity is a policy or technology object rather than a market outcome.

The energy distributor

The first case in that list, the CES allocation across suppliers, is where the distributor of the chapter title lives. The construction introduces a fictive agent, described in the code as follows: "for the purpose of clearing energy markets, a fictive agent, the energy distributor, gathers a bundle of domestically and imported energy, before selling it to the end-sector". The distributor treats suppliers as imperfect substitutes and allocates total demand across them by the usual share rule,

$$Q^e_{i,t} = \frac{s^Y_{e,i,t}\left(p^{Y,e}_{i,t}\right)^{-\sigma^D_e}}{\sum_j s^Y_{e,j,t}\left(p^{Y,e}_{j,t}\right)^{-\sigma^D_e} + \sum_j s^M_{e,j,t}\left(p^{M,e}_{j,t}\right)^{-\sigma^D_e}} D^e_t$$

and the mirror expression for each importing supplier. The elasticity σ^D_e is set to five for every carrier, and the preference weights are calibrated in the first model year from the observed quantities and prices, exactly as an Armington share would be.[34]

The distributor is the natural way to describe an energy market in which suppliers are not perfect substitutes — electricity from one plant is not identical to electricity from another once location and timing are taken into account — and it is how the module is written. It is, however, almost entirely inactive as the model currently stands. The

dummy that selects the CES branch is switched on for one carrier only, straw for energy purposes; the line that would also switch it on for electricity and district heat is present in the source but commented out.[35] Every other carrier is handled by one of the two simple cases, and for most of them that means the residual-import case. A reader tracing the electricity market should therefore not look for the share equations; they are not in the active model.

Why the supply price is an average and not a CES price index

The distributor raises a question that the rest of this section takes for granted. A firm that minimises cost subject to a CES aggregator sells the aggregate, and the price that satisfies its zero profit condition is the CES price index. This distributor does not sell the aggregate. What it sells is the sum of the carriers it bought, $Q = \sum_j q_j$, because the output of an energy distributor is physically the same thing as its inputs: a petajoule in is a petajoule out. So the price it charges cannot be the CES price index, and the model uses an average instead.

Write the CES aggregate as F, with price index P satisfying $P^{1-\eta} = \sum_j \mu_j p_j^{1-\eta}$, and the cost-minimising demands as $q_j = \mu_j F p_j^{-\eta} P^{\eta}$. The price the model uses is

$$\bar{P} = \frac{1}{\sum_i \mu_i p_i^{-\eta}} \sum_j \mu_j p_j^{1-\eta}$$

which is the value-weighted average supply price p_t^e of this section. Two facts make it the right choice. First, it preserves the zero profit condition in the form the accounts need: $\bar{P}Q = PF$, so charging $\bar{P}$ for Q raises exactly the revenue that charging P for F would have raised.[36] Second, the allocation implied by $\bar{P}$ is the cost-minimising one: the share rule above, written on Q rather than on F, returns the same q_j as the CES demands do.

The gain is that everything in the pair $(\bar{P}, Q)$ is observable. The aggregate F is an auxiliary construction with no counterpart in any energy balance, whereas Q is a recorded quantity of petajoules and $\bar{P}$ is a recorded value divided by it. That is what allows the preference weights to be calibrated from observed quantities and prices, as they are above. F has not become meaningless: it still carries the technological content of how poorly the carriers substitute for one another, and so it still shapes the allocation. It is simply never priced or observed.[37]

From the average supply price to the price a user pays

Prices are the one place where energy is not symmetric with other inputs. There is a single average supply price p_t^e for each energy type, and the price a given user pays is that average price adjusted by a user-specific distribution margin and, at the margin, by the user's tax wedge. So $p_{i,t}^e \neq p_t^e$ in general, and the difference between what users pay and what suppliers receive is carried by an explicit residual, distribution profits.

The cascade has four steps and each is one equation. The average supply price is the value-weighted average of the prices received by domestic and foreign suppliers, so it is defined by the value identity of the market as a whole rather than by a supply and demand schedule. The base price of a particular user is that average marked up by a margin f^e. The price the user is charged is the base price plus a tax wedge, and there are *two* of these: an average wedge used for valuation and a marginal wedge used in the user's own optimisation.[38]

The distinction between the average and the marginal wedge matters and is easy to overlook. Energy taxes in Denmark are not proportional: there are thresholds, refunds and sector-specific rebates, so the tax paid per unit on average is not the tax paid on the last unit. The model keeps both, uses the marginal one in the demand nests, where behaviour is determined, and the average one in the value objects, where the national accounts have to add up. Two further cases complete the price system. Some energy is used from the user's own production and never appears in the national accounts as a transaction; it is priced at the producer's own output price. And some energy carries a quantity tax but no base price at all; there the price facing the user is the tax itself.[39]

Because the average supply price is common to all users while the base prices are not, the value of energy bought does not equal the value of energy sold. The gap is explicit:

$$\Pi^e_t = \sum_{es}\sum_{d} p^{e,base}_{es,d,t}\, q^e_{es,d,t} - \sum_i p^{Y,e}_{i,t} Q^e_{i,t} - \sum_i p^{M,e}_{i,t} M^e_{i,t}$$

described in the code as distribution profits, and noted there as something that can be negative. It is the counterpart of the assumption that a single average supply price clears the market for each energy good. In the input-output accounts it is attributed to the refineries industry on the import side, which keeps the value of imports consistent between the two layers.[40]

The supply price is exogenous, and the markup absorbs

The last step is the one that governs how the whole market behaves in a policy experiment. Under the default setting the supply price of energy is not determined by the cost of producing it. It is fixed, and the producer's markup is endogenised to absorb the difference between the fixed price and the cost price of the production nest. The average supply price is fixed at the same time, and the import price is endogenised so that the average identity still holds.[41]

The reason is the one given in Chapter 5. Danish energy prices are European energy prices, and a domestic policy does not move them. The consequence for this chapter is that the price channel of Section 6.1 is switched off for energy: a carbon price raises the price energy *users* face, through the tax wedge, but not the price energy *producers* receive, and the adjustment on the supply side is in quantities and margins. This is why the exception noted at the end of Section 6.1 exists.

Retail and wholesale margins

Energy reaches its user through a distribution chain, and in the national accounts the service that chain provides is output of the trade industries, not of the energy producer. The module therefore carries a margin per unit of energy delivered, attributed to three industries: car dealers, wholesale and retail.[42] The margin price moves with the non-energy output price of the industry that supplies it, normalised on its base-year value, and the margin rate itself is calibrated in the first model year and held constant.[43]

The margins are then converted into an output of the trade industry, and the conversion is built so that it generates no profit: the value of margins supplied is priced at the industry's own output price, and the residual is defined and turns out to be zero.[44] The quantity $q_{i,t}^{margins}$ that comes out is the term that appeared in the non-energy clearing condition of Section 6.1. This is how a supermarket's sale of gasoline shows up as retail output rather than as refinery output.

Linking the energy balance to the input-output table

The bottom-up layer knows how much of each carrier each user consumes, for each end purpose. The input-output table needs to know how much each user bought from each *industry*, and whether it was domestic or imported. The energy balance does not contain that information, so it has to be supplied, and the module does it in three steps.

First, an economy-wide domestic share for each carrier and industry, in values:

$$s_{e,i,t}^{Y} = \frac{v_{i,t}^{Y,e}}{\sum_{j}\left(v_{j,t}^{Y,e} + v_{j,t}^{M,e}\right)}$$

Second, that share is adjusted so that it reproduces the observed energy input-output table in the data year, and bounded above at one. The adjustment is a calibrated constant in the data year, and after the data years it becomes an endogenous shift term, pinned by the requirement that the deliveries allocated to industry i across all demand components add up to industry i's total domestic supply of the carrier.[45] The residual, once every industry's supply has been allocated, is the import content of the cell, and this is where the distribution profits noted above end up.

Third, and this is the step that matters for reading Section 6.1, the input-output coefficients of the energy columns are *endogenised* so that the table reproduces the bottom-up layer. The price deviation $\phi_{i,d}^{Y}$ of Section 6.2 is endogenised against the value of the cell, and the industry composition $r_{i,d}^{YM}$ and the import share parameter $r_{i,d}^{M0}$ are endogenised against a chain index of the bottom-level quantities:

$$q_{i,d,t}^{Y}\, p_{i,d,t-1}^{Y,base} = \sum_{e}\sum_{es} s_{d,e,i,t}^{Y,adj}\, p_{es,d,t-1}^{e,base}\, q_{es,d,t}^{e}$$

with an extra term on the right for the three margin industries, whose deliveries include the margin as well as the energy.[46]

So the statement of Section 6.1 that the industry composition of demand is exogenous holds for the non-energy columns of the table only. In the energy columns the composition is whatever the bottom-up energy allocation implies, and it moves whenever the fuel mix or the domestic supply share moves. This is the mechanical form of the model's central design idea: the energy system is modelled from the bottom up, and the input-output table is made to follow it.

Closing the energy loop

How does energy cycle back to the input-output structure? To detail this question we use the problem of the household where there is demand for energy under housing and transportation. We look at transportation. The demand for energy here falls under a diverse collection of sets associated with different coding purposes. The set $[d]$ lists all 81 demand side items including *cCarEne*.[47] The set $[es]$ with 7 elements lists all end purposes of energy (heating, transport, etc).[48] Finally the data maps $d = cCarEne$ into $es = transport$ (in this case a one to one mapping as car energy is only used in the energy service "transport") and this into five items indexed in the energy products set $[e]$. This mapping is a horizontal CES aggregation with five elements, contained in `02_consumption_disaggregated_energy.gms`.[49]

Which sectors i then produce these five items? This is the mapping: $e =$ Liquid biofuels $i = 19000$, $e =$ Gasoline for transport $i = 19000$, $e =$ Diesel for transport $i = 19000$, $e =$ Natural gas including bionatural gas (biongas) $i = 35002$, and $e =$ Electricity $i = 35011, 38393$.[50] This mapping also highlights the multi-output nature of sector 19000 which requires additional modelling of the production process on the supply side.

For the household there is no energy technology choice. So, whatever energy is being used is coming directly from the input-output matrix without any added filter. For the producers, their energy input choice is more complex so there is that added filter. But eventually it all goes back to the set $[i]$.

Object	Code	Status
Energy demand, PJ	`qEpj[es,e,d,t]`	endogenous, from the demand nests
Total demand for a carrier	`qEtot[e,t]`	endogenous
Domestic supply of a carrier	`qY_CET[e,i,t]`	exogenous for most carriers; see text
Imports of a carrier	`qM_CET[e,i,t]`	endogenous, the residual supplier
Average supply price	`pE_avg[e,t]`	exogenous under the default switch
Domestic supply price	`pY_CET[e,i,t]`	exogenous under the default switch
Import price of a carrier	`pM_CET[e,i,t]`	endogenous under the default switch
Producer markup on a carrier	`rMarkup_out_i[e,i,t]`	endogenous under the default switch
User base price	`pEpj_base[es,e,d,t]`	endogenous
Distribution margin	`fpE[es,e,d,t]`	calibrated in t_1, then constant
Price incl. average tax	`pEpj[es,e,d,t]`	endogenous; used in valuation
Price incl. marginal tax	`pEpj_marg[es,e,d,t]`	endogenous; used in the demand nests
Distribution profits	`vDistributionProfits[e,t]`	endogenous residual
Distributor weights	`sY_Dist`, `sM_Dist`	calibrated in t_1; straw only
Distributor elasticity	`eDist[out]`	exogenous, 5
Retail/wholesale margin	`pE_margins[es,e,d,i,t]`	endogenous; rate calibrated in t_1
Margin output of trade industries	`qD_E_margins[i,t]`	endogenous, zero profit
Domestic supply share by cell	`sSupply_d_e_i_adj[d,e,i,t]`	endogenous after the data years

Table 6.4.1: The energy market block, all in `06_energy_markets.gms` except the tax wedges, which are in `04_energy_and_emissions_taxes.gms`, and the CET supply split, which is in `03_production_CET.gms`. "The default switch" is `exogenous_supply_prices = 1` in `settings.gms`.

6.5 Emissions and Environmental Accounts

Emissions are not a market. Nothing is traded, no price clears anything, and no agent optimises over the quantity of a gas. What `06_emissions.gms` does is convert the physical flows of the preceding section into greenhouse gases and report the result on the conventions that the national and international accounts use. It is an accounting layer, and it sits at the end of the chapter for that reason.

Accounting is nevertheless where the model's environmental content is realised, because emissions are the object that policy acts on. The carbon prices and the ETS quota price of Chapter 4 are levied on the quantities computed here, and the emission targets of a scenario are stated in terms of them. So the layer has to be exact, and it has to be consistent with the physical energy flows rather than a parallel calculation.

Three blocks, and which one is in charge

The module is written as three blocks that can be run separately. The first computes emissions from the bottom up, at the finest index of the model. The second computes aggregates at the level of the demand component, in the style of a top-down macro model. The third links them, and when it is active the aggregates are no longer an independent calculation: their coefficients are endogenised so that the top-down numbers reproduce the bottom-up ones.[51] A reader should therefore treat the bottom-up block as the model of emissions and the aggregates as a reporting device.

Energy-related emissions

The bottom-up equation is as simple as it can be. For each gas, end purpose, carrier and demand component,

$$E^{BU}_{g,es,e,d,t} = \varepsilon_{g,es,e,d,t}\; q^{e}_{es,d,t}$$

where q^e is the physical energy of Section 6.4 in petajoules and ε is an emission coefficient in kilotonnes of the gas per petajoule. The coefficient is calibrated against observed emissions in the first model year and held constant afterwards, and the whole object exists only where the data records a nonzero flow, which is what keeps the five-dimensional array manageable.[52]

Because the coefficient is constant, the implication for policy analysis is direct: there is no abatement of energy-related emissions at given fuel use. Emissions fall only if energy use falls or if the fuel mix shifts towards carriers with lower coefficients. That is exactly the margin that the energy demand nests of Chapter 3 and the energy technology model provide, and it is the reason those parts of the model carry so much of the weight.

Natural gas is the one carrier with an internal split. A share of it is biogas, and that share determines how the same physical flow is divided between biogenic and non-biogenic CO_2. The share is exogenous, and the code notes that it should perhaps be handled through the emission coefficient instead.[53]

Aggregation to CO_2e

Gases are added up with global warming potentials,

$$E^{\mathrm{CO_2e}}_{\cdot} = \sum_{g \neq \mathrm{CO_2e}} \mathrm{GWP}_g \, E_{g,\cdot}$$

and the potentials are assigned by hand in the module. Methane carries 28 and nitrous oxide 265, which are the AR5 hundred-year values. The three F-gas categories carry one, because the Danish source data already reports them in CO_2e. Non-biogenic CO_2 carries one by definition.[54]

Biogenic CO_2 is not in the list, and the omission is deliberate rather than an oversight: no potential is assigned to it, so it receives no dummy and drops out of the CO_2e sum. This is the standard convention — combustion of biomass is treated as carbon-neutral in the inventory, with the land-use side of the account carrying the stock change — but it means the model's CO_2e totals are not a measure of total carbon in the atmosphere and should not be read as one. The biogenic flows are computed and available separately.

Non-energy emissions, and why they cannot be abated

Process emissions, agricultural emissions and the rest of the non-combustion inventory are handled by a second coefficient, applied not to energy but to output:

$$E^{xE}_{g,i,t} = \varepsilon^{xE}_{g,i,t} \sum_{\text{top nodes}} Q_{i,t}$$

with the coefficient again calibrated in the first model year and held constant.[55]

This is worth stating explicitly because it is a real limitation of the current model. Non-energy emissions are proportional to output with a fixed intensity, so there is no technical abatement margin: a carbon price on process emissions does not induce a change in the emission intensity of production, it only makes the output more expensive and so reduces it. For the industries where process emissions dominate — cement, agriculture — the model's response to a carbon price is therefore an output response and nothing else. The tax instruments exist in Chapter 4, and they work; what is missing is the engineering response on the receiving end, which on the energy side is supplied by the technology model of Chapter 3.

Two sets of national accounts

Emissions are reported on two conventions, and the difference between them is a bridge with four items. The residence principle, used in the green national accounts, counts the emissions of resident units wherever they occur. The territory principle, used in reporting to the UNFCCC, counts emissions occurring inside the country whoever causes them. The model computes the residence total by summation and then subtracts the four items to obtain the territory total:

$$\begin{aligned} E^{tot}_{g,a,t} = {} & \sum_d E_{g,d,t} + \sum_d E^{xE}_{g,d,t} + \text{LULUCF}_t \Big|_a \\ & - \Big[\text{border trade} + \text{bunkering} + \text{aviation} + \text{shipping}\Big]_{g,t} \Big|_a \end{aligned}$$

where the bracketed bridge applies to the two UNFCCC variants of the account and the LULUCF term to the two variants that include land use, giving four accounts in total.[56]

Three of the four bridge items are computed from the bottom-up layer rather than taken from data, which is a nice illustration of how fine the physical indexing is. International aviation is jet petroleum burned by the air transport industry. Bunkering is the set of carriers flagged as bunkering fuels, across all industries. And the shipping adjustment is a single cell: diesel for transport, used for transport, by one industry.[57]

Land use, land-use change and forestry is exogenous. It is a single series loaded from data and added to the two accounts that include it, so there is no land allocation decision and no response of the sink to prices or to policy.

The data year is checked. A test compares the total CO_2e on the green national accounts convention against the figure in the data and aborts if they differ by more

than 0.1 kilotonnes. A second, stricter test against the published totals of Statistics Denmark is present but disabled, with a note that the GreenREFORM-DK data differ from the published figure by about 500 kilotonnes CO_2e and that the source of the difference is under investigation.[58]

Object	Code	Status
Bottom-up emissions	`qEmmE_BU[em,es,e,d,t]`	endogenous, proportional to energy
Bottom-up coefficient	`uEmmE_BU[em,es,e,d,t]`	calibrated in t_1, then constant
Biogas share of natural gas	`sBioNatGas[t]`	exogenous, from data
Global warming potential	`GWP[em]`	exogenous, assigned by hand
Aggregate energy emissions	`qEmmE[em,d,t]`	endogenous; slaved to the bottom-up layer
Aggregate energy intensity	`uEmmE[em,d,t]`	endogenous when the link block is active
Non-energy emissions	`qEmmxE[em,d,t]`	endogenous, proportional to output
Non-energy intensity	`uEmmxE[em,d,t]`	calibrated in t_1, then constant
LULUCF	`qEmmLULUCF[t]`	exogenous, from data
Border trade	`qEmmBorderTrade[em,t]`	exogenous, from data
Bunkering	`qEmmBunkering[em,t]`	endogenous, from the bottom-up layer
International aviation	`qEmmInternationalAviation[em,t]`	endogenous, from the bottom-up layer
Shipping adjustment	`qEmmOtherDifferencesShips[em,t]`	endogenous, one cell
Emission totals by account	`qEmmTot[em,em_accounts,t]`	endogenous

Table 6.5.1: The emissions block, all in `06_emissions.gms`. Quantities are in kilotonnes of the gas, or of CO_2e for the aggregate. The energy flows `qEpj[es,e,d,t]` that drive the bottom-up layer are determined in `06_energy_markets.gms` and in the demand nests of Chapters 2 and 3.

What the chapter adds up to

It is worth closing with the mapping between the five sections and the four things that a general equilibrium model has to do.

Quantities are made consistent in Sections 6.1 and 6.4: every unit produced is delivered somewhere, every unit demanded is supplied from somewhere, and the energy accounts in petajoules and the input-output accounts in kroner are forced to agree. Prices are formed in Sections 6.1 and 6.2, as a simultaneous solution of the markup system, anchored by the wage, world prices and exogenous energy prices. The level of activity is determined in Section 6.3, where the only genuine market clearing condition of the model sits and where the Phillips curve supplies both the model's nominal rigidity and its mechanism of external adjustment. And the physical consequences are accounted for in Section 6.5.

What the chapter does not contain is a price adjustment mechanism for goods, a monetary authority, an exchange rate, or a scarcity constraint on any produced good. Understanding which of these are absent is as useful for reading simulation output as understanding the equations that are present.

Notes

1. The set $[i]$ is declared as `Set i[d<] "Production industries."` in `model/sets/input_output.sets.gms` and its elements are loaded from `data.gdx`, where the partition is fixed by the aggregation mapping `iagg2i` in the data builder. The module takes that partition as given and writes every market clearing and aggregation identity over it.

2. In the code nominal variables $Y = C + I + G + X - M$ have a prefix "v" for value.

3. Imported goods are twins of the set $[i]$ where supply is horizontal at the exogenous foreign price.

4. `03_pricing.gms` contains `pY_i[i,t] =E= (1+rMarkup_i[i,t]) * pY0_i[i,t]`, with the markup swapped in as `-pY_i[i,t1], rMarkup_i[i,t1]` at calibration and held flat afterwards, so it is an exogenous constant in any counterfactual. The clearing identity is `vY_i[i,t] + vtY_i[i,t] =E= sum(d, vY_i_d[i,d,t])` in `06_input_output.gms`, and the corresponding condition in quantities is carried by `06_non_energy_markets.gms` as described at the end of this section. The quoted comment is at the top of the equation block in `03_production.gms`.

5. The two equations are `qProd[pf_top,i,t] =E= qY0_i[i,t] + sum(k, qInstCost_k_i[k,i,t])` and `pY0_i[i,t] * qY0_i[i,t] =E= pProd['TopPfunction',i,t] * qProd['TopPfunction',i,t] + vProdOtherProductionCosts[i,t]` in `03_production.gms`, with `qInstCost_k_i` defined in `03_factor_demand.gms` and `fInstCost_k_i.fx[k,i] = 0.5` for every capital type and industry. The coefficient is fixed rather than calibrated, and the module contains explicit handling for the case where it is set to zero.

6. `pK_k_i` contains `pProd['TopPfunction',i,t] * dInstCost2dK_k_i[k,i,t]` with `dInstCost2dK_k_i[k,i,t] =E= fInstCost_k_i[k,i] * 2 * (qI_k_i[k,i,t] / (qK_k_i[k,i,t-1]/fq))`. So the investment rate enters both the level of cost and the price of the capital factor.

7. `vProdOtherProductionCosts[i,t]` is built from `vtY_i_NetTaxSub`, which is `tY_i_tax[i,t]*qY_i[i,t]` net of the corresponding subsidy, and from `vNetGov2Corp_xIO[i,t]`, which is a sum of shares multiplying `vGVA_i[i,t]` plus tax terms. None of these is a fixed cost, so none of them makes average cost fall with output.

8. The three links are `pProd[RxE,i,t] =E= pD[i,t] + jpProd[Rxe,i,t]` for intermediates, `pProd[pf_bottom_capital,i,t] =E= sum(sameas[pf_bottom_capital,k], pK_k_i[k,i,t] / pK_k_i[k,i,tBase])` for capital, and `pProd[labor,i,t] =E= pL_i[i,t]` for labour, all in `03_production.gms`. Since `pD[i,t]` and the `pD[k,t]` inside `pK_k_i` are the composites of the second to last node of the diagram below, every industry price appears in every other industry's cost.

9. Exact, not approximate: since `qY_i_d = (1-rM) rYM qD` and `qM_i_d = rM rYM qD`, dividing `vD[d,t] = sum(i, vY_i_d + vM_i_d)` by `qD[d,t]` gives precisely the expression in the node. The prices in it are the purchaser prices of the cell, so they include the duties t^Y and t^M of Section 6.2.

10. `qD[i,t] =E= qR2qY_i[i,t] * qY_i[i,t]` in `03_factor_demand.gms`, with the comment "We use a one-to-one mapping between types of intermediate inputs and industries". In the data builder, `rx=gp.Set(...,records=i_records)` with the description "this is just equal to i ATM", and the set file carries the warning "i should not be subset of d - use re or rx instead". The single intermediate energy component $[re]$ is discussed in Section 6.4.

11. `qI[t] * pI[t-1] =E= sum(k, pD[k,t-1] * qD[k,t]) + pD['invt',t-1] * qD['invt',t] + pD['invt_ene',t-1]*qD['invt_ene',t]`. The capital set has three elements, machinery, buildings and transport equipment, so this index runs over five demand components in total.

12. More precisely, prices are initialised at one in growth-adjusted terms and the pre-sample year t_0, which the chain indices need but the data does not cover, is given the first model year's price by calibration equations such as `pY_i_d_base[i,d,t0] =E= pY_i_d_base[i,d,t1]`. The code comment in `06_input_output.gms` states the same thing: "Lagged values used in chain price indices - should be added to data unless we switch to fixed price indices".

13. `qY_CET[out_other,i,t] =E= sum(d_non_ene, qY_i_d[i,d_non_ene,t]/(1+tY_i_d[i,d_non_ene,tBase])) + qD_E_margins[i,t] + jqY_CET_tDataEnd[out_other,i]$(tDataEnd[t])`. The last term is a calibration residual, endogenous in the data year only, and a test in the same module aborts if it exceeds 10^{-4} for any industry other than `02000`.

14. `vC[t] =E= sum(c, vD[c,t]) + vC_WalrasLaw[t]` in `06_input_output.gms`, with `vC_WalrasLaw` described in `02_households.gms` as "Equal to zero implying that Walras law regarding the sum of demand for private consumption equal to the sum of supply of private consumption is fulfilled". It is endogenous, paired with `vC[t] =E= vC_CMP[t] + jC_ramsey[t]`, and the top of the consumption nest is tied to `vC[t]` by `qCHh[cf,t] * pCHh[cf,t] =E= vC[t]` in `02_consumption_disaggregated.gms`, so adding-up in the nest makes the residual vanish. No test aborts on it, so it is a diagnostic to inspect rather than a guarantee.

15. `-pY_i[i,t1], rMarkup_i[i,t1]` in the calibration group and `$GROUP+ G_flat_after_last_data_year rMarkup_i[i,t]`. In the data year the markup is therefore whatever reconciles the observed producer price with the cost of the observed input bundle: it absorbs profits, measurement error in the value added split, and any genuine market power, without distinguishing between them.

16. `pY_CET[out,i,t] =E= pY0_CET[out,i,t] * (1 + rMarkup_out_i[out,i,t])` in `03_production_CET.gms`, with the calibration equation `rMarkup_out_i[out,i,t] =E= rMarkup_out_i_calib[i,t1]` and the swap `-pY0_i[i,t1], rMarkup_out_i_calib[i,t1]`. The product-level markup is common across $[out]$ by construction; product prices differ across products of the same industry only through the cost price $p^0_{o,i}$ of the CET split.

17. `pY_i_d_base[i,d,t] =E= (1+jfpY_i_d[i,d,t])/(1+tY_i_d[i,d,tBase]) * pY_i[i,t]` and `pY_i_d[i,d,t] =E= (1+tY_i_d[i,d,t]) * pY_i_d_base[i,d,t]`. The deviation ϕ is `jfpY_i_d[i,d,t]`, "Deviation from average industry price". It is calibrated in the first model year against the observed table and held constant, except in the energy columns, where `06_energy_markets.gms` endogenises it so that the value of an energy cell in the input-output table matches the value of the same energy at the bottom level. See Section 6.4.

18. The mechanism is in `growth_adjustments.gms` and the classification of variables into price, quantity and value groups is in `variable_groups.gms`. Chapter 1 sets this out and notes that the adjustment factors are suppressed in the equations of this documentation.

19. `pGDP[t] * qGDP[t] =E= vGDP[t]` with `vGDP[t] =E= vC[t] + vI[t] + vG[t] + vX[t] - vM[t]` and `qGDP[t] * pGDP[t-1] =E= pC[t-1] * qC[t] + pI[t-1] * qI[t] + pG[t-1] * qG[t] + pX[t-1] * qX[t] - pM[t-1] * qM[t]` in `06_input_output.gms`. The same three-equation pattern is repeated for consumption, investment, government, exports, imports, intermediate inputs and energy inputs.

20. A. W. H. Phillips (1958). The LyX source refers the reader to a document `labor_market_extensions.pdf` for additional context and possible extensions of the labour market modelling. That document is not in the repository.

21. `qL[t] =E= sum(i, qL_i[i,t])` in `06_labor_market.gms`, where `qL_i` is the labour input of the bottom node of the production nest of Chapter 3, and $[i]$ includes the public sector `off`.

22. In the code l is `qL[t]`, labour in efficiency units, n is `nL[t]`, the number of employees, ρ is `qProductivity[t]` and w is `pW[t]`, the wage per efficiency unit. The clearing condition is `pW[t].. qL[t] =E= qProductivity[t] * nL[t]`, the compensation per employee is `vW[t] =E= pW[t] * qProductivity[t]`, and wage inflation is `rWageInflation[t] =E= vW[t] / (vW[t-1]/fv) - 1`. The division by `fv` is the growth adjustment: since `vW` is solved in detrended units, restoring the level requires multiplying the lag by the trend factor, so a constant detrended wage corresponds to nominal wage growth of $(1+g_p)(1+g_q)-1$, about three per cent.

23. Ravenna and Walsh (2008), Vacancies, unemployment, and the Phillips curve, European Economic Review, Volume 52, Issue 8, Pages 1494-1521, derive a Phillips curve from first principles.

24. `uPhillipsCurveEmpl.l[t] = 5` and `uPhillipsCurveExpWage.l[t] = 0.3`. In the terminal year the forward term is dropped and the curve becomes `rWageInflation[t] =E= rWageInflation[t-1] + uPhillipsCurveEmpl[t] * (nL[t]/snL[t] - 1) + jnL[t]`, which is the standard way of closing a forward-looking difference equation at the end of the horizon.

25. `pL_i[i,t] =E= pW[t] + pLWedge_i[i,t]`, with `pLWedge_i[i,t]` described as "Wedge between wage and usercost of labour (e.g. matching costs), by industry". It appears in no other equation and receives no value in the `exogenous_values` stage, so `pL_i.l[i,t] = fpt[t]` equals `pW.l[t]` exactly. It is a hook for a future extension.

26. `-pW[t1], qProductivity[t1]` in the calibration group, with `$Group+ G_flat_after_last_data_year qProductivity[t]`. The data variables are `vWages_i[i,t]`, `nL[t]` and `vW[t]`, all loaded from `data.gdx` and protected by `data_covered_variables` up to the calibration year.

27. `-nL[t1], snL[t1]` and `$(not t1[t]).. snL[t] =E= snL[t1]`. Neither `nL` nor `snL` is a growth-adjusted variable, so this is a constant number of persons, not a constant growth rate.

28. `qEtot[e,t] =E= sum((es,d)$(d1pEpj_base[es,e,d,t] or tl[d]), qEpj[es,e,d,t])`. Industries appear in this sum through the demand component `energy`, which is the single element of the intermediate energy set $[re]$. Transmission losses enter through the `or tl[d]` condition, which lets them be counted even where no base price exists for them.

29. The energy types are the set $[e]$, a subset of the output set $[out]$ loaded from `data.gdx`, with twenty-six elements. It should not be confused with $[re]$, which is a subset of the demand set $[d]$ with the single element `energy`, and which is the input-output column through which all industrial energy use passes.

30. `qY_CET[e,i,t]` and `qM_CET[e,i,t]`. Which pairs (e,i) exist is recorded by the dummies `d1pY_CET[e,i,t]` and `d1pM_CET[e,i,t]`, both set from the loaded price data. The multi-output CET split of Chapter 3 is what allows one industry to appear against several carriers, and this is the same `qY_CET` that Chapter 3 supplies.

31. The residuals are `jvY_i_d_base`, `jvM_i_d_base`, `jqY_i_d`, `jqM_i_d` and `j_adj_sSupply_d_e_i_adj`, all described as "capturing difference in energybalance/energy-IO" and all set to zero for

$t > t_1$ by the calibration block. The tests compare the value of energy demand and of domestic and imported energy supply between the two accounts and abort on a discrepancy above 10^{-6}, and further tests abort on negative prices or negative quantities in the energy columns of the table.

32. `d1OneSX[e,t]`, with `d1OneSX_y` and `d1OneSX_m` recording whether domestic production and imports respectively exist for the carrier. The dummies carry no description string in the source, so the name is not expanded there.

33. `qM_CET&_ExoSuppliers[e,i,t]$(d1OneSX_y[e,t] and d1OneSX_m[e,t]).. qM_CET[e,i,t] =E= qEtot[e,t] - sum(i_a, qY_CET[e,i_a,t])`. Since the equation is paired with `qM_CET`, and `qY_CET[e,i,t]` is named by no other equation in this branch, domestic energy output is exogenous in the pure CGE model for these carriers.

34. `eDist.l[e] = 5`, and the calibration equations `sY_Dist[e,i,t] =E= qY_CET[e,i,t]/qEtot[e,t] * pY_CET[e,i,t]**eDist[e]` and its import counterpart, active in t_1 only. Chapter 5 lists this elasticity alongside the trade elasticities, since it plays the same role.

35. `d1OneSX[e,t] = yes;` followed by `d1OneSX[e,t]$(straw[e]) = no;`, with `# d1OneSX[e,t]$(straw[e] or el[e] or distheat[e]) = no;` on the next line. Since the CES branch applies when `d1OneSX` is false, the distributor allocates supply for straw and nothing else.

36. The proof is a substitution: $\bar{P}Q = \bar{P}\sum_i q_i = \bar{P}\sum_i \mu_i F p_i^{-\eta} P^{\eta} = P^{\eta} F \sum_j \mu_j p_j^{1-\eta} = PF$.

37. The derivation, and the wider treatment of these markets, is in "Energy good markets in GreenREFORM", Peter Stephensen and Jens Sand Kirk, May 2020. In model names $Q \equiv$`qEtot[e,t]` and $\bar{P} \equiv$`pE_avg[e,t]`, with the suppliers j being industry-origin pairs, so q_j is `qY_CET[e,i,t]` for domestic supply and `qM_CET[e,i,t]` for imports.

38. In order, `pE_avg[e,t] * qEtot[e,t] =E= sum(i, pY_CET*qY_CET) + sum(i, pM_CET*qM_CET);` `pEpj_base[es,e,d,t] =E= (1+fpE[es,e,d,t]) * pE_avg[e,t];` `pEpj[es,e,d,t] =E= (1+tpE[es,e,d,t]) * pEpj_base[es,e,d,t];` and `pEpj_marg[es,e,d,t] =E= (1+tpE_marg[es,e,d,t]) * pEpj_base[es,e,d,t]`. The margin f^e is `fpE`, calibrated in t_1 against the observed base price by the swap `fpE[es,e,d,t1], -pEpj_base[es,e,d,t1]` and held constant afterwards. The two tax wedges are built in `04_energy_and_emissions_taxes.gms` and described in Chapter 4.

39. `pEpj_own[es,e,d,t] =E= sum(i, pY_CET[e,i,t])` with the code comment "should be modified with a mapping when including straw", and `pEpj[es,e,d,t] =E= tqE[es,e,d,t]` for the non-priced case, selected by the dummy `d1tqEpj`.

40. `vDistributionProfits[e,t]` in `06_energy_markets.gms`, and `vM_i[i,t] =E= sum(out, pM_CET[out,i,t]*qM_CET[out,i,t]) + sum(e,vDistributionProfits[e,t])$(i_refineries[i]) + jvM_i[i,t]` in `03_production_CET.gms`, where `i_refineries` is industry 19000. The adjustment identity for the domestic supply share carries the comment "note, that this means that distribution 'profits' (positive or negative) are allocated to imports". The `ABORT` test on `jvM_i` is active; what the source note says is that it "should be adjusted to handle distribution profits".

41. `apply_exogenous_supply_prices_logic()` in `conditional_logic_config.gms`, active when `exogenous_supply_prices = 1`. It performs three swaps: `-pE_avg[e,t]` for carriers with both production and imports; `rMarkup_out_i[e,i,t], -pY_CET[e,i,t]`, which endogenises

the markup and exogenises the price; and `pM_CET[e,i,t]`, which endogenises the import price. The set of carriers affected is `these_e` in `model/sets/energy_outputs.sets.gms`, which lists twenty-five of the twenty-six elements of $[e]$.

42. `i_energymargins[i] /45000, 46000, 47000/` in `model/sets/input_output.sets.gms`.

43. `pE_margins[es,e,d,i,t] =E= fpE_margins[es,e,d,i,t] * pY_CET['out_other',i,t]/pY_CET['out_other',i,tBase]`, with the calibration swap `fpE_margins[es,e,d,i,t1], -vE_margins[es,e,d,i,t1]`. So margins per petajoule are constant in real terms and inflate with the price of the trade services that produce them.

44. `vD_E_margins[i,t] =E= sum((es,e,d), vE_margins[es,e,d,i,t])`, `pD_E_margins[i,t] =E= pY_CET['out_other',i,t]`, `vD_E_margins[i,t] =E= pD_E_margins[i,t] * qD_E_margins[i,t]` and `vOtherDistributionProfits_E_margins[i,t] =E= vD_E_margins[i,t] - pY_CET['out_other',i,t] * qD_E_margins[i,t]`, which is identically zero given the two preceding equations. The variable's description says as much: "currently modelled to produce zero-profit".

45. `sSupply_d_e_i_adj_inp[d_ene,e,i,t] =E= sSupply_e_i_y[e,i,t] + sSupply_d_e_i_adj_calib[d_ene,i]` and `sSupply_d_e_i_adj[d_ene,e,i,t] =E= min(sSupply_d_e_i_adj_inp[...],1)` in the calibration block; then `sSupply_d_e_i_adj[d_ene,e,i,t] =E= sSupply_d_e_i_adj[d_ene,e,i,tDataEnd] + adj_sSupply_d_e_i_adj[e,i,t]` for $t > t_{DataEnd}$, with the shift determined by `sum(d, sum((es,d_a)$es_d2d(es,d_a,d), sSupply_d_e_i_adj[d,e,i,t] * vEpj_base[es,e,d_a,t])) =E= vY_CET[e,i,t]`. A test on the bounds of the share is present but commented out.

46. The four equations are `jfpY_i_d`, `jfpM_i_d`, `rYM` and `rM0` in the `energy_markets_IO_link` block. The comment above them reads "the IO-share rYM as well as the import-share rM0 are endogenised to reflect bottom-level". A note in the same place observes that a simpler quantity index would do, and that the chain index is kept for consistency with the national accounts method. It is `rM0` rather than `rM` because `05_imports.gms` takes over `rM`; the version of the equation for the case where the imports module is switched off is present but commented out.

47. The 81 are the 57 industries, the 12 private consumption types, the 3 capital types, one public consumption type, two export types, inventories, energy inventories, the intermediate energy component `energy`, transmission losses, and the two natural input rows `env` and `res`. The list is assembled as `subsets_of_d` in `data/preprocessing/read_data.py`. The intermediate input set $[rx]$ adds nothing, since it repeats the industry elements.

48. Heating, Transport, Appliances, Process_normal, Process_special, In_ETS and `unspecified`, the last of which is appended by hand in the data builder and is the one that carries energy exports. The model set declaration in `model/sets/output.sets.gms` describes $[es]$ as "End-purpose of ergy", which is why this text says end purpose rather than energy service.

49. The mapping $d = cCarEne \rightarrow es = transport$ is the entry `transport . cCarEne . cCarEne` of `es_d2d`, and which (es,e) pairs actually appear under the node is decided by the dummy `d1pEpj[es,e,c,t]`. The substitution elasticity is `eChh_cEne['cCarEne'] = 0.5`, the same value as for household heating. The code notes that the household nest uses average rather than marginal energy prices, in contrast to industries, and marks this for a later version.

50. We can see this in the `data.gdx` file by looking at the set i, the object `qY_CET[out,i,t]` and the Energy Balance entry, (rows 16, 49 and 128 in the file when we order them in column 1). Inside the

model the same mapping is the dummy `d1pY_CET[e,i,t]`, which is set from the loaded values of `pY_CET` in `06_energy_markets.gms` and records which industries produce which energy good.

51. The code comment on the second block states this: "can run separately of bottom-up emissions. When linking 1 and 2 (through 3) calibration variables in 2 are endogenised to match bottom-up emissions". The linking equation is `uEmmE[em,d,t]$(not CO2e[em]).. qEmmE[em,d,t] =E= sum((e,es), qEmmE_BU[em,es,e,d,t])`, and since it is paired with the coefficient rather than with the aggregate, the top-down emission intensity is whatever reconciles the two layers. All three blocks are in `model main` in the assembled model.

52. `qEmmE_BU[em,es,e,d,t] =E= uEmmE_BU[em,es,e,d,t] * qEpj[es,e,d,t]`, with the calibration swap `uEmmE_BU[em,es,e,d,t1], -qEmmE_BU[em,es,e,d,t1]` for every gas other than the aggregate, and the dummy `d1EmmE_BU[em,es,e,d,t] = yes$(qEmmE_BU.l[em,es,e,d,t])` set from the loaded data. One qualification: "held constant afterwards" describes the pure CGE model. When the energy technology model is integrated, `model/dummies_new_energy_use.gms` overwrites `uEmmE_BU.l` for the cells that new technologies open up after `t1`, so the coefficient is not constant in those runs.

53. `qEmmE_BU[em,es,e,d,t] =E= sBioNatGas[t] * uEmmE_BU * qEpj` for the biogenic gas and `(1-sBioNatGas[t])` for the fossil one, with `sBioNatGas[t]` loaded from data and described as "share of bio-natural gas in total natural gas consumption, should perhaps be modelled through emission coefficient in future".

54. `GWP.l['CO2ubio'] = 1`, `GWP.l['CH4'] = 28`, `GWP.l['N2O'] = 265`, and `GWP.l['HFC'] = GWP.l['PFC'] = GWP.l['SF6'] = 1` with the comment that these gases are already in CO_2e in Danish data.

55. `qEmmxE[em,i,t] =E= uEmmxE[em,i,t] * sum(pf_top, qProd[pf_top,i,t])` for industries, and `qEmmxE[em,d,t] =E= uEmmxE[em,d,t]` for demand components that are not industries, which makes those flows purely exogenous. The calibration swap is `uEmmxE[em,d,t1], -qEmmxE[em,d,t1]`.

56. `qEmmTot[em,em_accounts,t]` in `06_emissions.gms`, with `em_accounts` having the four elements `GNA`, `gna_lulucf`, `UNFCCC` and `unfccc_lulucf` declared in `model/sets/emissions.sets.gms`. The inline comment on the border trade line reads "border trade is added for GNA-categories", which describes the effect rather than the operation: the term is subtracted, and it is subtracted only for the UNFCCC categories, which is the same thing.

57. `qEmmInternationalAviation[em,t] =E= sum(i_international_aviation, qEmmE_BU[em,'transport','jet petroleum', i_international_aviation,t])` with `i_international_aviation /51009/`; `qEmmBunkering[em,t] =E= sum((i,es,eBunkering), qEmmE_BU[em,es,eBunkering,i,t])`; and `qEmmOtherDifferencesShips[em,t] =E= qEmmE_BU[em,'transport','diesel for transport','49509',t]`. Border trade is exogenous and loaded from data.

58. The active test is `ABORT$(abs(test_qEmmCO2eData[t] - qEmmTot.l['co2e','GNA',t]) > 0.1)` for the last data year. The disabled one uses a hand-entered table `testDKemm2020[em_accounts]` taken from Statistics Denmark's website, with 42573 for UNFCCC and 81337 for the green national accounts in 2020, plus 1292 for LULUCF.

A CES Trees and Smart Indexing

Constant Elasticity of Substitution (CES) functions are widely used as aggregation tools in Computable General Equilibrium (CGE) models as they allow for a flexible treatment of many different objects. In the models we look at here, they are used in both the production side of the economy aggregating multiple inputs, as well as in the consumption side of the economy by aggregating diverse goods. This aggregation is organised by pairing objects within a level and adding a level for every additional object included in the set-up.

This appendix collects the algebra that Chapters 2 and 3 use without deriving, and then explains how the whole of a tree is written as a single group of equations in the code. The second part is the more useful one for a reader working with the model, because the indexing device is not obvious on a first reading and it is used in every module that contains a nest.

A.1 CES algebra within a level

We use a production example. The firm's optimal demand for inputs is obtained from solving a sequence of cost minimisation problems at every level in the CES tree. At each level we have output Q and two inputs (x_1, x_2) with prices (p_1, p_2). Output is given by

$$\text{Output} \quad \Rightarrow \quad Q = \left[\mu_1^{\frac{1}{\eta}} (x_1)^{\frac{\eta-1}{\eta}} + \mu_2^{\frac{1}{\eta}} (x_2)^{\frac{\eta-1}{\eta}} \right]^{\frac{\eta}{\eta-1}}$$

and we seek to minimise total cost, $\mathscr{P}Q = p_1 x_1 + p_2 x_2$, by the optimal choice of (x_1, x_2).

The solution to this problem is well known and yields the following objects:

$$\text{Derivative} \quad \Rightarrow \quad \frac{\partial Q}{\partial x_i} = \left[\mu_i \frac{Q}{x_i} \right]^{\frac{1}{\eta}}$$

$$\text{Demand/F.O.C.} \quad \Rightarrow \quad x_i = \mu_i Q \left(\frac{\mathscr{P}}{p_i} \right)^{\eta}$$

$$\text{CES Price} \quad \Rightarrow \quad \mathscr{P} = \left[\mu_1 p_1^{1-\eta} + \mu_2 p_2^{1-\eta} \right]^{\frac{1}{1-\eta}}$$

The μ_i are share parameters and η is the elasticity of substitution between the two inputs. The input prices p_i are general user costs of input i. ***One detail to mention*** is that, although we display it here, the production function is never used in the numerical solution to the problem. Much like the utility function in the household problem only the derivatives are needed, and they enter the problem through the demand functions (the derivatives are not themselves used). The problem is solved using these demand functions and the zero profit expression (total cost constraint).[1]

We can also write the production function to include an explicit technology level object A as

$$Q = A\left[\gamma_1^{\frac{1}{\eta}}(x_1)^{\frac{\eta-1}{\eta}} + \gamma_2^{\frac{1}{\eta}}(x_2)^{\frac{\eta-1}{\eta}}\right]^{\frac{\eta}{\eta-1}}$$

which maps into the above expressions by setting $\gamma_2 = 1-\gamma_1$, and $\mu_1 = \gamma_1 A^{\eta-1}$ and $\mu_2 = (1-\gamma_1)A^{\eta-1}$. The model works with μ and never separates the two, so a change in technology at a node and a change in the composition of that node are the same object in the code. Technical progress that is specific to one input has to be imposed as a path for the corresponding μ.

The CES price is implied, not imposed

There is one more consequence of solving the problem with demands and the cost constraint only, and it is worth spelling out because it explains several features of the code. The CES price displayed above is never written anywhere. It does not have to be. Substitute the demand functions into the cost constraint,

$$\mathscr{P}Q = \sum_i p_i \mu_i Q\left(\frac{\mathscr{P}}{p_i}\right)^{\eta} = \mathscr{P}^{\eta} Q \sum_i \mu_i p_i^{1-\eta} \qquad \Longrightarrow \qquad \mathscr{P}^{1-\eta} = \sum_i \mu_i p_i^{1-\eta}$$

and the CES price index comes back out. The cost constraint is therefore the equation that determines the price of the composite, and in the code it is paired with exactly that variable.[2]

Two practical points follow. First, the limit cases need no special treatment, which is the subject of the next subsection. Second, an elasticity of exactly one is a singular point of this formulation. At $\eta = 1$ the substitution above collapses to $\mathscr{P}Q = \mathscr{P}Q\sum_i \mu_i$, which either has no solution or leaves $\mathscr{P}$ undetermined, so the equation can no longer deliver the nest price. The elasticities used in the consumption tree are 0.94, 1.04, 1.1 and 1.25 at the nests where substitution is close to unitary: near one on either side, but never at it.

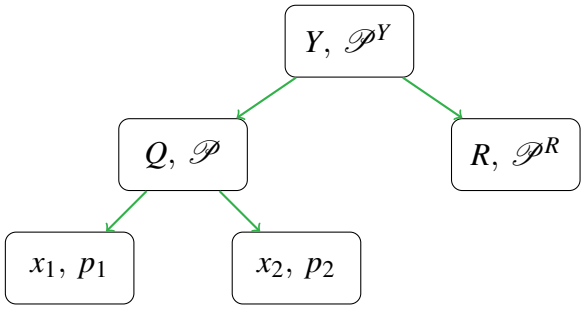
Leontief

It is easy to show that this math applies to the limit case of the Leontief function ($\eta = 0$). Since we only use the demand functions and the total cost constraint we have $x_i = \mu_i Q\left(\frac{\mathscr{P}}{p_1}\right)^0 = \mu_i Q$ and $\mathscr{P}Q = p_1 x_1 + p_2 x_2$. Replacing the demand expressions in the constraint we obtain the correct measure of the CES price $\mathscr{P} = \mu_1 p_1 + \mu_2 p_2$.

So a Leontief node is a CES node with the elasticity set to zero, and no code has to change. The same is true of the other direction: as η grows the node approaches perfect substitutability, and nothing in the equations breaks.

A.2 Multiple levels

We now move up one tree level by aggregating Q and another input R. The character $\mathscr{F}$ here denotes the CES function.



$$Q = \mathscr{F}(x_1, x_2) \qquad Y = \mathscr{F}(Q, R)$$

The total cost expression is now $\mathscr{P}^Y Y = \mathscr{P} Q + \mathscr{P}^R R$. The algebra repeats for every level of the tree. Importantly, all prices apart from those of the original inputs at the bottom of the tree are CES prices, that is, CES aggregator prices or user costs of the composite object. Only the prices at the bottom are determined by markets. Even those can be user cost measures. The wage is only one component of the unit cost of labour.

Quantities flow down, prices flow up

A tree needs exactly two things from outside itself, and it is useful to know which two, because it tells the reader where to look when a nest behaves unexpectedly.

The quantity at the top comes from outside. In the consumption tree it is the value of aggregate consumption delivered by the consumption function of Chapter 2. In the production tree it is the output of the industry, delivered by the market clearing condition of Chapter 6.[3] Given that quantity and the prices, the demand functions push quantities down the tree, one level at a time, until the leaves are reached.

The prices at the bottom come from outside. In the consumption tree each leaf price is the purchaser price of the corresponding demand component in the input-output system. In the production tree the leaves are intermediate inputs, three types of capital, labour and three energy nodes, and each carries a price built elsewhere: a composite input-output price, a user cost of capital, the wage, and a marginal energy price. The four production links, and the consumption counterpart in `02_consumption_disaggregated.gms`, are

```
.. pProd[RxE,i,t] =E= pD[i,t] + jpProd[Rxe,i,t];
.. pProd[labour,i,t] =E= pL_i[i,t];
.. pProd[pf_bottom_capital,i,t]
     =E= sum(sameas[pf_bottom_capital,k],
```

```
                pK_k_i[k,i,t]/pK_k_i[k,i,tBase]);
.. pProd[pf_bottom_e,i,t]
     =E= sum(pf_bottom_e2re[pf_bottom_e,re], pE_re_i[re,i,t])
         + jpProd[pf_bottom_e,i,t];

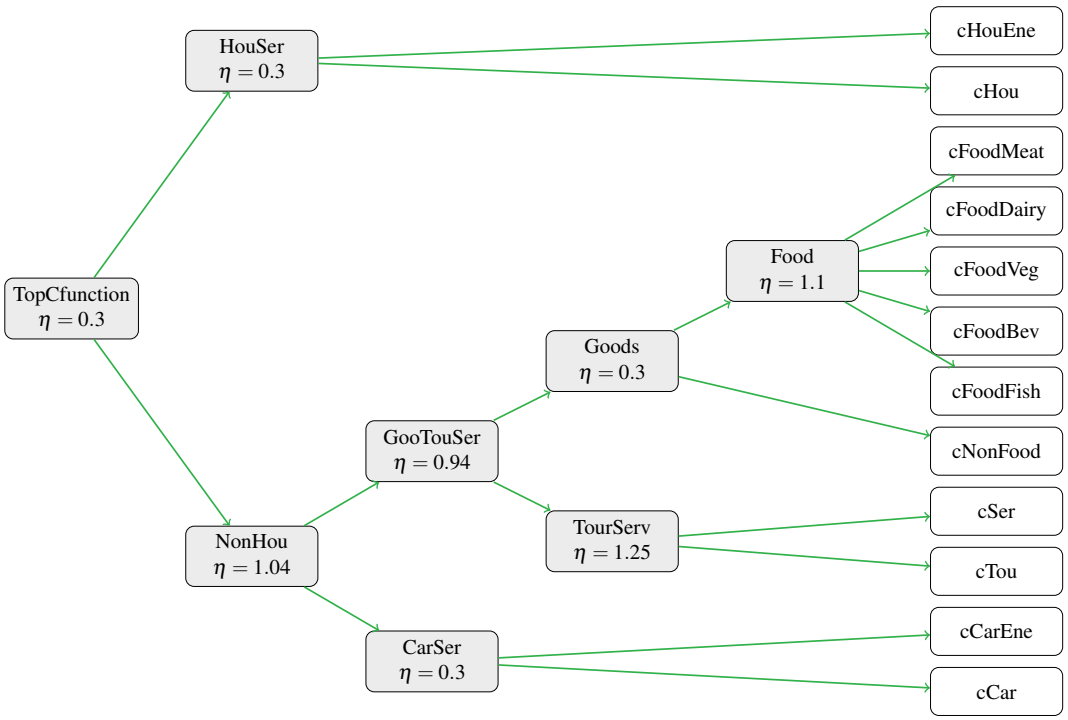
pCHh&_cf_bottom[cf_bottom,t]..
  pCHh[cf_bottom,t] =E= sum(c$c2cf_bottom_mapping[c,cf_bottom], pD[c,t]);
```

where the capital price is divided by its base-year value, so that the capital leaf price is one in the base year and the quantity carries the units. Given the leaf prices, the cost constraints push prices up the tree, one level at a time, until the top is reached, where the price of the composite is the unit cost of the whole bundle.

Everything else in a tree is internal. That is why a nest can be inspected on its own: give it a top quantity and a set of leaf prices and it is a closed system.

A.3 GAMS code. Smart indexing.

The use of CES trees is enhanced by the smart indexing capability of GAMS. This allows us to express the entire tree in a single expression. This particular coding feature can be hard to grasp when someone is first introduced to the code so we detail it here. Consider the consumption tree of Chapter 2, drawn here with the names the code uses rather than with descriptive labels, and with the substitution elasticity of each node.



The entire tree is captured in a single group of equations. These are the zero profit condition at every point in the tree, $\mathscr{P}Q = p_1x_1 + p_2x_2$, and the demand curves

$x_i = \mu_i Q\,(\mathscr{P}/p_i)^{\eta}$. In the code they will be written with an extra index which marks the tree node n this set of equations lies in.[4] The exact way the code is written requires interpretation.

One set for leaves and nests

Before the equations, one observation about the sets, because it is what makes the single expression possible. A node in the middle of a tree is two things at once: it is an input to its parent and it is an aggregate of its children. It therefore has to appear on both sides of the algebra, and the code arranges this by putting the leaves and the nests into *one* set. The leaves and the nests are then subsets of it.[5] The tree itself is then a mapping from a nest to its children, which is nothing more than the edge list of the tree written as a GAMS set.[6]

The demand equation

The demand equation is written as

$$x_i \;=\; \mu_i \;\times\; \sum_{\mathscr{F}_{i,n}} \left\{ Q_n \left(\frac{p_i}{\mathscr{P}_n} \right)^{-\eta_n} \right\}$$

This must be understood as follows: the i index runs through all objects in the tree, so that it characterises all demand curves.[7] The left hand side pins down a single element i. This means it solves the equations for every i but computes each i equation separately. But the right hand side object is a sum that sets most elements to zero except the exact pairs (i,n) which are described in an index mapping $\mathscr{F}_{i,n}$ because each element in i is only compatible with a certain node n in the tree.[8] So in this exact case i is fixed and the sum is over n. But there is only one non-zero element as we loop over all $n|i$ possibilities.

That last sentence is the whole device, and it is worth restating in the other direction. A child has exactly one parent. So a sum over the mapping with the child held fixed is not really a sum: it is a lookup, and it returns the one parent of that child. The sum notation is used because GAMS has no other way of writing "the element of n that this i belongs to".

The zero profit condition, and the same sum read the other way

Since the left hand side selects a single element of i, the sum on the right hand side also selects a single element in the set (i,n). For example, if $n = Goods$, then $i|n \in (Food, cNonFood)$ and $\mathscr{F}_{i,n}$ is the mapping that binds exactly these nodes together. The zero profit condition is then written

$$\mathscr{P}_n Q_n = \sum_{\mathscr{F}_{i,n}} \{p_i x_i\}$$

and here the sum on the right hand side adds both elements i which belong to the selected node n.

The two equations use the same mapping and the same sum, and they mean two different things. The difference is which index the equation's own domain has already fixed. In the demand equation the domain is the child, so the sum walks *up* and finds one element. In the zero profit condition the domain is the nest, so the sum walks *down* and adds over all of them. Nothing in the syntax announces which of the two is happening; the reader has to look at the left hand side. This is the single most confusing thing about the notation and, once seen, the least confusing thing about the code.

One generalisation of the example above. The tree is not binary. The number of children a nest has is whatever the mapping says, and the algebra of Section A.1 extends to any number of them by replacing the two-term sums with sums over the children. In the consumption tree seven nests have two children and the food nest has five.

The production tree carries an extra index

The production version of the same three equations is identical except that every object carries an industry index, and so does the mapping. Written out, the three equations of a production nest are the price ratio, the demand and the cost constraint:

```
pProd2pNest[pf,pfNest,i,t]$(pf_mapping[pfNest,pf,i])..
  pProd2pNest[pf,pfNest,i,t] =E= pProd[pf,i,t] / pProd[pfNest,i,t];

qProd[pf,i,t]$(not pf_top[pf])..
  qProd[pf,i,t] =E= uProd[pf,i,t]
                  * sum(pf_mapping[pfNest,pf,i],
                      pProd2pNest[pf,pfNest,i,t]**(-eProd[pfNest,i])
                        * qProd[pfNest,i,t]
                  );

.. pProd[pfNest,i,t] * qProd[pfNest,i,t]
     =E= sum(pf_mapping[pfNest,pf,i], pProd[pf,i,t] * qProd[pf,i,t]);
```

Three things are worth noticing. The price ratio is given its own variable and its own equation rather than being written inline; this keeps the exponent operating on a single symbol and gives the solver a well-scaled object. The elasticity is indexed by the nest, so each node has its own, and it is also indexed by the industry, so in principle each industry could have its own; in practice a single value of 0.7 is assigned to every nest of every industry.[9] And the mapping itself carries the industry index, which means the shape of the tree could differ across industries. It does not: every row of `pf_mapping` is declared over `set.i_standard`, and that set has been given all of $[i]$, so all industries have the same tree.[10]

Appendix	Consumption tree	Production tree
node index i	`cf`	`pf`
nest index n	`cfNest`	`pfNest`
leaves	`cf_bottom` (the 12 elements of $[c]$)	`pf_bottom`
root	`cf_top` = `TopCfunction`	`pf_top` = `TopPfunction`
edge list $\mathscr{F}_{i,n}$	`cf_mapping[cfNest,cf]`	`pf_mapping[pfNest,pf,i]`
quantity x_i, Q_n	`qCHh[cf,t]`	`qProd[pf,i,t]`
price p_i, $\mathscr{P}_n$	`pCHh[cf,t]`	`pProd[pf,i,t]`
price ratio $p_i/\mathscr{P}_n$	`pCHh2pNest[cf,cfNest,t]`	`pProd2pNest[pf,pfNest,i,t]`
share μ_i	`uCHh[cf,t]`	`uProd[pf,i,t]`
elasticity η_n	`eCHh[cfNest]`	`eProd[pfNest,i]`
module	`02_consumption_disaggregated.gms`	`03_production.gms`
sets	`sets/households.sets.gms`	`sets/production.sets.gms`

Table A.3.1: The two main trees of the model and the correspondence with the notation of this appendix. The energy nests of Chapters 2 and 3 hang below the leaves `cHouEne`, `cCarEne` and the three `pf_bottom_e` nodes, and are written with the same algebra but without a mapping set; see Section A.4.

A.4 The same algebra elsewhere in the model

Once the pattern is recognised it turns up in most modules, sometimes with the mapping set and sometimes written out inline, and in two places with a twist that is worth understanding. This section is a reading guide.

Nests without a mapping set

The energy demand nests are written with the same three equations, but the hierarchy is short and fixed, so instead of a mapping set the code writes the parent explicitly in each equation. The household energy nest and the industry energy nests are both of this form.[11] Reading them is easier than reading the main trees, precisely because the parent is visible.

Imports: the same condition written as a ratio

The choice between domestic and imported supply in each input-output cell is the same first-order condition, but the code writes it as a ratio of the two demands rather than as two demands relative to a composite:

$$\frac{M_{i,d}}{Y_{i,d}} = \frac{r^{M0}_{i,d}}{1 - r^{M0}_{i,d}} \left(\frac{p^{Y}_{i,d}}{p^{M}_{i,d}} \right)^{\eta^{M}}$$

This is algebraically equivalent to a two-input nest with shares r^{M0} and $1 - r^{M0}$, and it has the advantage that no composite price has to be defined for the cell.[12] The ratio form is what one uses when the nest has exactly two children and the composite is not needed elsewhere.

Exports: a nest with an exogenous parent

The export demand functions have the same functional form with the market size playing the role of $\mu_i Q_n$ and the foreign price playing the role of the composite price $\mathscr{P}_n$. Since both are exogenous, an export equation is a single branch of a nest whose parent lies outside the model.[13] This is why the export elasticity is the parameter that carries so much of the model's behaviour: it is the one demand elasticity in the model whose parent quantity does not shrink when the price rises.

The CET split: CES with the sign reversed

The multi-output split of Chapter 3 is the same equation with the sign of the exponent flipped:

$$Q_{o,i} = \mu_{o,i} \left(\frac{p^0_{o,i}}{p^0_i} \right)^{+\varepsilon_i} Q^0_i \qquad \text{against} \qquad x_i = \mu_i \left(\frac{p_i}{\mathscr{P}_n} \right)^{-\eta_n} Q_n$$

A higher relative price raises the quantity instead of lowering it, which turns substitution into transformation: the node is now a way of splitting one input into several outputs rather than combining several inputs into one output.[14]

Everything in Section A.1 then carries over with η replaced by $-\varepsilon$. In particular the same substitution into the adding-up condition delivers the implied price of the composite,

$$\left(p^0_i\right)^{1+\varepsilon} = \sum_o \mu_{o,i} \left(p^0_{o,i}\right)^{1+\varepsilon}$$

so the revenue price of an industry is a CET aggregate of its product prices, and again it is never written out: the adding-up condition determines it.

The energy distributor: shares that conserve quantity

The one construction that is genuinely a different functional form is the energy distributor of Chapter 6. It allocates total demand for an energy carrier across suppliers as

$$Q^e_i = \frac{s_i \, p_i^{-\sigma}}{\sum_j s_j \, p_j^{-\sigma}} \, D^e$$

which looks like a CES share but is not the same object. Summing over suppliers gives $\sum_i Q^e_i = D^e$ exactly: the construction conserves *quantity*. The CES nests of this appendix conserve *value*, since their adding-up condition is $\mathscr{P}_n Q_n = \sum_i p_i x_i$, and they conserve quantity only in the base year where all prices are one.[15] The difference is deliberate and correct: energy is measured in petajoules, and petajoules delivered must equal petajoules supplied whatever the prices do. A value-conserving nest would not have that property, and the accounting of Chapter 6 depends on it.

A.5 Calibration and numerical notes

Four practical points close the appendix.

The shares are calibrated, not estimated

At each node the share parameters are recovered from the data by inverting the demand function. Given observed quantities and prices in the calibration year and an assumed elasticity,

$$\mu_i = \frac{x_i}{Q_n}\left(\frac{p_i}{\mathscr{P}_n}\right)^{\eta_n}$$

and this is done in the code by swapping the observed quantity out of the endogenous group and the share in.[16] The consequence is the usual one: the tree reproduces the base-year data exactly by construction, and the elasticities, which are the only free parameters, carry the entire behavioural content. Nothing about the fit of the tree to the data tells the reader anything about whether the elasticities are right.

Why the shares are extrapolated flat

After the last data year the shares are held constant, which sounds like a strong assumption and is in fact the neutral one.[17] The reason is visible in the formula above. Both terms are ratios: a quantity over a quantity and a price over a price. Under balanced growth the numerator and the denominator of each grow at the same rate, so μ_i has no trend, and a constant is what a trendless object should be extrapolated as. Any drift in the composition of a nest has to be imposed deliberately, as a path for μ, and this is how technical progress that favours one input is introduced.

The same observation explains a detail of the growth adjustment. Since the demand equation contains only ratios of quantities and ratios of prices, it is invariant to the detrending of Chapter 1: it holds in exactly the same form whether it is written in levels or in growth-adjusted variables. The share parameters are correspondingly left out of the adjustment, and so are the price ratio variables, which the code excludes by a rule on their names.[18]

Prices are one in the base year, so quantities are values

At calibration the nest prices are normalised to one in the base year, and the top price of the production tree is normalised by an explicit scaling variable.[19] The practical implication is that every quantity inside a tree is measured in base-year prices, that is, it is a value at constant prices rather than a physical quantity. This is what makes the numbers comparable across nodes, and it is also why the capital leaf divides the user cost by its base-year value and lets the quantity absorb the units.

What to check when a nest misbehaves

Finally, a short list, since the same three causes account for most problems in a nest. An elasticity too close to one makes the cost constraint ill-conditioned, for the reason given in Section A.1. A node whose observed value is zero in the data has no share to calibrate, and the model handles this by dummying the node out rather than by setting the share to zero; the dummy is what decides whether a branch exists at all.[20] And a leaf price that has been given the wrong wedge — an average energy price where a marginal one belongs, or a user cost that has not been normalised — shows up as a share parameter far from the observed expenditure share, which is the easiest symptom to look for.

Notes

1. The claim can be checked directly. The only place in the model where the CES function itself is written out is a test in `03_production.gms` that recomputes `qProd[pfNest,i,t]` from the shares and the input quantities and aborts on a discrepancy. The test is commented out. Nothing in the solved model refers to it.

2. In `02_consumption_disaggregated.gms` the equation is `.. pCHh[cfNest,t] * qCHh[cfNest,t] =E= sum(cf_mapping[cfNest,cf], pCHh[cf,t] * qCHh[cf,t])`. With no name in front of the `..`, gamY pairs the equation with the variable on the left hand side, which is the nest price. The production version in `03_production.gms` is the same equation with an industry index added.

3. `qCHh[cf,t] * pCHh[cf,t] =E= vC[t]` for `cf_top` in `02_consumption_disaggregated.gms`, and `qY0_i[i,t] =E= qPFtop2qY[i,t] * qY_i[i,t]` in `03_production.gms`. Note that the top of the production tree is output gross of installation costs: `qProd[pf_top,i,t] =E= qY0_i[i,t] + sum(k, qInstCost_k_i[k,i,t])`.

4. In the code we write $i \equiv cf$ and $n \equiv cfNest$ for consumption, and $i \equiv pf$ and $n \equiv pfNest$ for production.

5. `set cf "Inputs and their nests in consumption function" / set.c set.consumption_nests /` in `model/sets/households.sets.gms`, with `cf_bottom[cf] /set.c/`, `cfNest[cf] /set.consumption_nests/` and `cf_top[cf] /TopCfunction/`. The production side is identical: `set pf / set.factors_of_production set.production_nests /` in `model/sets/production.sets.gms`, where `factors_of_production` is loaded from `data.gdx`. So $[cf]$ has 12 leaves and 8 nests, and the production tree has 7 nests and eight leaves in the mapping: the three capital types `im`, `it`, `ib`, the three energy nodes, `labor` and `RxE`.

6. `cf_mapping[cfNest, cf]` in `households.sets.gms` and `pf_mapping[pfNest,pf,i]` in `production.sets.gms`. The consumption mapping has eight rows, one per nest; the production mapping has seven, and carries an industry index discussed below.

7. Therefore i covers HouSer, NonHou, cHouEne, cHou, GooTouSer, CarSer, cCarEne, cCar, Goods, TourServ, and so on: every node except the top, which has no parent and therefore no demand curve. The code writes the exclusion as `qCHh[cf,t]$(not cf_top[cf])`.

8. This mapping $\mathscr{F}$ is called `cf_mapping`.

9. `eProd.l[pfNest,i] = 0.7` in `03_production.gms`. The consumption tree is the opposite case: `eCHh[cfNest]` has no second index and takes eight different values, the ones shown in the figure above.

10. `set i_standard[i] /set.i/` in `production.sets.gms`, with the earlier version that excluded industries `35002` and `19000` commented out just above it. Chapter 3 discusses this.

11. `qEpj[es,e,c,t] =E= uChh_Epj[es,e,c,t] * sum(cf_ene$es2cf2d(es,cf_ene, c), pEpj2pChh[es,e,c,cf_ene,t]**(-eChh_cEne[cf_ene]) * qChh[cf_ene,t])` in `02_consumption_disaggregated_energy.gms`, and `qEpj[es,e,i,t] =E= uEpj[es,e,i,t] * (pEpj_marg[es,e,i,t]/pREes[es,i,t])**(-eEpj[es,i]) * qREes[es,i,t]` in `03_production_CES_energydemand.gms`, where the parent node `pREes[es,i,t]` is named rather than looked up. The elasticities are `eChh_cEne` of 0.5 for households and `eEpj`, `eREes` of 0.1 for industries, so fuel switching within an end purpose is much harder for a firm than for a household. The household version is also the one exception to the rule that the leaf price is a marginal price: it uses the average energy price, and the code marks this for a later version.

12. `qM_i_d[i,d,t] * (1-rM0[i,d,t]) =E= qY_i_d[i,d,t] * rM0[i,d,t] * pY2pM_i_d[i,d,t]**eM_i_d[i,d]` in `05_imports.gms`, with $\eta^M = 2$. Chapter 5 describes it. No composite price for the cell is built anywhere: once the ratio fixes the split, `06_input_output.gms` works with the two quantities `qY_i_d` and `qM_i_d` and their own prices, and the value of the cell is their sum.

13. `qD[x,t] =E= qXMarket[x,t] * pX2pRoW_x[x,t]**(-eX[x])` in `05_exports.gms` and `qEpj[es,e,xEne,t] =E= uEpj_xEne[es,e,t] * (pEpj_marg[es,e,xEne,t]/pEpj_foreign[es,e,t]) **(-eX_ene[e])` in `05_exports_energy.gms`, both with an elasticity of five. Chapter 5 describes them.

14. `qY_CETgross[out,i,t] =E= uY_CET[out,i,t] * (pY0_CET[out,i,t]/pY0_i[i,t])**eCET[i] * qY0_i[i,t]` in `03_production_CET.gms`, with `eCET.l[i] = 5`, alongside the adding-up condition `pY0_i[i,t] * qY0_i[i,t] =E= sum(out, pY0_CET[out,i,t] * qY_CETgross[out,i,t])`.

15. `qY_CET[e,i,t] * (sum(i_a, sY_Dist * pY_CET** (-eDist)) + sum(i_a, sM_Dist * pM_CET**(-eDist))) =E= sY_Dist[e,i,t] * pY_CET[e,i,t]**(-eDist[e]) * qEtot[e,t]` in `06_energy_markets.gms`, with `eDist.l[e] = 5`. Writing the denominator explicitly rather than as a composite price is what makes the difference: a CES nest would carry $\mathscr{P}^{\sigma}$ with $\mathscr{P}^{1-\sigma} = \sum s_j p_j^{1-\sigma}$, and the exponent on the weights would differ.

16. `-qCHh[cf,t1]$(not cf_top[cf]), uCHh[cf,t1]$(not cf_top[cf])` in `02_consumption_disaggregated.gms`, and `uProd[pf_bottom,i,t1], -qProd[pf_bottom,i,t1]` together with `uProd[pfNest,i,t1]$(not pf_top[pfNest]), -pProd[pfNest,i,t1]$(not pf_top[pfNest])` in `03_production.gms`. So on the production side the leaf shares are recovered against observed quantities and the nest shares against observed nest prices.

17. `$GROUP+ G_flat_after_last_data_year uCHh[cf, t]$(not cf_top[cf])` and `$Group+ G_flat_after_last_data_year uProd, qPFtop2qY`.

18. `variable_groups.gms` assigns variables to the inflation, growth and value groups by prefix, but

excludes any name containing `2p` from the price group and any name containing `2q` from the quantity group. That is what keeps `pCHh2pNest`, `pProd2pNest`, `pY2pM_i_d`, `pX2pRoW_x`, `qR2qY_i`, `qL2qY_i` and `qPFtop2qY` undeflated, which is correct because each of them is a ratio. Variables whose prefix is `u` are in none of the three groups, so the shares are undeflated as well.

19. `pProd.l[pfNest,i,tDataEnd] = 1`, `pY0_i.l[i,tDataEnd] = 1` and the swap `qPFtop2qY[i,tBase], -pProd[pf_top,i,tBase]` with the comment “normalise price at 1”.

20. `d1Prod[pf,i,t] = yes$(qProd.l[pf,i,t])` in `03_production.gms`. So a branch exists where the data records a flow, and the same convention is used for the energy nests through `d1pEpj`. Chapter 1 describes the dummy mechanism.

References

Published literature

Anderson, James E., 1979. "A Theoretical Foundation for the Gravity Equation," *American Economic Review*, 69(1), 106–116.

Anderson, James E., and Eric van Wincoop, 2003. "Gravity with Gravitas: A Solution to the Border Puzzle," *American Economic Review*, 93(1), 170–192.

Armington, P. S., 1969. "A Theory of Demand for Products Distinguished by Place of Production," *International Monetary Fund Staff Papers*, 16, 159–178.

Blanchard, O., 2018. "On the future of macroeconomic models," *Oxford Review of Economic Policy*, 34(1–2), Spring–Summer 2018, 43–54.

Blanchard, O., and Galí, J., 2008. "Labor Markets and Monetary Policy: A New-Keynesian Model with Unemployment," NBER Working Paper 13897, National Bureau of Economic Research, Inc.

Campbell, John Y., and Mankiw, N. Gregory, 1989. "Consumption, Income and Interest Rates: Reinterpreting the Time Series Evidence," *NBER Macroeconomics Annual 1989*, Volume 4, 185–246.

Coibion, O., and Gorodnichenko, Y., 2013. "Is the Phillips Curve alive and well after all?" NBER Working Paper 19598, October 2013.

Dennis, R., 2009. "Consumption habits in a new Keynesian business cycle model," *Journal of Money, Credit and Banking*.

Eaton, J., and Kortum, S., 1999. "International Technology Diffusion: Theory and Measurement," *International Economic Review*, 40(3), 537–570.

Eaton, J., and Kortum, S., 2002. "Technology, Geography, and Trade," *Econometrica*, 70(5), 1741–1779.

Galí, J., Smets, F., and Wouters, R., 2012. "Unemployment in an Estimated New Keynesian Model," *NBER Macroeconomics Annual*, 26, 329–360.

Gourinchas, P., and Parker, J., 2002. "Consumption Over the Life Cycle," *Econometrica*, 70, 47–89.

Greenwood, J., Hercowitz, Z., and Huffman, Gregory W., 1988. "Investment, capacity utilization, and the real business cycle," *American Economic Review*, 78(3), 402–17.

Kocherlakota, N., 2010. "Modern Macroeconomic Models as Tools for Economic Policy," Federal Reserve Bank of Minneapolis, Annual Report Essay, May 4, 2010.

Kronborg, A., Poulsen, K., and Kastrup, C., 2020. "Estimering af udenrigshandelselasticiteter i MAKRO," Working paper, DREAM.

Krugman, Paul R., 1979. "Increasing returns, monopolistic competition, and interna-

tional trade," *Journal of International Economics*, 9(4), 469–479, November.

Krugman, P., 1980. "Scale Economies, Product Differentiation, and the Pattern of Trade," *American Economic Review*, 70(5), 950–959.

Phillips, A., 1958. "The Relation between Unemployment and the Rate of Change of Money Wage Rates in the United Kingdom, 1861–1957," *Economica*.

Ravenna, F., and Walsh, C., 2008. "Vacancies, unemployment, and the Phillips curve," *European Economic Review*, 52(8), 1494–1521.

Rebelo, S., 2005. "Real Business Cycle Models: Past, Present, and Future," *The Scandinavian Journal of Economics*, 107(2), 217–238.

DREAM technical notes on GreenREFORM-EU and GreenREFORM

Stewart, Louis, Stephensen, Peter, and Kirk, Jens Sand, June 2025. "Documentation of the Abatement Model of the Green Reform EU Model."

Stephensen, Peter, November 2024. "Including bottom-up energy data in a Computable General Equilibrium Model."

Stephensen, Peter, Berg, Rasmus, and Beck, Ulrik, August 2020. "End-of-pipe emissions abatement technologies in a CGE model."

Stephensen, Peter, and Kirk, Jens Sand, May 2020. "Energy good markets in GreenREFORM."

Kirk, Jens Sand, Stephensen, Peter, and Berg, Rasmus, May 26, 2020. "Strategy for integrating the energy sector model and the CGE model in Green Reform."