Resolution of HANK models in discrete and continuous time

Sofia Ruiz Romanos
Université Paris-Dauphine – PSL
Supervisor: Prof. Fabien Tripier

12/11/2025

Acknowledgements

I would like to express my gratitude to my supervisor, Fabien Tripier, for his guidance and support throughout this thesis. I am also deeply thankful to my professor Selma Malmberg for introducing me to heterogeneous-agent models and for the time and dedication she has given both in class and during the development of this work. I also wish to thank the teachers and students of the Quantitative Economics Master's program at Université Paris-Dauphine, who have accompanied me over the past three years.

I am profoundly grateful to my friends and family for their constant encouragement during this project. In particular, I would like to thank Bob, Salma, and Lucas for their advice and companionship during many long working sessions on our respective theses. Finally, I owe special thanks to Artur, whose feedback, technical support, and help with Python were invaluable, especially in figuring out virtual environments, developing the brute-force calibration algorithm, and for his constant enthusiasm throughout the entire process.

Abstract

This master thesis introduces the different steps needed to replicate and adapt the continuous-time HANK model with energy shortages of Pieroni (2023) into a discrete-time framework using the Sequence-Space Jacobian method. The model is implemented in Python with the SSJ toolkit. The calibration parameters are adjusted manually and with the help of a brute-force search algorithm to match steady-state statistics. The discrete-time model reproduces the qualitative behavior of aggregate variables and prices, but quantitative responses to shocks are larger and more volatile than in the continuous-time benchmark. In contrast, the nonlinear solver fails to converge. These results suggest that while the discrete-time Sequence-Space Jacobian framework offers simplicity and flexibility compared to the continuous-time model, significant work remains to be done to achieve a reliable transcription of the continuous-time formulation into discrete time. Future research could explore alternative sets of equilibrium unknowns, improved calibrations, or using smaller shock magnitudes to adjust the model.

Contents

1 Introduction

The interest in heterogeneous agent models in macroeconomics has grown quickly over the last decade. On the one hand, economists are becoming increasingly interested in studying the effects of macroeconomic policies on the distribution of inequalities among households (Malmberg et al. (in press), Benmir and Roman (2022)). Simultaneously, a growing number of researchers are also becoming interested in studying how the presence of household inequalities disrupts established policy transmission channels and mechanisms long established in macroeconomics (Luetticke (2021), Kaplan et al. (2018)).

The growing popularity of heterogeneous agent models has been enabled by the increasing number of methodological advances that perfect existing methods for solving heterogeneous agent models and develop new ones, with the goal of reducing algebraic and computational complexity. Two frequently used methods are the continuous-time approach by Achdou et al. (2022) and the discrete-time Sequence Space Jacobian framework developed by Auclert et al. (2021).

The objective of this master thesis is to translate and replicate the continuous-time HANK model with energy shortages of Pieroni (2023) (which uses the Achdou et al. (2022) continous-time method) using the Sequence-Space Jacobian method of Auclert et al. (2021) with the help of the existing SSJ toolkit [1]. More concretely, we will first attempt to replicate the steady-state to reproduce a set of stationary statistics, and then compute the impulse response of aggregate variables to a negative energy supply shock (sections 3 and 4.1 of Pieroni (2023)). The replication aims to preserve the economic structure of Pieroni (2023)'s model while adapting its equations, calibration, and numerical implementation to the discrete-time setting.

The goal of this replication is twofold: first, to develop a methodological framework for transforming continuous-time models with heterogeneous agents into discrete-time models; second, to explore the SSJ toolkit and illustrate how to implement and adapt it to fit a model with household heterogeneity and energy shortages.

The motivation for choosing this model is of particular relevance today. Exploring how energy shortages can affect household inequalities is crucial to developing efficient policies with the goal of minimizing the impact of such a shock. This is especially true since energy dependence between regions has become an important subject in geopolitics, as it can be both a vulnerability and a policy tool. Additionally, the current and upcoming consequences of climate change raise the urgency of developing effective climate mitigation policies, which rely heavily on the retreat from fossil fuels.

This paper can be broken down into different sections. Section 2 presents heterogeneous agent models in more detail and introduces the principles behind the continuous- and discrete-time resolution methods. Section 3 introduces the model created by Pieroni (2023) and highlights all the modeling decisions that

[1] A Python toolkit for solving and simulating HANK models using the Sequence-Space Jacobian method, developed by Bardóczy, Bence and Cai, Michael and Rognlie, Matthew and Auclert, Adrien and Souchier, Martin and Straub, Ludwig: https://github.com/shade-econ/sequence-jacobian.

are specific to continuous-time models. Section 4 introduces the main principles that can be used to transform the model from continuous to discrete time and presents the model's discretized equations. Section 5 briefly explains the code structure used by Pieroni (2023), illustrates how to use the SSJ toolkit to program the model, and discusses the computational advantages and disadvantages of this programming method. Finally, 6 explains the choices behind the calibration used, presents the obtained steady-statistics as well as the results of the model in reaction to a shock of the energy supply, and discusses the limits and caveats of this replication. Section 7 concludes.

2 Literature Review

Heterogeneous agent (HA) models differ from standard dynamic macro models[2] through the characterization of the household block. Most simple heterogeneous agent models (see, for example, Aiyagari (1994); Huggett (1993)) generally follow the following structure:

There is a continuum of households (or agents) that are heterogeneous in their wealth a and their labor endowment e [3]. Each household derives utility from consumption $u(c)$. Each household also faces idiosyncratic risk (risk specific to each person) in their labor endowment e: every period, endowment changes according to certain probabilities over a set of lower and higher endowment states, which can be summarized by a Markov transition matrix. Therefore, households cannot rely on a stable source of income and can suddenly switch from high income to low income, and vice versa. To insure themselves against income risk, households can invest part of today's income into assets a_t instead of consuming it, and receive additional income next period in the form of returns on those assets ra_{t-1}. Agents also face a limit on the minimum amount of assets they can invest in. This borrowing constraint is usually set at zero, which means that households cannot invest in a negative number of assets, or borrow.

Therefore, each household maximizes their value function with respects to consumption c_t and assets a_t, given labor endowment and the return on the assets invested last period ra_{t-1}, for a given return r:

$$\begin{cases} V(a_{t-1}, e_t) = \max\limits_{c_t, a_t}(u(c_t) + \beta\, \mathbb{E}_t[V(a_t, e_{t+1})]) \\ \text{s.t.} \quad c_t + a_t = (1 + r_t)a_{t-1} + e_t \\ \text{s.t.} \quad a_t \geq \underline{\mathrm{a}} \end{cases} \tag{1}$$

We will refer to the optimal choice of c_t and a_t given r_t, e_t and a_{t-1} as policy functions. Using the optimal policy function to obtain the distribution of households $f_t(a, e)$ across the set of possible values of c and a, allows us to compute total consumption and asset C_t and A_t:

$$C_t = \int_e \int_a c_t(a, e)\, f_t(a, e)\, da\, de, \qquad A_t = \int_e \int_a a_t(a, e)\, f_t(a, e)\, da\, de. \tag{2}$$

The model features incomplete markets, meaning that the assets used to insure the households are not contingent on the different possible endowment states. Households, therefore, cannot fully insure themselves against income fluctuations or perfectly smooth consumption across all possible future endowment states. As a result, households usually save when income is high to protect themselves for when income is low.

The first heterogeneous agent models (Aiyagari (1994); Huggett (1993)) simplified the computational complexity of these models by focusing only on the stationary general equilibrium. First, assuming that prices are constant, they solve for the individual agent's problem and obtain the optimal consumption and savings decision policies for all agents, based on idiosyncratic income and the return on assets from

[2] Also called, by opposition, representative-agent (RA) models.

[3] This can be used to represent important inequalities of wealth and of outcomes in the labor market — such as getting fired, receiving a promotion or a change in the hours worked — among individuals in an economy.

the previous period. Second, they compute the distribution associated with these optimal rules, as well as the household aggregates that result, Consumption and Assets. They then solve the rest of the economy. Finally, they repeat this same step multiple times for different values of r to find the $r*$ that clears the markets.

Due to the complexity of the partial equilibrium, which requires root finding algorithms in order to be solved, this resolution approach is slow and costly to compute. This method becomes even more costly once we become interested in observing the transition dynamics of the economy after a shock, since we need to solve for the complex partial equilibrium of households for all values of the state space of (a, e), for all values of r until $r*$ is found, and for all times t after the shock until the economy reaches the new stationary equilibrium.

A different method proposed by Achdou et al. (2022) bypasses the difficulties of this approach by simplifying the problem of the household and the steps needed to solve it, making the dynamics of the transition easier and faster to compute. Their method is centered on a continuous-time approach and is based on the mathematical theory of "Mean Field Games", a system that revolves around two coupled equations. In the context of heterogeneous agent models, the Hamilton–Jacobi–Bellman (HJB) equation solves the optimal policy choices of an individual who takes prices and the evolution of the distribution as given. Then, the Kolmogorov Forward (KF) equation characterizes the distribution given the policy choices of each individual. The rest of the model is then solved as usual, with a system of equations that determine the prices in the economy, given the distribution and the aggregates, updating r until the markets are clear.

This method is faster when it comes to solving the partial equilibrium of the households, since the solution to the HJB equation is a very sparse matrix, of tri-diagonal shape, which reduces the cost of computing. Once found, the solution to the KF equation is a transpose of the HJB's solution matrix, further reducing computational costs without loss of accuracy. Another significant advantage of the continuous-time method lies in the fact that the variables in the equations are static in time. The two key variables, individual consumption and assets, are linked by the mathematical relationships from the current period onward. This avoids the slow root-finding operations of the discrete-time method, which links individual consumption to assets at two different times.

Auclert et al. (2021) propose another method to speed up the computation of transition dynamics in models with heterogeneous agents in discrete time. Instead of solving the partial equilibrium of the household for every t, they develop a method that bypasses this step entirely. They introduce the Sequence-Space Jacobian, a matrix mapping each aggregate output to a sequence of inputs, which allows us to compute transition dynamics after a shock without solving the household problem every period. It is of dimensions $n_y \times n_x$, where n_y and n_x are the number of elements in the Y and X vectors of inputs and outputs respectively [4]. Each block in the matrix is a sub-Jacobian of dimensions $T \times T$: they contain the reactions of output Y_i to input X_i for all possible response and shock times. Each Jacobian cell is the derivative of an output at time t to the shift of an input at time s. The Jacobian itself is built on the underlying heterogeneity of agents but is obtained by solving the household problem only once, at the steady-state. The transition and policy rules at all t are then found with a shortcut which allows one to do it through a single backward iteration. This method improves the speed and space needed for computation, since:

- The calculation of the sequence-space Jacobian itself is faster than a direct estimation due to the exploitation of several mathematical relationships.
- The Jacobian only needs to be computed once and can be used directly to explore the effects of shocks hitting the economy.

Without a detailed explanation of the mathematical proofs, the method to construct the Jacobian can be broken down into several steps[5]:

[4] It is important to note that in a heterogeneous agents model, this Jacobian does not hold information about the distribution of households across variables at the micro level, That is, for example, we do not care about the individual consumption choices, but the total consumption aggregated over all households in period t.

[5] See Auclert et al. (2021) for a more rigorous explanation of their method.

First, Auclert et al. (2021) define the partial equilibrium with three main equations:

$$\mathbf{v}_t = v(\mathbf{v}_{t+1},\, \mathbf{X}_t)\,, \tag{3}$$
$$\mathbf{D}_{t+1} = \Lambda(\mathbf{v}_{t+1},\, \mathbf{X}_t)'\, \mathbf{D}_t, \tag{4}$$
$$\mathbf{Y}_t = y(\mathbf{v}_{t+1},\, \mathbf{X}_t)'\, \mathbf{D}_t \tag{5}$$

Equation (3) explains how the value function of the household $\mathbf{v}_t$ relates to the vector of household inputs and the value function of the next period. This helps us to establish policy choices. Equation (4) explains how the distribution of households $\mathbf{D}_t$ is updated in the next period with respect to the value function, inputs, and distribution of the current period, with Λ the transition matrix between periods. Therefore, this equivalence pins down the inter-temporal evolution of the distribution. Finally, equation (5) relates the aggregate outputs of the partial equilibrium to the individual level outcomes (y, given by the function y) and the shape of the distribution along those outcomes.

The goal is to establish the Jacobian $\mathcal{J}_{t,s}$ of the system of equations which link the stacked vector of inputs to the stacked vector of aggregate outputs defined by these three equations, $\mathbf{Y} = h(\mathbf{X})$. All three functions are differentiable around the steady-state. We assume that at the beginning of the period, the initial distribution at the time of a shock at time $t = 0$, D_0, is given and equal to D_{ss}, and only changes at the end of the period, that is, becomes $D_1 \neq D_{ss}$. Additionally, after a certain number of periods T following a shock, we arrive at a new steady-state such that terminal values are at their steady-state level, $\mathbf{v}_T = \mathbf{v}_{ss}$, and inputs are at the steady-state level one period before: $X_{T-1} = X_{ss}$.

The direct way to compute the s$^{\text{th}}$ column of the Jacobian (the responses of output Y_i from a shock on input X_i in s), we can iterate equation (3) backward, starting with $\mathbf{v}_T = \mathbf{v}_{ss}$, and compute the value function, the transition matrix, and the individual outputs defined by y. Then we iterate equation (5) forward, starting from $D_0 = D_{ss}$ and recursively find the distributions for every period from $t = 1$ to $t = T - 1$. For every period, we use the distribution and the individual outputs obtained to compute the aggregate outputs. We can then compute the values of the s$^{\text{th}}$ column of the Jacobian since $\mathcal{J}_{t,s} \equiv \frac{\mathbf{Y}_t^s - \mathbf{Y}_{ss}}{dx}$. The two iterations that solve the household problem have to be executed T times, for each possible value of s.

Auclert et al. (2021) propose a different method to build the Jacobian, called the fake news algorithm, which exploits several shortcuts to reduce computational complexity. They exploit the fact that the model is stationary and therefore its parameters do not vary with time to reduce the number of iterations needed to compute the Jacobian. In reality, households do not care about when the shock occurs in "calendar time" (whether the shock takes place at $t = 2$ or $t = 3$) but rather about what the distance from the shock to their current time period t is (the value $s - t$). The reaction of agents in $t = 2$ to a shock in $t = 4$ would be the same reaction of agents in $t = 0$ to a shock in $t = 2$. This means that the backward iterations of the household problem (the transition matrix lambda and the individual policy choices) only need to be iterated backward once. Computing a single backward iteration to find the response in t of a shock that takes place in $s = T - 1$ allows us to map the reaction of t of all possible shock times s, simply using the response of our one iteration and moving it so that it is the right distance from t.

They then define a matrix $\mathcal{F}_{t,s}$ that defines the difference between the aggregate response of the output at t to a shock at date s, and its response at $t + 1$ to a shock at date $s + 1$:

$$\mathcal{F}_{t,s} \cdot dx \;\equiv\; d\mathbf{Y}_t^s \;-\; d\mathbf{Y}_{t-1}^{s-1}\,. \tag{6}$$

They find that the only difference between both responses is that the starting distribution at time $t + 1$ has shifted. They finally show that this matrix is a function only of the steady-state transition matrix Λ, the policy function y and the distribution at time 1 after a shock at time s, D_1^s:

$$\mathcal{F}_{t,s} \cdot dx \;=\; \mathbf{y}_{ss}' \left(\Lambda_{ss}'\right)^{t-1} d\mathbf{D}_1^s\,. \tag{7}$$

Therefore, the difference in initial distribution affects the aggregates at all dates. In doing so, they bypass the forward iteration of the distribution.

Since the Jacobian $\mathcal{J}$ of the system satisfies can be defined as $\mathcal{J}_{t,s} = \sum_{k=0}^{\min\{s,t\}} \mathcal{F}_{t-k,s-k}$, computing it is straightforward. First, we compute the transition matrix and policy function through backward iteration only once. They can be used at all times t and s, and for all aggregate outputs of the household problem. Combining them with the initial steady-state distribution, we obtain the T values for the changes in aggregate output at time 0 from the shock at different values of s, and the T vectors for the changes in the distribution at date 1 from the shock at different values of s. Finally, these results give us $\mathcal{F}_{t,s}$ from which we can directly build the Jacobian.

In summary, the Jacobian is computed with only one backward iteration and a recursive computation of the steady-state transition matrix. It allows us to observe the effects of a shock on aggregates without having to solve the household problem at every t. An important caveat of this method is that the sequence-space Jacobian relies on a linear approximation of the system, which forces the shocks to be relatively small. This limits the use of this method for large shocks with higher order effects, which can cause an important share of the households to suddenly be at the borrowing constraint. Auclert et al. (2021) propose to solve for non-linearities by using the standard Newton Method, which computes the Jacobian every t by iterating over different possible values, but using the sequence-space Jacobian as an approximate first guess for every iteration to speed up the process.

Another caveat is that this method does not allow us to directly observe the evolution of the distribution throughout the transition, which is useful when studying the effects of different macroeconomic policies on inequalities at the household level. A way solve this problem is to declare the values of individual-level outputs at different quantiles of the mass of households to obtain the aggregate level of output at that quantile. This allows us to compare, for instance, the evolution of aggregate consumption or assets of households in the bottom and top 10 percentiles of the income distribution after a shock.

3 Presentation of the model

The model of Pieroni (2023) follows the Achdou et al. (2022) continuous-time method, with incomplete markets. In the steady-state, there is only idiosyncratic risk and no aggregate risk. The aggregate distribution of households is stationary, and so are all aggregate variables. The model does not feature accumulation of capital or a government. The economy can be divided into several blocks as follows.

3.1 The household's intertemporal problem

Households face a choice every period between current and future consumption. To trade resources between periods and protect themselves against idiosyncratic productivity shocks z_t, households trade assets a_t. The state space of an individual is defined within $(a, z) \in X = A \times Z$, with A and Z being the asset and productivity spaces, respectively. Ψ_t is the probability distribution on X, the fraction of the population in every combination of assets a, and the productivity state z. $f_t(a, z)$ is the associated density function. The household program is the following:

$$\max_{c_t} \quad \mathbb{E}_0 \int_0^{\infty} e^{-\rho t} \left(\frac{c_t^{1-\gamma}}{1-\gamma} - \frac{n_t^{1+\nu}}{1+\nu} \right) dt \tag{8}$$

$$\text{s.t.} \quad da_t = (w_t n_t z_t + r_t a_t + d_t - c_t)\, dt \tag{9}$$

$$a_t \geq -\phi \tag{10}$$

Households choose the total consumption level c_t to maximize their value function every period, where $\gamma > 0$ is the inverse elasticity of intertemporal substitution. They earn a wage proportional to the effective labor provided $w_t n_t z_t$, where the effective labor is measured by the productivity of the household. The households also receive a return on the assets saved, $(1 + r_t)a_t$, and earn dividends that come from the total profits of the firms, D_t, which are distributed proportionally to the productivity of each household during period t: $d_t = \left(z_t / \int z_t \, d\psi_t\right) D_t$. Idiosyncratic risk follows a lognormal process, where σ_z is the standard deviation rate of the log-income process, ν_z the mean reversion parameter, and $d\hat{w}_{z,t} \sim \mathcal{N}(0, dt)$ is a standard Brownian motion: $d \ln z_t \;=\; -\nu_z \ln z_t dt \;+\; \sigma_z \, d\hat{w}_{z,t}$.

Solving this problem with the HJB and KF equations gives us the optimal policy choices a_t and c_t and the stationary distribution such that:

$$A_t = \int_X a_t \, d\psi_t(x), \qquad C_t = \int_X c_t \, d\psi_t(x) \tag{11}$$

The notation of the budget constraint is characteristic of continuous-time notation and highlights the law of motion of assets with respects to time. This is equivalent to $\frac{da_t}{dt} = w_t n_t z_t + r_t a_t + d_t - c_t$. The interpretation is that over time, the shape of a is proportional to the flow of income minus consumption.

The borrowing constraint is negative, which is a standard practice in continuous-time methods. As explained by Achdou et al. (2022), in practice, the borrowing constraint never binds: households never hit it, and it is customary to place a borrowing constraint that is inferior to the level of assets where most of the mass of households must be grouped. Because Pieroni wants most of the distribution of households to be grouped at $a = 0$, he ensures that, by convention, the borrowing constraint must at least be negative.

$\rho < 1$ represents the rate of discount for future periods. It is similar to the parameter $\beta < 1$ in discrete-time models. However, instead of being a multiplicative factor of future utility like β, ρ is the rate at which households discount time, and therefore is often much smaller than β. Where by convention β is usually at least superior to 0.95, ρ is rarely superior to 0.1.

Despite the disutility of labor of the household, the quantity of labor provided is determined by a labor recruiting firm and does not depend on the households' choice. Therefore, households maximize their value function only with respect to consumption. This disutility will come into play later in the union's program.

3.2 The household's intratemporal consumption choice

The consumption bundle c_t is composed of a consumption good, $c_{g,t}$ and energy consumption, $c_{e,t}$. In this model, households must to consume an incompressible level of energy expenditures, $\underline{c}$, representing the subsistence level of energy consumption. An example of this would be indispensable appliances needing energy, such as lighting, cooking, or washing. This parameter restricts households' elasticity of reaction to a change in the price of energy, and restricts the poorest of households, since it represents a bigger share of their income.

$$c_t = \left[\alpha^{\frac{1}{\sigma}} \left(c_{e,t} - \underline{c}\right)^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} c_{g,t}^{\frac{\sigma-1}{\sigma}}\right]^{\frac{\sigma}{\sigma-1}} \tag{12}$$

The elasticity of substitution between goods is given by σ, and α is the distribution parameter. The prices of each composite good are $p_{g,t}$ and $p_{e,t}$. The consumption bundle is the numeraire in this economy, such that that prices are relative to the price index $P_t = 1$ and $P_t c_t = p_{g,t} c_{g,t} + p_{e,t} c_{e,t}$. Therefore, the following CES relationship for prices holds:

$$P_t = \left[\alpha p_{e,t}^{1-\sigma} + (1-\alpha) p_{g,t}^{1-\sigma}\right]^{\frac{1}{1-\sigma}} \tag{13}$$

Differentiating the change in consumption of the aggregate bundle with respect to each of the composite goods, using both the CES consumption relationship and the budget constraint for the consumption bundle, we obtain the demand for each composite good:

$$c_{g,t} = c_t \, (1-\alpha) \left(\frac{p_{g,t}}{P_t}\right)^{-\sigma}, \qquad c_{e,t} = \underline{c} + c_t \, \alpha \left(\frac{p_{e,t}}{P_t}\right)^{-\sigma} \tag{14}$$

3.3 Labor demand from a competitive recruiting firm

Households supply a continuum of labor services that are imperfect substitutes, indexed as $j \in [0,1]$. A competitive recruitment firm chooses the demand for each type of labor input j given the nominal wage of that service, subject to an elasticity of substitution across different labor inputs θ_w:

$$\begin{cases} \max\limits_{N_{jt}} & W_t\, N_t - \int_0^1 W_{jt}\, N_{jt}\, dj \\ \text{s.t.} & N_t = \left(\int_0^1 N_{jt}^{\frac{\theta_w - 1}{\theta_w}}\, dj \right)^{\frac{\theta_w}{\theta_w - 1}} \end{cases}$$

This problem gives us the demand for each individual labor input:

$$N_{jt} = \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \tag{15}$$

3.4 Union's Wage-Setting Problem

The domestic labor disutility function $v(n_t) = \frac{n_t^{1+\nu}}{1+\nu}$ enters the economy through the union's problem. Unions set nominal wages for each labor input j to maximize the average welfare of the union members, given their marginal utility of consumption and disutility of labor. Wage adjustment is subject to a quadratic utility cost. In continuous time, the representative union's optimization problem is given by:

$$\begin{cases} \max\limits_{\dot{W}_{jt}} & \int_0^\infty \exp\left(- \int_0^t r_s ds \right) \left[\int_0^1 \left(\frac{W_{jt}}{P_t} N_{jt} - \frac{v(N_{jt})}{u'(C_t)} - \frac{\Psi_w}{2} \left(\frac{\dot{W}_{jt}}{W_{jt}} \right)^2 N_t dj \right) \right] dt \\ \text{s.t.} & N_{jt} = \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \\ & \pi_{w,t} = \frac{\dot{W}_t}{W_t} \end{cases} \tag{16}$$

This problem yields the following continuous-time New Keynesian wage Phillips curve:

$$\pi_{w,t} \left(r_t - \frac{\dot{N}_t}{N_t} \right) = \dot{\pi}_{w,t} + \frac{\theta_w}{\Psi_w} \left(\frac{v'(N_t)}{u'(C_t)} - w_t \mu_w^{-1} \right) \tag{17}$$

3.5 Firms

The program of the firm is the second place where energy enters the model. First, the representative final goods producer chooses intermediate inputs $\{Y_{it}\}_{i \in [0,1]}$ to maximize aggregate profits subject to a CES production function over differentiated varieties, where $\theta_p > 1$ denotes the elasticity of substitution between intermediate goods:

$$\begin{cases} \max\limits_{Y_{it}} & P_t\, Y_t - \int_0^1 P_{it}\, Y_{it}\, di \\ \text{s.t.} & Y_t = \left(\int_0^1 Y_{it}^{\frac{\theta_p - 1}{\theta_p}}\, di \right)^{\frac{\theta_p}{\theta_p - 1}} \end{cases}$$

This gives us the intermediate good demand for each intermediate variety:

$$Y_{it} = \left(\frac{P_{it}}{P_t} \right)^{-\theta_p} Y_t \tag{18}$$

Equation (18) describes the demand for intermediate good i, which is decreasing in its relative price P_{it}/P_t and proportional to aggregate output Y_t.

A share of the total profits is distributed as dividends:

$$D_t = Y_t \left(1 - \mu_p^{-1}\right) \tag{19}$$

where $\mu_p = \dfrac{\theta_p}{\theta_p - 1}$ is the price markup.

Each intermediate firm $i \in [0,1]$ chooses labor N_{it} and energy E_{it} to minimize total real costs given factor prices w_t and $p_{e,t}$, subject to a CES production technology for intermediate output Y_{it}:

$$\begin{cases} \min\limits_{N_{it},E_{it}} & w_t N_{it} + p_{e,t} E_{it} \\ \text{s.t.} & Y_{it} = \left[\alpha^{\frac{1}{\sigma}} E_{it}^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} N_{it}^{\frac{\sigma-1}{\sigma}}\right]^{\frac{\sigma}{\sigma-1}} \end{cases} \tag{20}$$

where σ denotes the elasticity of substitution between inputs, and α is the share parameter reflecting the relative weight of energy. Differentiating the Lagrangian with respect to N_{it} and E_{it} gives the conditional input demands of the intermediate firm:

$$N_{it} = (1-\alpha)\left(\frac{w_t}{\lambda_t}\right)^{-\sigma} Y_{it}, \qquad E_{it} = \alpha \left(\frac{p_{e,t}}{\lambda_t}\right)^{-\sigma} Y_{it}. \tag{21}$$

The Lagrange multiplier λ represents the marginal cost of producing one unit of the intermediate good. It is equivalent to the real marginal cost of production:

$$mc_t = \left[\alpha\, p_{e,t}^{1-\sigma} + (1-\alpha)\, w_t^{1-\sigma}\right]^{\frac{1}{1-\sigma}} = \lambda_t \tag{22}$$

In symmetric equilibrium, the demand for each factor by intermediate firms represents the total demand for factors in the economy: $E_{it} = E_{f,t}$ snd $N_{it} = N_t$.

3.6 Price-Setting Problem

Intermediate producers set prices subject to quadratic adjustment costs. In continuous time, the representative intermediate firm i solves the following:

$$\begin{cases} \max\limits_{\dot{P}_{it}} & \displaystyle\int_0^\infty \exp\!\left(-\int_0^t i_s\, ds\right) \left[(P_{it} - m_{it})Y_{it} - \frac{\Psi_p}{2}\, \pi_{it}^2\, P_t Y_t\right] dt \\ \text{s.t.} & Y_{it} = \left(\frac{P_{it}}{P_t}\right)^{-\theta_p} Y_t, \\ & \pi_{it} = \frac{\dot{P}_{it}}{P_{it}}, \\ & m_{it} = mc_{it}\, P_t, \end{cases} \tag{23}$$

where $\theta_p > 1$ is the elasticity of substitution across intermediate goods, Ψ_p governs the cost of price adjustment, and mc_{it} is the real marginal cost. This problem implies the following continuous-time New Keynesian Phillips Curve:

$$\pi_t \left(i_t - \frac{\dot{Y}_t}{Y_t}\right) = \dot{\pi}_t + \frac{\theta_p}{\Psi_p}\left(mc_t - \mu_p^{-1}\right) \tag{24}$$

where $\mu_p = \frac{\theta_p}{\theta_p - 1}$.

3.7 Monetary Policy

The central bank sets the nominal interest rate according to a Taylor rule that responds to inflation, where r denotes the steady-state real interest rate:

$$i_t = r + \phi_\pi \pi_t + \epsilon_t, \tag{25}$$

with $\phi_\pi > 1$ the policy reaction coefficient and ϵ_t a monetary policy shock.

The following relationships also hold:

$$r_t = i_t - \pi_t, \tag{26}$$

$$\frac{\dot{w}_t}{w_t} = \pi_{w,t} - \pi_t \tag{27}$$

Equation (26) defines the real interest rate through the Fisher equation, while equation (27) relates real wage growth to wage and price inflation.

3.8 General Equilibrium

The general equilibrium of the economy is characterized by the market-clearing conditions for assets, labor, and energy:

$$A_t = B \tag{28}$$

$$N_t = \int_X z_t n_t \, d\psi_t(x) \tag{29}$$

$$E_t = E_{f,t} + C_{e,t} \tag{30}$$

where $A_t = \int_X a_t \, d\psi_t(x)$ and $C_t = \int_X c_t \, d\psi_t(x)$ are the total demand for assets and consumption by households, B is the aggregate constant supply of assets, $N_t = \int n_{it} z_{it} \, di$ denotes both labor demand and the effective labor supplied by households, $C_{e,t} = \int_X c_{e,t} \, d\psi_t(x)$ is the aggregate household energy consumption, $E_{f,t}$ is the energy demand by firm and E_t is the total energy supply. Because there are no government, capital, or trade components in this model, the aggregate resource constraint is given by:

$$C_t = Y_t - p_{e,t} E_{f,t} + Q_t, \tag{31}$$

Where $Q_t = r_t B + D_t$ is an endowment component of income. The added value in the economy is either consumed by households or used to pay the energy demands of the firms, minus dividends, which are included in C_t, and the exogenous payments of financial assets $r_t B$.

4 Discretization of the model

In discrete time, a number of equations still hold: the equations characterizing the CES relationship for prices (13), the individual consumption goods demand (14), the firms' production, demand, profits and marginal cost ((15), (18), (19), (21), (22)), the Taylor rule (25), the Fisher equation (26)) and the market-clearing conditions ((28), (29), (30), (31)). However, the households' problem (equations (8), (9) and (10)), the New Keynesian wage and price Phillips curves ((17), (24)) and the equation characterizing the wage growth to wage and price inflation (27) are in continuous time and must be discretized. To do so, we can follow a number of principles to help us discretize this set of equations.

4.1 Change of variables across time

In continous time, the growth of a variable across time is represented with differential equations in the form: $\dot{x}_t = \frac{dx_t}{dt}$. In the discrete-time model, these are rewritten as as difference equations between times t and $t+1$: $\dot{x}_t = x_{t+1} - x_t$. In continuous time, the differential equation indicates the change in x as t changes. In contrast, the difference equation answers the question: what is the difference between x today and x tomorrow?

This is a more general case of $\dot{x}_t \approx \frac{x_{t+1} - x_t}{\Delta t}$, but we take our change between periods to be $\Delta t = 1$. By the same principles, instantaneous expectations $E_t[\dot{x}_{t+dt}]$ become one-period ahead expectations $\mathbb{E}_t[x_{t+1}]$. This is particularly relevant for the discretization of the price and wage Phillips curves.

4.2 Future discounting

As indicated in the presentation of the household's problem (section 3.1), the dicsounting of future values follows different rules and notations in each method. In discrete time, future values are discounted by multiplying the value by the per-period factor β_t, whereas in continuous time they are discounted exponentially at the instantaneous rate the $e^{-\rho}$, i.e. $e^{\left(-\int_0^t r_s ds\right)}$. Additionally, aggregation of a variable or equation through time is done over integration in continuous-time and over a sum in discrete-time.

Therefore, in all intertemporal equations, we replace the discount factor often expressed as $\int_0^\infty \exp\left(-\int_0^t r_s ds\right)$ (see (16), (23)) with the following formula: $\mathbb{E}_0 \sum_{t=0}^\infty \beta^t$.

4.3 The discrete-time household's problem

A couple of principles applies to the partial equilibrium of the household. First, the process of idiosyncratic productivity becomes a Markov chain process standard in the Sequence-Space Jacobian method: $d \ln z_t = \rho_z \ln z_{t-1} + \epsilon_{z,t}$, where $\epsilon_{z,t} \sim \mathcal{N}(0, \sigma_z)$. This process is characterized by a transition matrix, which defines how households transition productivity states between periods, and a stationary matrix, which defines the invariant fraction of households in every state [6]. Therefore, the rule for aggregating variables over the distribution of households $C_t = \int_X c_t \, d\psi_t(x)$ becomes $C_t = \int c_t \, d\Gamma_t(z, a_{\_})$ according to the discrete time notation of Auclert et al. (2021), where $\Gamma_t(z, a_{\_})$ is a distribution over discrete states updated via a Markov transition matrix between t and $t+1$. Second, the borrowing constraint in continuous time $-\phi$ binds in discrete time and becomes strictly positive: ϕ.

4.4 Pieroni's model in discretized time

Applying these principles gives the following set of discretized model equations. First, the household's value function, budget and borrowing constraints become:

$$\begin{cases} V(a_{t-1}, z_t) = \max\limits_{c_t} \frac{c_t^{1-\gamma}}{1-\gamma} - \frac{n_t^{1+\nu}}{1+\nu} + \beta \, \mathbb{E}_t[V(a_t, z_{t+1})] \\ \text{s.t.} \quad c_t + a_t = (1 + r_t)a_{t-1} + w_t n_t z_t + d_t \\ \text{s.t.} \quad a_t \geq \phi \end{cases} \tag{32}$$

With utility $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$ so that $u'_c(c) = c^{-\gamma}$, we get the Euler equation, which characterizes the optimal tradeoff between periods:

$$\beta \, \mathbb{E}_t \left[\left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} (1 + r_{t+1}) \right] = 1 \tag{33}$$

Discretizing the union's problem gives the following maximization program:

$$\begin{cases} \max\limits_{W_{jt}} \quad \mathbb{E}_0 \sum_{t=0}^\infty \beta^t \left[\left(\frac{W_{jt}}{P_t} N_{jt} - \frac{v(N_{jt})}{u'(C_t)} \right) - \frac{\Psi_w}{2} \left(\frac{W_{jt} - W_{jt-1}}{W_{jt-1}} \right)^2 N_t \right] \\ \text{s.t.} \quad N_{jt} = \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \end{cases}$$

Solving this problem gives in the discrete-time New Keynesian Wage Phillips Curve, linking wage inflation to expected future wage inflation, real marginal cost, and the wage markup:

$$\pi_{w,t} = \beta \mathbb{E}_t[\pi_{w,t+1}] \frac{N_{t+1}}{N_t} + \frac{\theta_w}{\Psi_w} \left(\frac{v'(N_t)}{u'(C_t)} - \mu_w^{-1} w_t \right). \tag{34}$$

[6] For more details on the Rouwenhorst method, see the NBER workshop on how to use the SSJ toolkit for heterogeneous agent models: https://github.com/shade-econ/nber-workshop-2023/tree/main.

In discrete time, the price setting problem of firm i becomes:

$$\begin{cases} \max\limits_{P_{it}} & \mathbb{E}_0 \sum\limits_{t=0}^{\infty} \beta^t \left[(P_{it} - m_{it}) Y_{it} - \dfrac{\Psi_p}{2} \left(\dfrac{P_{it} - P_{it-1}}{P_{it-1}} \right)^2 P_t Y_t \right] \\ \text{s.t.} & Y_{it} = \left(\frac{P_{it}}{P_t} \right)^{-\theta_p} Y_t \\ & m_{it} = mc_{it} \, P_t \end{cases}$$

Taking $P_{it} = P_t$, $mc_{it} = mc_t$, $\pi_t = P_t - P_{t-1}$ and $\mu_p = \frac{\theta_p}{\theta_p - 1}$, and linearizing around a zero-inflation steady-state (so that $\pi_t(1 + \pi_t) \approx \pi_t$), we obtain the discrete-time New Keynesian price Phillips curve:

$$\pi_t = \beta \, \mathbb{E}_t \left[\pi_{t+1} \right] \frac{Y_{t+1}}{Y_t} + \frac{\theta_p}{\Psi_p} \left(mc_t - \mu_p^{-1} \right). \tag{35}$$

The following relationship also holds:

$$\frac{\dot{w}_t}{w_t} = w_{t+1} - w_t = \pi_{w,t} - \pi_t \tag{36}$$

See appendices A to E for a detailed derivation of the model's equations.

5 Using the SSJ toolkit to program the model

5.1 Structure of the code in continuous-time

Before explaining how to program the discrete-time version of the model, it is important to outline the structure of Pieroni (2023)'s original continuous-time implementation, using the available MATLAB files provided in the'supplementary' files' folder. This will allow us to highlight more practical differences between the continuous-time implementation of Achdou et al. (2022) and the discrete-time implementation of the SSJ toolkit. Additionally, it is useful to use the original code to have a more complete understanding of the equations and variables. This section therefore summarizes the algorithms behind the steady and dynamic states of the model.

Figure 1 illustrates the general structure of the algorithm used by Pieroni (2023) to find the values of all variables at the steady-state.

First, the algorithm creates the asset and productivity grids, given the parameters that define the shape and bounds of the processes and the number of points in each grid. The discretization of the states is a necessary step in solving the model. The steady-state system function then takes as input the model's parameters, initial guesses for the values of the unknowns (the interest rate r, labor and energy demand by the firms N and E_f), and the value of energy supply E_s, the exogenous variable of the model. The firm's program is a set of equations that compute total production Y given the quantities N and E_f, the prices of the factors w and p_e, total dividends and the array of transfers, which are the dividends distributed proportionally to productivity.

The household program takes the firm's outputs as inputs, and is divided into two separate functions. First, the HJB function finds the stationary value function and computes individual policy functions and the transition matrix between states by iterating the value function until the change between iterations is minimal. Then, the KF function uses the transition matrix to obtain the corresponding stationary distribution of households along the asset and productivity states.

The algorithm then computes aggregate consumption and assets given the individual policy functions c and a and the stationary distribution. Given aggregate consumption and the prices of energy and of the final good, it computes aggregate energy consumption and aggregate final good consumption. It also computes aggregate labor supply using the New Keynesian wage Phillips curve at the steady-state.

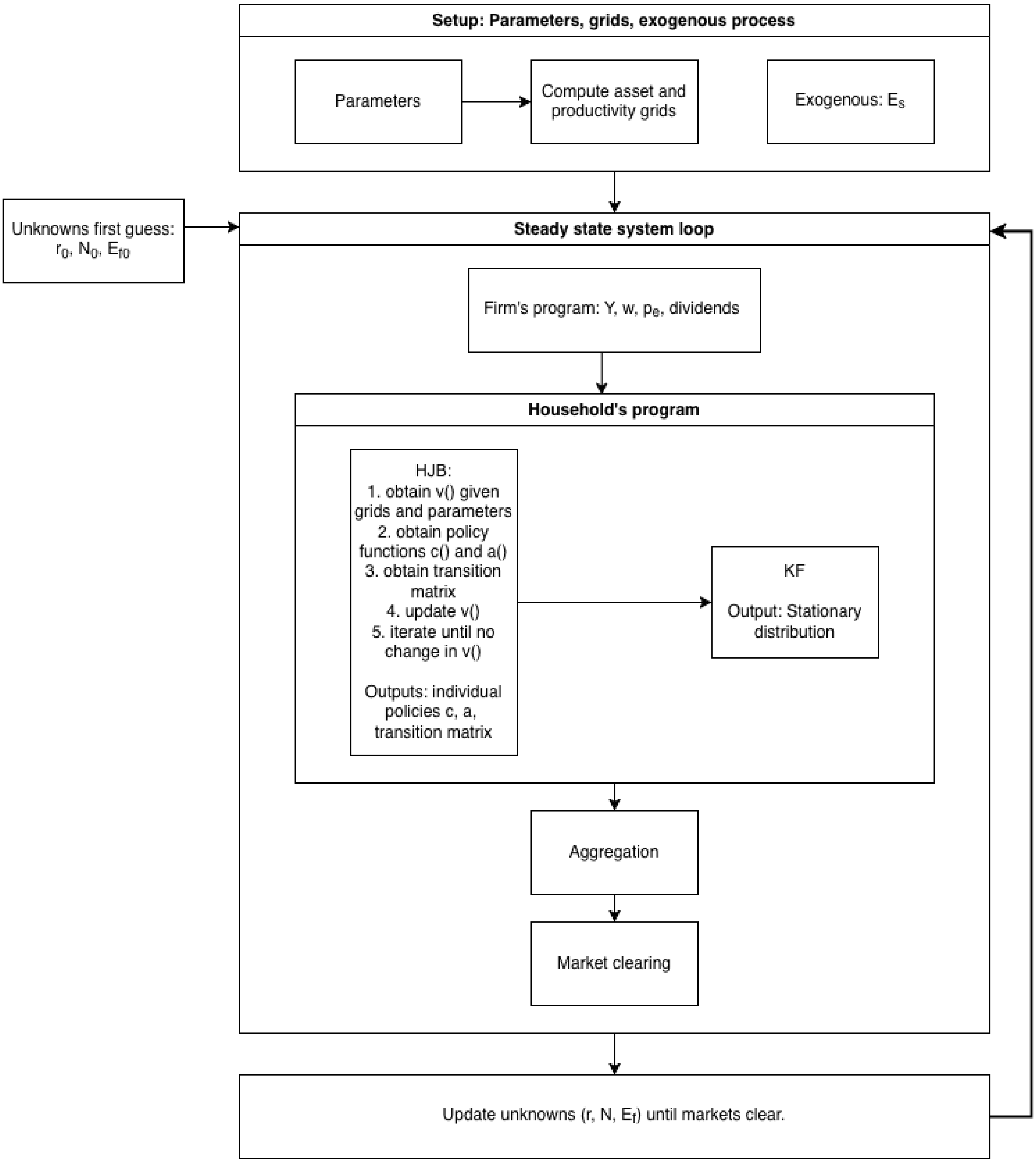


Figure 1: Structure of the steady-state loop in Pieroni (2023)'s code.

Finally, it computes the outputs of the steady-state system function, which are the residuals of the asset, labor, and energy market clearing equations. The algorithm updates the values of the unknowns and executes the steady-state function until the residuals are zero.

The structure of the algorithm defining the dynamic economy after a shock follows a similar structure. First, a sequence of values following the exogenous path of energy supply E_s enter into the model every period. Then, the model's equilibrium is found by iterating the values of the unknowns until the markets clear, except it now does so for every period after the shock (for every value of E_s). Additionally, the economy's system also computes the values of price and wage inflation.

It is important to note that although this is a comprehensive summary of the general steps of Pieroni (2023)'s algorithm, certain functions such as the creation of state grids or the HJB and KF functions follow advanced and complex computations such as trapezoid integration, which distract from a simpler explanation of the program's main functionality. However, this is not a requirement to understand the general structure of the continuous-time algorithm and how it differs from the structure followed in discrete-time.

5.2 Implementing the discrete-time model with the SSJ toolkit

In order to use the SSJ toolkit to order the model's equations into blocks and to program the steady and transition states using Python, we rely on the SSJ toolkit documentation and example notebooks that accompany it [7].

There are two important decisions to make when computing the model, namely how to structure the economy and what role will each variable play in the algorithm. The Sequence-Space Jacobian framework requires the economy to be divided into different blocks that can be ordered to form a directed acyclic graph (DAG), a graph in which nodes are linked by one way connections that do not form any cycles. Additionally, it is important to differentiate between different types of variables:

- Unknown variables are the variables that start the algorithm with a random, pre-determined value as a first guess. They will condition the rest of the variables of the model. If, for a certain unknown value, the markets do not clear at the end of the system, the values of the unknowns will be updated at the beginning of the next iteration. Each unknown is connected to a target equation that needs to clear at equilibrium.
- Exogenous variables are the variables that will suffer the shock and through which the rest of the economy will be impacted. Similarly to parameters, in the steady-state, we define their value once at the beginning of the code and never update it. However, by declaring a variable as exogenous, the toolkit will automatically compute the derivatives for the Jacobian matrix with respect to a change in these variables.
- Endogenous variables are the outputs of the model. They are completely dependent on the values of the parameters, unknowns, and exogenous variables. Each endogenous variable is associated to an individual equation.

When structuring our economy and designing our DAG, we must simultaneously consider which variables we choose as unknowns, exogenous, and endogenous variables. These rules are also valid even if the model's equations differ in a dynamic equilibrium.

We can start crafting our DAG by organizing our economy into blocks. Each block of the economy will be defined by a function. The SSJ toolkit offers different decorators that can be assigned to the different types of functions of the model, depending on their role in the economy. We will now go over the two main types of blocks.

[7] https://github.com/shade-econ/sequence-jacobian.

5.2.1 Defining the household block

The main block of the household's problem, also called a `HetBlock`, revolves around a main function we will call `household_decision`, which computes the resolution of the value function and takes as inputs:

- The grids of all possible asset and productivity states, two arrays of size $1 \times n_a$ and $1 \times n_z$ respectively;
- A grid of endowment values `wz` of size $1 \times n_z$ defining the salary earned by each productivity state;
- A grid of dividend values of size $1 \times n_z$ defining the dividend distributed to each productivity state;
- total labor demand `N`;
- The values of the parameters defining the discount rate β and elasticity of intertemporal substitution for consumption, γ;
- A grid of dimensions $n_a \times n_z$ containing the value of the derivative of the value function at the beginning of each iteration.

On the surface, the function computes the asset and consumption policy choices associated with the initial value function and defines the value function of the next iteration given these policies. However, by declaring this function with the decorator `@ssj.het()`, the toolkit will understand that this is the household's main function and:

1. Automatically iterate over it until the value function reaches its stationary value;
2. Use it to create a variable for the stationary distribution of households;
3. Automatically aggregate all of its outputs with the help of the distribution.

We must therefore create another function which sets up the initial value of derivative of the value function to enable the iteration to start. We will call it `hh_initial_guess`. This function will be defined as the `backward_init` parameter of the decorator. Additional inputs of the `@ssj.het()` decorator are:

- The transition matrix defining the Markov process for productivity `Pi_trans`;
- The name of the variable defining the individual level of assets;
- The name of the variable defining the first guess of the derivative of the value function. By convention, in the household's main function, the name of the derivative must be the same as in the initial guess but encoding `"_p"` at the end.

The code below illustrates the household's main functions for the discretized model.

```python
# Household Block

def hh_initial_guess(a_grid, wz, transfers, r, gamma_c):
    # define cash on hands as disposable income
    coh = (1 + r) * a_grid[np.newaxis, :] + wz[:, np.newaxis] + transfers[:, np.newaxis]
    # first guess: consumption is equal to 10% of cash on hands
    Vprime_a = (1 + r) * (0.1 * coh) ** (-gamma_c)
    return Vprime_a


@ssj.het(
    exogenous="Pi_trans",
    policy="a",
    backward="Vprime_a",
    backward_init=hh_initial_guess,
)
def household_decision(Vprime_a_p, a_grid, z_grid, wz, transfers, r, beta, gamma_c, N):
    # Endog Gridpoint Method: Get c on the next asset grid
```

```
    Uprime_c_nextgrid = beta * Vprime_a_p
    c_nextgrid = Uprime_c_nextgrid ** (-1 / gamma_c)

    # Interpolate to get the values of c on the current period asset grid
    coh = (
        (1 + r) * a_grid[np.newaxis, :]
        + wz[:, np.newaxis] * N
        + transfers[:, np.newaxis]
    )
    a = ssj.interpolate.interpolate_y(c_nextgrid + a_grid, coh, a_grid)

    ssj.misc.setmin(a, a_grid[0])  # Cannot borrow more than a_grid[0]
    c = coh - a

    # Total resources and effective labor of the agent
    inc = wz[:, np.newaxis] * N + (1 + r) * a + transfers[:, np.newaxis]
    nz = z_grid[:, np.newaxis] * np.ones_like(c) * N

    # Update the marginal value function
    Vprime_a = (1 + r) * c ** (-gamma_c)

    return Vprime_a, a, c, inc, nz
```

Two more types of function are needed to finish the household block: `hetinputs` and `hetoutputs`, special functions that handle the flow of information between heterogeneous-agent computations and the aggregate system. `Hetinputs` are functions that take aggregate variables and parameters to create individual-level quantities that will be used as inputs to the household function. They run before the household function. An example of a `hetinput` function is the function `make_grids`, which takes the parameters defining the asset and productivity grids to create the grids. It is important to highlight that the toolkit provides the function `ssj.grids.markov_rouwenhorst` which automatically computes the stationary and transition matrices associated with the idiosyncratic productivity process.

```
def make_grids(rho_z, sd_z, n_z, amin, amax, n_a):
    z_grid, pi_stationary, Pi_trans = ssj.grids.markov_rouwenhorst(
        rho=rho_z, sigma=sd_z, N=n_z
    )
    a_grid = ssj.grids.agrid(amin=amin, amax=amax, n=n_a)
return z_grid, pi_stationary, Pi_trans, a_grid
```

In contrast, `hetoutputs` are functions whose individual-level output will be automatically aggregated over the distribution. An example of a `hetoutput` function is `intratemporal_consumption`, which creates the grids of energy and final good consumption given the grid of individual-level consumption and the prices of each good. The toolkit will then automatically compute aggregated energy and final good consumption at the equilibrium. By convention, individual-level variable names must be lowercase, as the aggregated variable's name will be capitalized by the toolkit for to differentiate them from the individual-level arrays. These functions will be added to the household's main function with the use of the functions `add_hetinputs` and `add_hetoutputs`, to form the household block.

```
def intratemporal_consumption(ce_min, alpha_c, sigma_c, p_e, p_g, p_index, c):
    ce = ce_min + alpha_c * c * ((p_e / p_index) ** (-sigma_c))
    cg = (1 - alpha_c) * c * ((p_g / p_index) ** (-sigma_c))
    return ce, cg

household_block = household_decision.add_hetinputs(
        [set_wages, make_grids, set_transfers]
    )
household_block = household_block.add_hetoutputs(
        [get_mpc_transfers, intratemporal_consumption]
    )
```

5.2.2 Defining the rest of the economy

The rest of the economy in the discretized energy shortages model can be separated in five groups: the firm's block, the equations governing the Taylor rule, the New Keynesian wage and price Phillips curve, and the market clearings. These are relatively straightforward functions, which take aggregate inputs and compute aggregate outputs, for which no additional computations such as iteration or aggregation are required. It is good practice to declare them as simple blocks using the `@ssj.simple` decorator. This will allow the toolkit to differentiate between the values of the same variable at two different times, in the event that difference equations are required.

```
@ssj.simple
def firm(alpha_c, sigma_c, w, p_e, theta_p, Y, p_index):
    # compute the price of the other composite good
    numerator = p_index ** (1 - sigma_c) - alpha_c * p_e ** (1 - sigma_c)
    denominator = 1 - alpha_c
    p_g = (numerator / denominator) ** (1 / (1 - sigma_c))

    mc = (alpha_c * (p_e ** (1 - sigma_c)) + (1 - alpha_c) * (w ** (1 - sigma_c))) ** (
        1 / (1 - sigma_c)
    )

    N = (1 - alpha_c) * ((w / mc) ** (-sigma_c)) * Y
    E_f = alpha_c * ((p_e / mc) ** (-sigma_c)) * Y

    mu_p = theta_p / (theta_p - 1)
    profit = (1 - (mu_p ** (-1))) * Y
    return mc, E_f, N, profit, p_g
```

5.2.3 Declaring the model and computing the steady-state

Once each individual block of the economy has been written as a function, with the order of DAG and the types of variables in mind, we can group all blocks together to declare the model. The function `ssj.create_model` takes as input a list of the functions of the model and creates a custom `CombinedBlock` class which groups all equations together, automatically sorting them to create the DAG. Printing the properties of the model allows us to see the list of inputs and outputs.

The first step to solving the model is to program the steady-state. This is a crucial step since:

1. The absence of dynamics at the steady-state allows us to set some endogenous variables such as price and wage inflation to zero. This makes the model less constrained and allows us to play with different calibration values. This is especially useful when translating the calibration from continuous- to discrete-time, since the values of certain parameters will need to change to fit the properties of the new model.

2. It establishes the stationary values of all model variables, i.e., the starting values at the beginning of the shock.

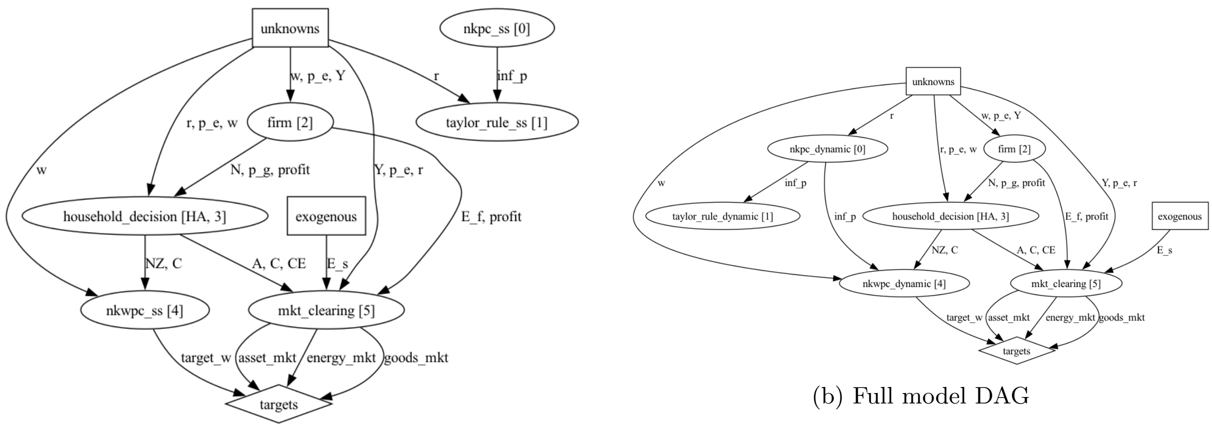
Usually, the equations inside each block are slightly different at the steady-state than in the dynamic equilibrium. Therefore, it is common to write two different functions for each block, one for the steady-state and one for the dynamic equilibrium.

We will first create the steady-state version of the model. The unknowns will be total production Y, the interest rate r, the price of energy p_e, and the wage w, and will respectively have as targets the resource constraint, the asset and energy markets, as well as the New Keynesian wage Phillips Curve. Instead of using the wage Phillips curve to compute labor supply and target the labor market clearing, we use the wage Phillips curve directly as the target. We choose four unknowns instead of three to have a less constrained model to work with and to allow production to fluctuate given that production cannot be an endogenous variable if we take only prices as unknowns. Like in Pieroni (2023), energy supply is the exogenous variable. The remaining variables are all endogenous.

```python
## creating the model

household_block = household_decision.add_hetinputs([set_wages, make_grids, set_transfers]
                                                   )
household_block = household_block.add_hetoutputs([get_mpc_transfers,
                                                   intratemporal_consumption])
blocks_ss = [household_block, firm, taylor_rule_ss, nkwpc_ss, nkpc_ss, mkt_clearing]
hank_ss = ssj.create_model(blocks_ss, name="Pieroni HANK SS")
```

The function `ssj.drawdag` draws the DAG of the model. We can use it to confirm the correct order of the model. Figure 2a shows the steady-state DAG. As we can see, the firm's block must go before the household block, as its outputs are inputs to the household function. The market clearing runs after every other block.



(a) Steady-state DAG

(b) Full model DAG

Figure 2: DAG of the heterogeneous agents model with energy shortages

To compute the steady-state equilibrium, we can use the list of inputs printed by the model as reference to create:

1. A calibration dictionary that connects the name of each parameter to its desired value ;
2. A dictionary that connects each of the unknowns to their initial guess.
3. A dictionary that connects each of the targets to the value they must have at equilibrium (generally, for market clearings, it is zero).

The function `solve_steady_state` takes these dictionaries and the model as inputs, computes the steady-state by iterating over the unknowns until the targets clear, and returns a dictionary with the equilibrium steady-state values of every parameter and variable of the model.

5.2.4 Computing the shock

Once we have successfully programmed the steady-state, computing the shock is relatively straightforward. Although the functions defining the Taylor rule and the price and wage Phillips curves differ with respect to their steady-state counterparts, in this case the chosen unknowns and targets do not change. Figure 2b shows the DAG of the dynamic equilibrium.

Using the same function `ssj.create_model`, we create the dynamic model:

```python
household_block = household_decision.add_hetinputs([set_wages, make_grids, set_transfers]
                                                    )
household_block = household_block.add_hetoutputs([get_mpc_transfers,
                                                    intratemporal_consumption])
blocks = [household_block, firm, taylor_rule_dynamic, nkpc_dynamic, nkwpc_dynamic,
                                                    mkt_clearing]
hank = ssj.create_model(blocks, name="Pieroni HANK Dynamic")
```

We then create an array of length T that stores the values of energy supply throughout and following the shock. Using Pieroni (2023)'s paper and code as a reference, we compute the same energy supply shock so that:

1. The energy supply decreases by 10% on impact.
2. The half-life of the shock is in the third quarter.
3. More than 70% of the shock is absorbed after 1 year.
4. The energy supply is fully back at the steady-state level after 3 years.

Given the shock, the functions `solve_jacobian` and `solve_impulse_linear` both compute the Jacobian of the model with respect to energy supply and output the path of all model aggregates given the sequence of energy supply shocks. Additionally, the function `solve_impulse_nonlinear` calculates the path of aggregates after a nonlinear shock following Auclert et al. (2021), using the Sequence-Space Jacobian as an initial guess to accelerate computation. The trajectory of the shock as well as the results will be presented in section 6.

5.3 Assessment of the SSJ approach

Using the SSJ toolkit to compute discrete-time heterogeneous agent models has several advantages. First, the toolkit is written in Python, a popular programming language, and is compatible with other frequently used scientific libraries such as Numpy, Scipy or Numba. The code also has a clean and usable structure. The economy is divided into intuitive blocks, and it is easy to modify existing functions or add new blocks to the model without having to make large re-arrangements to do so. The toolkit provides distinct and clear functions, such as a function which draws the DAG of the model to get a better idea of what the economy looks like. It is also easy to use and differentiate the different linear and non linear computations for the shock, since each has its own distinctive function.

Additionally, the time it takes for the model to solve the steady-state and dynamic equilibria proves that this method is indeed fast and efficient. The time required to compute the steady-state given the optimal equilibrium values is 0.1 seconds on a standard mac M1 silicone chip. In the worst case, if the calibration or first guesses are incorrect, it can take up to 2 to 3 minutes for the function to stop and display an error. The time required to compute the linear shock using the functions `solve_jacobian` and `solve_impulse_linear` is also 0.1 s. This performance is better than the time needed to compute the individual policy functions and aggregation using the endogenous grid point method according as reported in section 5.6 of Achdou et al. (2022).

Another important advantage of this method is that, in contrast to the algebraic complexity of the continuous-time algorithm, all complex computation (such as the steps needed to solve the value system, aggregate the variables, or create the Jacobian) is handled directly by the toolkit. Therefore, the code remains unburdened by lengthy mathematical formulas, and relatively less mathematics knowledge is needed to use the toolkit. On the other hand, it can sometimes be difficult to implement this algorithm, since certain naming conventions or errors can be hard to understand and are not all clearly presented in the notebook examples provided in the toolkit's GitHub. A basic understanding of the underlying code and documentation is necessary.

There are some additional difficulties in using the toolkit. It can be difficult to find the steady-state if the values of calibration parameters or initial guesses for the unknowns are poorly chosen or far from their equilibrium values. This often creates errors when trying to find the steady-state, such as "No convergence found after X iterations", or "Complex class not supported" (which occurs when one of the variables becomes a complex number during one of the iterations), which are difficult to interpret. Identifying the source of the problem and adjusting the relevant parameters can be challenging and extremely time-consuming. While this issue is common to all programming methods for macroeconomic models, the opacity between the surface, "user-level code" of the toolkit and its underlying computation makes debugging difficult.

Finally, as mentioned in section 2, a disadvantage of the SSJ framework in general is the loss of individual-level details along the transition. In order to track the evolution of distributional statistics, we need to define the relevant moments or quantiles as `hetoutputs` so that they are aggregated by the model. This requires a large number of output variables needed to observe the behavior of households with sufficient granularity.

6 Calibration, results and limitations

6.1 Adjusting the calibration

Implementing the model in discrete time requires adjusting the calibration with respect to its continuous time counterpart. Pieroni (2023) sets parameter values to either follow standard continuous-time conventions or to target moments from the German and Italian economies, which are among the most exposed to energy shortages. This section explains the modifications reported in table 1 made to align the calibration with discrete-time conventions and to preserve the same empirical targets. Table 2 summarizes the target aggregate statistics for the German economy, along with the values obtained by Pieroni (2023) and by the discrete-time calibration proposed here.

Figure 3 presents additional targets across the income distribution, comparing the results of Pieroni (2023) with those of the discrete-time model: the Lorenz curve of household income (with its corresponding Gini index in table 2), the share of energy expenditure over income across both household income percentiles and quintiles, and finally the distribution of marginal propensity to consume across income percentiles.

The parameters defining the asset and productivity processes are adjusted to match the standard values used in the discrete-time examples of Auclert et al. (2021). Since the borrowing constraint ϕ binds, it is set to zero. The number of grid points is increased relative to the continuous-time calibration and extended beyond the maximum asset level, following common practice in the SSJ toolkit notebooks. Because the productivity process in Pieroni (2023) differs from the discrete-time specification, the number of productivity states is reduced and its persistence and standard deviation adjusted to match the the calibration of the simple HANK notebook in the SSJ toolkit. However, the standard deviation is set to 0.4 (rather than 0.5) to better match the Gini coefficient and average wealth-to-income ratio in Table 2.

The parameters α and σ represent the distribution parameter and the elasticity of substitution—between consumption goods for households and between production factors for firms. Following Pieroni (2023), their values are taken from Bachmann et al. (2024), who estimate

- A 4% share of gas, oil, and coal in Germany's Gross National Expenditure.
- An elasticity of substitution between brown energy and other inputs such that a 10% negative shock to the energy input reduces production by 0.5% in a two-factor production framework.

We do not change these values, nor those of the price and wage adjustment parameters, the firm's input elasticities, and the Taylor rule coefficient.

We change the elasticity governing the intertemporal utility of the household γ and the Frisch parameter of labor disutility ν from 1 to 0.5 to obtain both a higher average MPC and a higher total energy as a share of output. We choose an unconstrained value for the discount factor β that does not create an error in the calculation of the steady state given the rest of the calibration.

Table 1: Model parameters in continuous and discrete time

Parameter	Description	Continuous-time value	Discrete-time value
Asset and Productivity grids			
n_a	Number of asset states	40	200
$-\phi$	Borrowing constraint	-1	0
$\bar{a}$	Maximum asset level	150	150
n_z	Number of productivity states	25	7
ρ_z	Mean reversion parameter / Persistence	0.0263	0.9
σ_z	Standard deviation	0.2	0.4
Households			
α	Distribution parameter	0.04	0.04
σ	Elasticity of substitution between goods	0.1	0.1
γ	CRRA/Inverse IES	1	0.5
ν	Frisch elasticity of labor supply	1	0.5
$\underline{c}$	Minimum energy consumption	0.0015	0.04
ρ (β)	Discount rate	0.08	0.995
B	Net asset supply	5.8	5
Firms and policy			
α	Distribution parameter	0.04	0.04
σ	Elasticity of substitution between factors	0.1	0.1
E_s	Energy supply	0.067	0.124
Ψ_p	Price adjustment cost	100	100
Ψ_w	Wage adjustment cost	100	100
θ_p	Intermediate goods elasticity	10	10
θ_w	Labor inputs elasticity	10	10
ϕ_π	Taylor coefficient	1.25	1.25

Finally, the values of the net asset supply B, the stationary energy supply E_s, and the minimum amount of energy consumption $\underline{c}$ were jointly chosen to best match the target statistics.

This step proved to be the most difficult, as these parameters are interdependent and inaccurate initial unknown guesses can easily prevent the steady state from converging. To facilitate and accelerate the calibration, we developed an algorithm that performs a brute-force search over different parameter combinations: it attempts to compute the steady state, skips to the next iteration when the computation fails, and records the set of parameters that minimizes the distance between the model-generated and aggregate target statistics. The algorithm also stores all successfully computed steady-state dictionaries in a cache to speed up future runs.

Table 2: Target statistics of the steady-state equilibrium

Statistic	Target	Continuous-time	Discrete-time
Average wealth-to-income share	4.2	4.4178	4.9315
Total energy as a share of output	0.04	0.0403	0.0472
Average energy expenditure share of consumption	[0.06:0.12]	0.0892	0.076
Gini coefficient	0.35	0.4812	0.4791
Average marginal propensity to consume	[0.15:0.25]	0.1069	0.1288

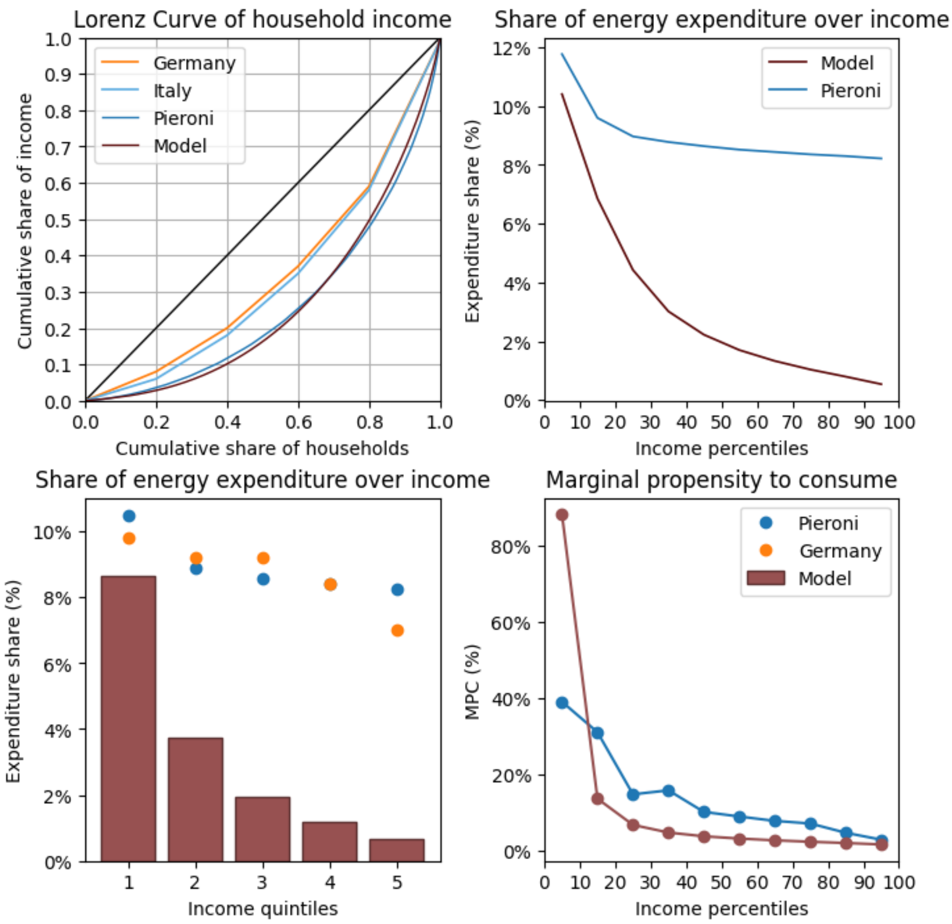


Figure 3: Steady-state income distribution, energy consumption and MPCs across the income distribution.

The discrete-time model fits the aggregate statistics reasonably well. The Gini coefficient is practically identical to that of the continuous-time model. While the total energy share of output is slightly higher than both the target and the continuous-time benchmarks, it remains close to the desired value. This is an improvement over previous calibrations, in which this statistic exceeded 30%.

The average marginal propensity to consume (MPC) is also closer to the target range than in the continuous-time model. However, the distribution of MPC does not match that of Pieroni (2023)'s model: the average MPC of the bottom income decile is more than twice that in the continuous-time case and roughly half the size of the average MPC observed for other income percentiles.

The average share of consumption of energy expenditure is lower in the discrete-time model but still within the acceptable target range. This is due to a relatively smaller share of energy expenditure among the top 90 percentiles (3).

Since the brute-force algorithm targets aggregate statistics, the fit could be improved with respect to the distribution of individual-level statistics, especially when it comes to the distribution of energy expenditure across the income distribution, which exhibits a high disparity between the bottom 10 and the rest of the households. This could be due to: (i) excessive wealth disparities between the top and bottom of the distribution; (ii) a minimum level of energy consumption that is too low relative to total energy supply; and (iii) a distribution parameter between energy and composite goods α that is too low and therefore reduces the relative weight of energy in the consumption bundle. Combined tweaks in the total supply of assets B, the standard deviation of the productivity process σ_z, the energy parameters E_s and $\underline{c}$, and the distribution parameter α might achieve a closer fit for the share of energy consumption across the income distribution.

6.2 Discrete-time impulse responses to a negative 10% energy supply shock

Once we have obtained a calibration that properly approximates the stationary target statistics, we can compute the impulse response functions of the model's aggregate parameters and prices. Both the `solve_jacobian` and the `solve_impulse_linear` functions produce the same results. Figure 4 shows the dynamics of the energy supply during the shock. These dynamics follow the calibration objectives described in section 5.2.4.

Figure 5 compares the behavior of the price of energy after shock between both models. Figures 6a and 6b show the behavior of other prices in the economy, such as the interest rate, the wage, and price inflation. Figures 7a and 7b show the responses of consumption and labor in the economy. Finally, figure 8 shows the aggregate responses of energy demand by both households and firms, as well as the demand for other composite good by households, in discrete-time. All response functions presented here are linear around the steady-state.

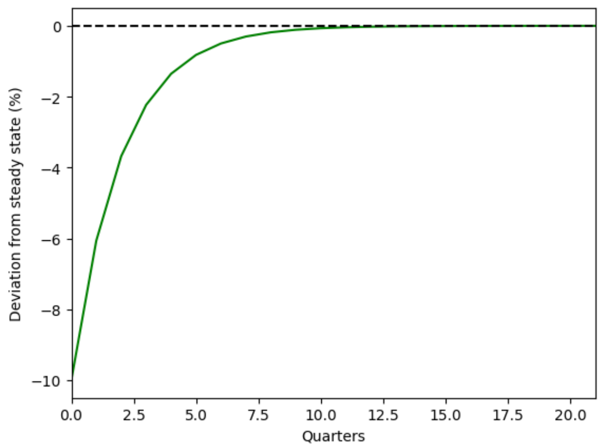


Figure 4: Energy supply shock

Upon impact, the price of energy increases to compensate for the decrease in energy supply, as initial energy demand by firms and households does not shift (figure 5). In response, the energy demand of firms decreases. Finally, labor demand falls in response to the shock, because labor and energy are complementary inputs in the production function. This leads to lower wages and less hours worked. In Pieroni (2023)'s continuous-time model, nominal wage rigidities shift the adjustment of wage towards hours. This limits the fall of real wages and prevents the fall of household consumption, which is dependent on earnings. In discrete-time, however, this does not happen. The price of energy increases by more than 1000% instead of 140% in continuous-time, and both wages and labor decrease by 46% and 27% respectively, compared to 1.3% and 2.7% in continuous-time (figures 6 and 7).

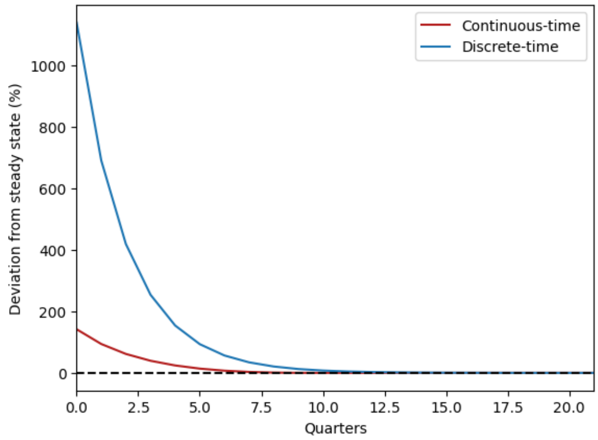
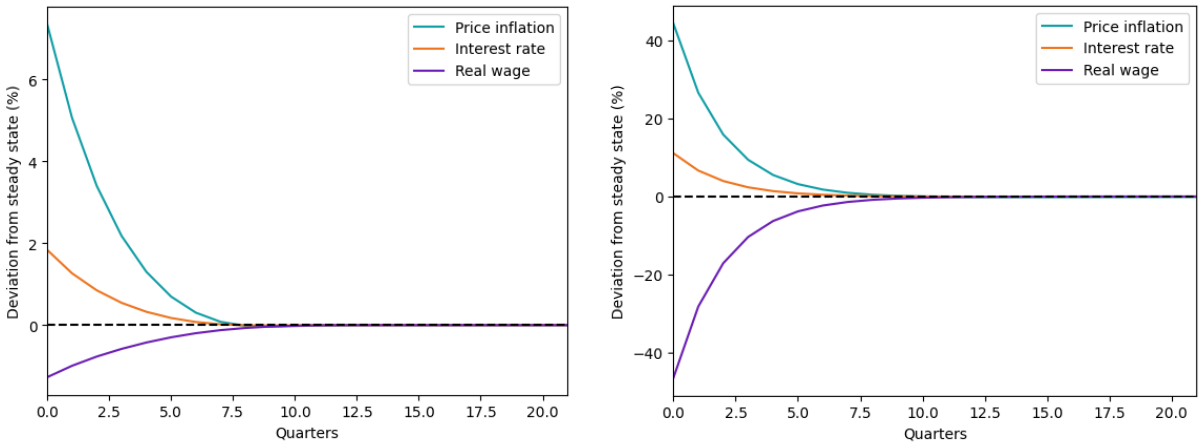


Figure 5: IRF of the price of energy in both continuous and discrete-time models



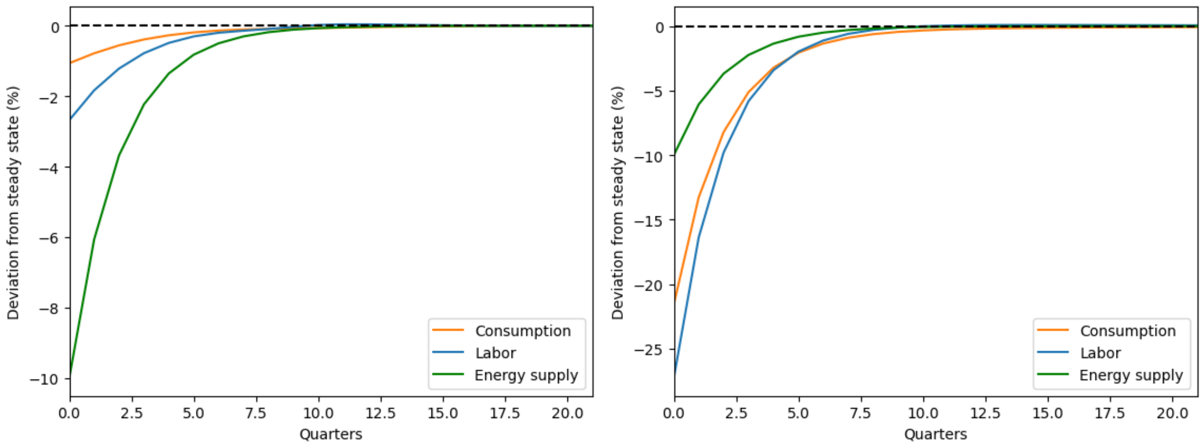
(a) IRF of prices, continuous time

(b) IRF of prices, discrete time

Figure 6: IRFs of prices to a 10% negative energy supply shock

The increase in the price of energy also causes marginal costs and CPI inflation to increase, despite the fall in wages (figure 6). The fall in consumption in both models is due to lower earnings, higher rates, and higher overall price inflation (see figure 7). The decrease in consumption reaches respectively 1% and 21% for the continuous- and discrete-time models upon impact.

The decomposition of household expenditure shows that aggregate energy expenditure falls by 4%, probably because of the increase in the price of energy, but not as much as aggregate consumption of the other composite good, which falls by almost 16%, despite the price of the composite good decreasing by 46% (see figure 8). This probably indicates the presence of a minimum constrained amount of energy expenditure. In comparison, given a reduction in overall income for the households, households discard other commodities to prioritize energy consumption.



(a) IRF of labor and consumption, continuous time

(b) IRF of labor and consumption, discrete time

Figure 7: IRFs of consumption and labor to a 10% negative energy supply shock

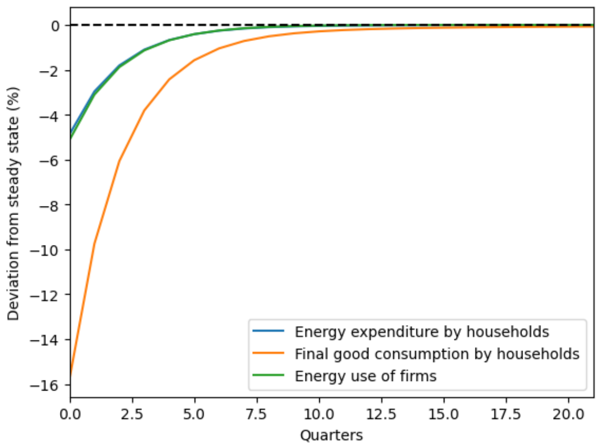


Figure 8: IRF of energy expenditure by households and firms and households consumption of the composite good

6.3 Analysis of the results

Several problems are apparent in the results. The linear responses of the aggregate variables and prices in the discrete-time model are of correct sign, but their magnitudes are noticeably larger than in the continuous-time benchmark. The model also presents high sensitivity to small changes made in the calibration. Slight alterations to certain parameter's values can result in an increase of the price of energy of more than 2000%, and consumption and labor responses that reach a -60% deviation at their peak. In addition, the function `solve_impulse_nonlinear` diverges after a few iterations.: the market-clearing residuals, initially small, increase rapidly in iterations until a numerical overflow produces `NaN` values.

These findings highlight the remaining challenges in obtaining a faithful discrete-time transcription of the continuous-time model. In this case, the discretized model appears to amplify the propagation of shocks and create numerical fragility.

A possible reason could be the current DAG and the selection of equilibrium unknowns and targets. If the unknowns and target relationships are too nonlinear and interdependent with each other, it could lead to an unstable system. A potential strategy could be to re-specify the discrete-time model to replace the unknowns with the quantities of factors N and E_f, use the New Keynesian wage Phillips curve to compute hypothetical labor supply, and then use the labor market clearing as a target, in the spirit of Pieroni (2023)'s algorithm. Additionally, the chosen calibration might reproduce stationary statistics effectively but enhance shock amplification. Therefore, further revisions on the calibration and the translation of the model's equations are also needed.

7 Conclusion

Macroeconomic models with heterogeneous agents enable a more complex analysis of inequalities between households, which is crucial to design better policies. We introduce a methodological step-by-step guide to translate the continuous-time HANK model with energy shortages introduced in Pieroni (2023) to discrete-time using the Sequence-Space Jacobian framework of Auclert et al. (2021). This framework comes with an available toolkit that significantly reduces complexity, and therefore the entry cost to computing heterogeneous agent models.

We offer a set of methodological principles to translate the model's equations to discrete-time and introduce how the SSJ toolkit can be used to implement and solve heterogeneous agent models. We find that while continuous-time models are mathematically rigorous and very useful to compute nonlinear shocks without modifying the model's structure, they require a more advanced knowledge of mathematics and complex, detailed code, which makes them difficult to implement and replicate. In contrast, the discrete-time Sequence-Space Jacobian framework provides a more accessible and transparent approach with the help of the SSJ toolkit. Its structure is cleaner and easier to extend, and therefore useful for replications and model extensions. However, the abstraction between the user-written code and the underlying toolkit algorithm can still make certain computational errors difficult to understand.

Adapting the calibration from continuous to discrete-time is an additional complex step to obtain a properly working model. In this case, an acceptable calibration was obtained with the help of a brute-force search algorithm, as we lacked a reference on how to translate several of the model's parameters. Targeting stationary statistics can help stabilize the model and the calibration correctly before computing the response of the economy to a shock. The discretized model matches well the aggregate stationary statistics of the economy but exhibits pronounced differences between the bottom percentile of the income distribution and the rest of the distribution of households, which do not match the realistic distribution.

Finally, the results show that the dynamics of the discretized model given a large 10% negative shock to the energy supply are several orders of magnitude superior to the continuous-time model. Furthermore, these magnitudes are extremely sensitive to small changes in the model's calibration. A nonlinear resolution approach fails to converge to a solution. Future work to understand what is causing these problems could attempt to change the structure behind the numerical computation of the model, such as changing the unknowns and the targets, and therefore the linear dependencies of the model, to reduce numerical complexities and minimize the chance of extreme results or `NAN`s. Other possible paths include exploring different calibrations to try and limit extreme numerical results, or comparing the behavior of both models when exposed to a smaller shock to assess whether such a shock is more suitable for the discrete-time framework.

Therefore, we conclude that developing a systematic methodology to translate continuous-time models into discrete time would be highly valuable to expand the use of heterogeneous agent models in the literature.

A Derivation of the households' intertemporal problem in discrete-time

Let the household's intertemporal problem be defined in discrete-time as:

$$\begin{cases} V(a_{t-1}, z_t) = \max_{c_t} \frac{c_t^{1-\gamma}}{1-\gamma} - \frac{n_t^{1+\nu}}{1+\nu} + \beta\,\mathbb{E}_t[V(a_t, z_{t+1})] \\ \text{s.t.} \quad c_t + a_t = (1+r_t)a_{t-1} + w_t n_t z_t + d_t \\ \text{s.t.} \quad a_t \geq -\phi \end{cases}$$

The Lagrangian is as follows:

$$\mathcal{L} = \frac{c_t^{1-\gamma}}{1-\gamma} - \frac{n_t^{1+\nu}}{1+\nu} + \beta\,\mathbb{E}_t[V(a_t, z_{t+1})] - \lambda_t\left[c_t + a_t - (1+r_t)a_{t-1} - w_t n_t z_t - d_t\right] - \mu_t^a[a_t + \phi]$$

The First-Order Conditions (FOC) with respect to c_t and a_t are the following:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial c_t} &= c_t^{-\gamma} - \lambda_t = 0, \\ \Rightarrow \quad u'_c(c_t) &= c_t^{-\gamma} = \lambda_t, \end{aligned} \tag{37}$$

$$\frac{\partial \mathcal{L}}{\partial a_t} = \beta\,\mathbb{E}_t\left[\frac{\partial V(a_t, z_{t+1})}{\partial a_t}\right] - \lambda_t - \mu_t^a = 0. \tag{38}$$

Because the asset constraint is non-binding, we have $\mu_t^a = 0$. The envelope condition implies:

$$\begin{aligned} \frac{\partial V(a_t, z_{t+1})}{\partial a_t} &= \frac{\partial \mathcal{L}_{t+1}}{\partial a_t} = -\lambda_{t+1} \cdot \big(-(1 + r_{t+1})\big) \\ &= \lambda_{t+1}(1 + r_{t+1}) \end{aligned} \tag{39}$$

Plugging this into the FOC for a_t, we get:

$$\begin{aligned} &\beta\,\mathbb{E}_t[\lambda_{t+1}(1+r_{t+1})] - \lambda_t = 0 \\ \Rightarrow \quad &\beta\,\mathbb{E}_t\left[\frac{\lambda_{t+1}}{\lambda_t}(1+r_{t+1})\right] = 1 \end{aligned} \tag{40}$$

Using $\lambda_t = u'_c(c_t)$, this becomes:

$$\beta\,\mathbb{E}_t\left[\frac{u'_c(c_{t+1})}{u'_c(c_t)}(1+r_{t+1})\right] = 1 \tag{41}$$

With utility $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$ so that $u'_c(c) = c^{-\gamma}$, the Euler equation can be written as:

$$\beta\,\mathbb{E}_t\left[\left(\frac{c_{t+1}}{c_t}\right)^{-\gamma}(1+r_{t+1})\right] = 1 \tag{42}$$

B Derivation of the households' intratemporal choice

In both continuous and discrete-time, the consumption bundle c_t is defined as:

$$c_t = \left[\alpha^{\frac{1}{\sigma}}\left(c_{e,t} - \underline{c}\right)^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} c_{g,t}^{\frac{\sigma-1}{\sigma}}\right]^{\frac{\sigma}{\sigma-1}}$$

The prices of each composite good are $p_{g,t}$ and $p_{e,t}$. The consumption bundle is the numeraire in this economy, such that that prices are relative to the price index $P_t = 1$ and $P_t c_t = p_{g,t} c_{g,t} + p_{e,t} c_{e,t}$. Therefore, the following CES relationship for prices holds:

$$P_t = \left[\alpha p_{e,t}^{1-\sigma} + (1-\alpha) p_{g,t}^{1-\sigma}\right]^{\frac{1}{1-\sigma}}$$

Deriving the change in consumption of the aggregate bundle with respects to each of the composite goods, using both the CES consumption relationship and the budget constraint for the consumption bundle, we obtain the demand for each composite good:

$$\begin{aligned}
\frac{\partial c_t}{\partial c_{g,t}} &= \frac{\sigma}{\sigma-1}\left[\alpha^{\frac{1}{\sigma}}\left(c_{e,t}-\underline{c}\right)^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} c_{g,t}^{\frac{\sigma-1}{\sigma}}\right]^{\frac{1}{\sigma-1}} (1-\alpha)^{\frac{1}{\sigma}} \frac{\sigma-1}{\sigma} c_{g,t}^{-\frac{1}{\sigma}} \\
&= c_t^{\frac{1}{\sigma}} (1-\alpha)^{\frac{1}{\sigma}} c_{g,t}^{-\frac{1}{\sigma}} \\
&= \left(\frac{(1-\alpha)\, c_t}{c_{g,t}}\right)^{\frac{1}{\sigma}}, \qquad (43)
\end{aligned}$$

$$\frac{\partial c_t}{\partial c_{e,t}} = \left(\frac{\alpha\, c_t}{c_{e,t}-\underline{c}}\right)^{\frac{1}{\sigma}} \qquad (44)$$

And

$$\frac{\partial c_t}{\partial c_{g,t}} = \frac{p_{g,t}}{P_t}, \qquad \frac{\partial c_t}{\partial c_{e,t}} = \frac{p_{e,t}}{P_t} \qquad (45)$$

Therefore:

$$\frac{p_{g,t}}{P_t} = \left(\frac{(1-\alpha)\, c_t}{c_{g,t}}\right)^{\frac{1}{\sigma}} \implies c_{g,t} = c_t\,(1-\alpha)\left(\frac{p_{g,t}}{P_t}\right)^{-\sigma}, \qquad (46)$$

$$c_{e,t} = \underline{c} + c_t\,\alpha\left(\frac{p_{e,t}}{P_t}\right)^{-\sigma} \qquad (47)$$

C Derivation of the union's problem in discrete time

In both continuous and discrete time, a competitive recruitment firm chooses the demand for each type of labor input j given the nominal wage of that service, subject to an elasticity of substitution across different labor inputs θ_w:

$$\begin{cases}
\max\limits_{N_{jt}} & W_t\, N_t - \int_0^1 W_{jt}\, N_{jt}\, dj \\
\text{s.t.} & N_t = \left(\int_0^1 N_{jt}^{\frac{\theta_w-1}{\theta_w}}\, dj\right)^{\frac{\theta_w}{\theta_w-1}}
\end{cases}$$

Deriving the Lagrangian allows us to obtain the demand for each individual labor input. To simplify the model, we assume a symmetric equilibrium in which $W_{jt} = W_t$ and $N_{jt} = N_t$.

$$\mathcal{L} = W_t \left(\int_0^1 N_{jt}^{\frac{\theta_w - 1}{\theta_w}} \, dj \right)^{\frac{\theta_w}{\theta_w - 1}} - \int_0^1 W_{jt} \, N_{jt} \, dj. \tag{48}$$

The first-order condition with respect to N_{jt} is:

$$\frac{\partial \mathcal{L}}{\partial N_{jt}} = W_t \left(\int_0^1 N_{jt}^{\frac{\theta_w - 1}{\theta_w}} \, dj \right)^{\frac{1}{\theta_w - 1}} N_{jt}^{-\frac{1}{\theta_w}} - W_{jt} = 0. \tag{49}$$

Rearranging, we obtain:

$$W_t \, N_t^{\frac{1}{\theta_w}} \, N_{jt}^{-\frac{1}{\theta_w}} = W_{jt},$$

$$\Rightarrow \quad N_{jt} = \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \tag{50}$$

In discrete time, the representative union's optimization problem is given by:

$$\begin{cases} \max\limits_{W_{jt}} & \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left[\left(\frac{W_{jt}}{P_t} N_{jt} - \frac{v(N_{jt})}{u'(C_t)} \right) - \frac{\Psi_w}{2} \left(\frac{W_{jt} - W_{jt-1}}{W_{jt-1}} \right)^2 N_t \right] \\ \text{s.t.} & N_{jt} = \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \end{cases} \tag{51}$$

The Lagrangian for this problem is:

$$\begin{aligned} \mathcal{L} = {} & \frac{W_{jt}^{1-\theta_w}}{W_t^{-\theta_w}} \frac{N_t}{P_t} - \frac{v\left(\left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \right)}{u'(C_t)} - \frac{\Psi_w}{2} \left(\frac{W_{jt}}{W_{jt-1}} - 1 \right)^2 N_t \\ & + \beta \left[\frac{W_{jt+1}^{1-\theta_w}}{W_{t+1}^{-\theta_w}} \frac{N_{t+1}}{P_{t+1}} - \frac{v\left(\left(\frac{W_{jt+1}}{W_{t+1}} \right)^{-\theta_w} N_{t+1} \right)}{u'(C_{t+1})} - \frac{\Psi_w}{2} \left(\frac{W_{jt+1}}{W_{jt}} - 1 \right)^2 N_{t+1} \right] \end{aligned} \tag{52}$$

The first-order condition with respect to W_{jt} is:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{jt}} = {} & (1-\theta_w) \left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} \frac{N_t}{P_t} - \frac{v'\left(\left(\frac{W_{jt}}{W_t} \right)^{-\theta_w} N_t \right)}{u'(C_t)} \left(-\theta_w \frac{N_t W_{jt}^{-\theta_w - 1}}{W_t^{-\theta_w}} \right) \\ & - \Psi_w \left(\frac{W_{jt}}{W_{jt-1}} - 1 \right) \frac{N_t}{W_{jt-1}} + \beta \Psi_w \left(\frac{W_{jt+1}}{W_{jt}} - 1 \right) \frac{N_{t+1} W_{jt+1}}{W_{jt}^2} = 0 \end{aligned} \tag{53}$$

Under symmetric equilibrium $W_{jt} = W_t$:

$$(1-\theta_w) \frac{N_t}{P_t} + \theta_w \frac{v'(N_t)}{u'(C_t)} \frac{N_t}{W_t} - \Psi_w \pi_{w,t} \frac{N_t}{W_{t-1}} + \beta \Psi_w \pi_{w,t+1} N_{t+1} \frac{W_{t+1}}{W_t^2} = 0 \tag{54}$$

Multiplying by W_t and defining the real wage $w_t = W_t / p_t$, we obtain:

$$(1-\theta_w) w_t N_t + \theta_w \frac{v'(N_t)}{u'(C_t)} N_t - \Psi_w \pi_{w,t} (1 + \pi_{w,t}) N_t + \beta \Psi_w \pi_{w,t+1} (1 + \pi_{w,t+1}) N_{t+1} = 0 \tag{55}$$

Finally, using $\mu_w = \frac{\theta_w}{\theta_w - 1}$ and assuming that $\pi_{w,t}(1 + \pi_{w,t}) \approx \pi_{w,t}$ around the steady-state, we obtain the New keynesian wage Phillips curve:

$$(1-\theta_w)w_t N_t + \theta_w \frac{v'(N_t)}{u'(C_t)} N_t - \Psi_w \pi_{w,t} N_t + \beta \Psi_w \pi_{w,t+1} N_{t+1} = 0, \tag{56}$$

$$\Rightarrow \quad \Psi_w \pi_{w,t} N_t = N_t \theta_w \frac{v'(N_t)}{u'(C_t)} + (1-\theta_w) w_t N_t + \beta \Psi_w \pi_{w,t+1} N_{t+1},$$

$$\Rightarrow \quad \pi_{w,t} N_t = N_t \frac{\theta_w}{\Psi_w} \left(\frac{v'(N_t)}{u'(C_t)} - \mu_w^{-1} w_t \right) + \beta \pi_{w,t+1} N_{t+1},$$

$$\Rightarrow \quad \pi_{w,t} = \beta \mathbb{E}_t[\pi_{w,t+1}] \frac{N_{t+1}}{N_t} + \frac{\theta_w}{\Psi_w} \left(\frac{v'(N_t)}{u'(C_t)} - \mu_w^{-1} w_t \right) \tag{57}$$

D Derivation of the firm's problem

In both continuous and discrete time, the representative final goods producer's intermediate input problem is:

$$\begin{cases} \max\limits_{Y_{it}} & P_t Y_t - \int_0^1 P_{it} Y_{it}\, di \\ \text{s.t.} & Y_t = \left(\int_0^1 Y_{it}^{\frac{\theta_p - 1}{\theta_p}}\, di \right)^{\frac{\theta_p}{\theta_p - 1}} \end{cases}$$

Differentiating the Lagrangian with respect to Y_{it}, we get the intermediate good demand for each intermediate variety:

$$\mathcal{L} = P_t \left(\int_0^1 Y_{it}^{\frac{\theta_p - 1}{\theta_p}}\, di \right)^{\frac{\theta_p}{\theta_p - 1}} - \int_0^1 P_{it} Y_{it}\, di \tag{58}$$

The FOC is:

$$\frac{\partial \mathcal{L}}{\partial Y_{it}} = P_t \left(\int_0^1 Y_{it}^{\frac{\theta_p - 1}{\theta_p}}\, di \right)^{\frac{1}{\theta_p - 1}} Y_{it}^{-\frac{1}{\theta_p}} - P_{it} = 0. \tag{59}$$

Rearranging:

$$P_t Y_t^{\frac{1}{\theta_p}} Y_{it}^{-\frac{1}{\theta_p}} = P_{it}, \tag{60}$$

$$\Rightarrow \quad Y_{it} = \left(\frac{P_{it}}{P_t} \right)^{-\theta_p} Y_t \tag{61}$$

The factor demand problem is:

$$\begin{cases} \min\limits_{N_{it}, E_{it}} & w_t N_{it} + p_{e,t} E_{it} \\ \text{s.t.} & Y_{it} = \left[\alpha^{\frac{1}{\sigma}} E_{it}^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} N_{it}^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}} \end{cases}$$

Differentiating the Lagrangian with respect to N_{it} and E_{it} gives:

$$\mathcal{L} = w_t N_{it} + p_{e,t} E_{it} - \lambda_t \left[\left(\alpha^{\frac{1}{\sigma}} E_{it}^{\frac{\sigma-1}{\sigma}} + (1-\alpha)^{\frac{1}{\sigma}} N_{it}^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}} - Y_{it} \right]. \tag{62}$$

$$\frac{\partial \mathcal{L}}{\partial N_{it}} = w_t - \lambda_t \frac{\partial Y_{it}}{\partial N_{it}} = 0, \tag{63}$$

$$\frac{\partial \mathcal{L}}{\partial E_{it}} = p_{e,t} - \lambda_t \frac{\partial Y_{it}}{\partial E_{it}} = 0. \tag{64}$$

The conditional input demands of the intermediate firm are:

$$N_{it} = (1-\alpha)\left(\frac{w_t}{\lambda_t}\right)^{-\sigma} Y_{it} \tag{65}$$

$$E_{it} = \alpha\left(\frac{p_{e,t}}{\lambda_t}\right)^{-\sigma} Y_{it} \tag{66}$$

The Lagrange multiplier λ represents the marginal cost of producing one unit of the intermediate good. It is equivalent to the the marginal cost of production, which we can find by differentiating the total cost function with respect to output Y_t:

$$\begin{aligned}
& mc_t = \frac{\partial \text{cost}_t}{\partial Y_t} \\
\Rightarrow \quad & mc_t = \alpha\, p_{e,t}\left(\frac{p_{e,t}}{mc_t}\right)^{-\sigma} + (1-\alpha)\, w_t \left(\frac{w_t}{mc_t}\right)^{-\sigma} \\
\Rightarrow \quad & mc_t = \alpha\, p_{e,t}^{1-\sigma}\, mc_t^{\sigma} + (1-\alpha)\, w_t^{1-\sigma}\, mc_t^{\sigma} \\
\Rightarrow \quad & mc_t^{1-\sigma} = \alpha\, p_{e,t}^{1-\sigma} + (1-\alpha)\, w_t^{1-\sigma}.
\end{aligned}$$

Hence, the real marginal cost of production is:

$$mc_t = \left[\alpha\, p_{e,t}^{1-\sigma} + (1-\alpha)\, w_t^{1-\sigma}\right]^{\frac{1}{1-\sigma}} = \lambda_t \tag{67}$$

E Derivation of the price-setting problem in discrete time

Finally, in discrete time, intermediate producers solve:

$$\begin{cases}
\max\limits_{P_{it}} & \mathbb{E}_0 \sum\limits_{t=0}^{\infty} \beta^t \left[(P_{it} - m_{it})Y_{it} - \frac{\Psi_p}{2}\left(\frac{P_{it} - P_{it-1}}{P_{it-1}}\right)^2 P_t Y_t\right] \\
\text{s.t.} & Y_{it} = \left(\frac{P_{it}}{P_t}\right)^{-\theta_p} Y_t \\
& m_{it} = mc_{it}\, P_t
\end{cases}$$

The associated Lagrangian can be written as:

$$\mathcal{L} = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left\{ (P_{it} - mc_{it} P_t)\left(\frac{P_{it}}{P_t}\right)^{-\theta_p} Y_t - \frac{\Psi_p}{2}\left(\frac{P_{it} - P_{it-1}}{P_{it-1}}\right)^2 P_t Y_t \right\} \tag{68}$$

Taking first-order conditions with respect to P_{it} gives:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial P_{it}} = & \frac{\partial}{\partial P_{it}}\left[P_{it}\left(\frac{P_{it}}{P_t}\right)^{-\theta_p} Y_t - mc_{it} P_t \left(\frac{P_{it}}{P_t}\right)^{-\theta_p} Y_t - \frac{\Psi_p}{2}\left(\frac{P_{it} - P_{it-1}}{P_{it-1}}\right)^2 P_t Y_t\right] \\
& + \beta \frac{1}{1+\pi_{t+1}} \frac{\partial}{\partial P_{it}}\left[P_{it+1}\left(\frac{P_{it+1}}{P_{t+1}}\right)^{-\theta_p} Y_{t+1} - mc_{it+1} P_{t+1}\left(\frac{P_{it+1}}{P_{t+1}}\right)^{-\theta_p} Y_{t+1}\right. \\
& \left. - \frac{\Psi_p}{2}\left(\frac{P_{it+1} - P_{it}}{P_{it}}\right)^2 P_{t+1} Y_{t+1}\right] = 0.
\end{aligned} \tag{69}$$

Rewrite in a cleaner form using $Y_{it} = (P_{it}/P_t)^{-\theta_p} Y_t$:

$$
\begin{aligned}
0 = (1-\theta_p) P_t^{\theta_p} P_{it}^{-\theta_p} Y_t + \theta_p \, mc_{it} \, P_t^{1+\theta_p} P_{it}^{-\theta_p - 1} Y_t - \Psi_p \left(\frac{P_{it} - P_{it-1}}{P_{it-1}} \right) \frac{P_t}{P_{it-1}} Y_t \\
+ \beta \frac{1}{1+\pi_{t+1}} \Psi_p \left(\frac{P_{it+1} - P_{it}}{P_{it}} \right) \frac{P_{it+1}}{P_{it}^2} P_{t+1} Y_{t+1}.
\end{aligned} \tag{70}
$$

In a symmetric equilibrium across varieties, we have:

$$
P_{it} = P_t, \qquad mc_{it} = mc_t, \qquad \pi_t = P_t - P_{t-1}
$$

Then (70) becomes:

$$
0 = (1-\theta_p) Y_t + \theta_p \, mc_t \, Y_t - \Psi_p \, \pi_t (1+\pi_t) \, Y_t + \beta \, \Psi_p \, \pi_{t+1} (1+\pi_{t+1})^2 \, \frac{Y_{t+1}}{1+\pi_{t+1}}
$$

This simplifies to:

$$
\begin{aligned}
&\Psi_p \, \pi_t (1+\pi_t) Y_t = (1-\theta_p) Y_t + \theta_p \, mc_t \, Y_t + \beta \Psi_p \, \pi_{t+1} (1+\pi_{t+1}) \, Y_{t+1} \\
\Rightarrow \quad &\pi_t (1+\pi_t) = \beta \, \pi_{t+1} (1+\pi_{t+1}) \frac{Y_{t+1}}{Y_t} + \frac{1-\theta_p}{\Psi_p} + \frac{\theta_p}{\Psi_p} mc_t
\end{aligned}
$$

Finally, linearizing around a zero-inflation steady-state (so that $\pi_t(1+\pi_t) \approx \pi_t$) and using $\mu_p = \frac{\theta_p}{\theta_p - 1}$, we obtain the discrete-time New Keynesian price Phillips curve:

$$
\pi_t = \beta \, \mathbb{E}_t \left[\pi_{t+1} \right] \frac{Y_{t+1}}{Y_t} + \frac{\theta_p}{\Psi_p} \left(mc_t - \mu_p^{-1} \right) \tag{71}
$$

References

Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., & Moll, B. (2022). Income and wealth distribution in macroeconomics: A continuous-time approach. *The review of economic studies*, *89*(1), 45–86.

Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics*, *109*(3), 659–684.

Auclert, A., Bardóczy, B., Rognlie, M., & Straub, L. (2021). Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. *Econometrica*, *89*(5), 2375–2408.

Bachmann, R., Baqaee, D., Bayer, C., Kuhn, M., Löschel, A., Moll, B., ... Schularick, M. (2024). What if? the macroeconomic and distributional effects for germany of a stop of energy imports from russia. *Economica*, *91*(364), 1157–1200.

Benmir, G., & Roman, J. (2022). The distributional costs of net-zero: A hank perspective. *Work. Pap., Lond. Sch. Econ. Political Sci., London*.

Huggett, M. (1993). The risk-free rate in heterogeneous-agent incomplete-insurance economies. *Journal of economic Dynamics and Control*, *17*(5-6), 953–969.

Kaplan, G., Moll, B., & Violante, G. L. (2018). Monetary policy according to hank. *American Economic Review*, *108*(3), 697–743.

Luetticke, R. (2021). Transmission of monetary policy with heterogeneity in household portfolios. *American Economic Journal: Macroeconomics*, *13*(2), 1–25.

Malmberg, S., Langot, F., Tripier, F., & Hairault, J.-O. (in press). The macroeconomic and redistributive effects of the french tariff shield. *Journal of Political Economy Macroeconomics*. Retrieved from https://malmberg.fr/Documents/TariffShield_paper.pdf

Pieroni, V. (2023). Energy shortages and aggregate demand: Output loss and unequal burden from hank. *European Economic Review*, *154*, 104428.