Solving Dynamic Models with Heterogeneous Agents and Aggregate Uncertainty with Dynare or Dynare++

Wouter J. DEN HAAN and Tarık Sezgin OCAKTAN *

September 22, 2011

Abstract

This paper shows how models with heterogeneous agents and aggregate uncertainty can be solved using Dynare or Dynare++ software that implements a perturbation approach. Using the explicit aggregation algorithm (XPA) to obtain aggregate laws of motion is possible by combining a Dynare program with a very simple Matlab program. We calculate and compare 1^{st} and 2^{nd}-order numerical solutions using both algorithms. These numerical procedures are also compared with the algorithm that solves the individual policy rules with a projection instead of a perturbation procedure. Finally, we discuss a procedure that efficiently chooses which cross-sectional moments to include as aggregate state variables when nonlinearities are important and the mean is not a sufficient statistic.

Key Words: numerical solutions, projection methods
JEL Classification: C63, D52

*den Haan: London School of Economics and CEPR, e-mail: wdenhaan@uva.nlLondon Business School, email: tocaktan@london.edu. This chapter is based on a joint work with Wouter den Haan. I would like to thank Michel Juillard, Christophe Cahn and Markus Kirchner for stimulating discussions and their useful comments .

1 Introduction

The most popular model in macroeconomics is the New Keynesian model. There are many quite sophisticated versions of this model, but most of them are based on the representative agent paradigm. If there are heterogeneous agents in the model, the degree of heterogeneity is limited, for example, to just having two representative agents, such as a patient and an impatient agent. The assumptions made to justify a representative agent framework, like perfect insurance of idiosyncratic risk, are believed to be unrealistic. Even if there are representative agent models that can describe the aggregate economy well, they obviously cannot explain any distributional aspects, although these are of interest to many economists. Fortunately, there are now numerous papers in which heterogeneity and idiosyncratic risk play a key role. There are even quite a few papers depicting both heterogeneity and aggregate risk, a combination that makes it more cumbersome to obtain a numerical solution. Nevertheless, the overwhelming majority of papers still rely on models with a representative agent.

One important reason for the ongoing dominance of models with a representative agent is without doubt the availability of user-friendly software, like Dynare, to solve and estimate these types of models. In principle, Dynare can handle many state variables and, thus, should be able to solve models with multiple agents. The problem is that in macroeconomic models we would like to have millions of agents and such a large number of agents is problematic, even for Dynare.[1] The standard approach is to approximate these millions of agents as a continuum, but Dynare

[1] An interesting question is whether using a relatively small number, say 100, is enough to approximate a macroeconomy with millions of agents. It would be cumbersome to use Dynare to solve models with 100 agents, but this is likely to be feasible at least for some models and for lower-order perturbation solutions.

requires a finite number of agents.

In this paper, we show how to use Dynare to solve models with (*i*) a continuum of heterogeneous agents and (*ii*) aggregate risk. To do this, two things are needed. First, a *mother* program is needed to solve for the laws of motion that describe the aggregate variables taking the solution of the individual policy rules as given. Second, a Dynare program is needed to solve for the individual policy rules taking the aggregate law of motion as given. This Dynare program, the `*.mod` file, needs to be able to read the coefficients of the aggregate law of motion. We show that this can be accomplished with an easy procedure for a Dynare program and with a somewhat more cumbersome procedure for a Dynare++ program.

Preston and Roca [16] were the first to suggest that one can use perturbation techniques to solve models with a continuum of heterogeneous agents and aggregate risk. den Haan and Rendahl [7] developed the XPA algorithm with which aggregate laws of motion are obtained by explicitly aggregating the individual policy rules, even when the individual policy rules are not linear. In the algorithm of den Haan and Rendahl [7], the individual policy rules can be obtained with any algorithm. In this paper, we show that if the individual policy rules are obtained with perturbation techniques, the XPA solution is identical to the one obtained by the algorithm of Preston and Roca [16]. The advantage of the XPA algorithm is the simplicity and transparency and this approach is used here to write the programs.

What, then, is required to be able to implement XPA with Dynare? The Dynare program itself that solves for the individual laws of motion would be almost a standard Dynare program. However, it differs from a standard Dynare program in two aspects. As explained above, the program needs to be able to incorporate the values of the coefficients of the aggregate laws of motion. The second modification related to XPA is that auxiliary policy rules are needed in order to explicitly aggregate higher-order individual policy functions. For example, if the model has a policy rule for the capital choice, k, then one needs an auxiliary policy rule for k^2 is needed if a second-order perturbation solution is used. This would result in simply adding one variable and one equation to the standard Dynare file. The two-part mother program would be very simple. The first part would contain a few steps of algebra to get the coefficients of the aggregate laws of motion from the individual laws of motion. The second part has to contain a procedure to make the laws of motion of the aggregate variables consistent with the laws of motion of the individual variables. For example, one might start with an aggregate policy rule, solve for the individual policy rule using perturbation techniques, obtain the aggregate law of motion by explicit aggregation of the individual policy rule, and iterate until the aggregate law of motion has converged. However, there are better, more efficient ways of obtaining consistency, and this paper will demonstrate a way to write the mother program and to do this.

To summarize the approach, we use this algorithm to generate first and second-order solutions. We then assess the accuracy and compare the solutions with those obtained with alternative algorithms. The first alternative algorithm uses the same Dynare program to solve for the individual policy rules, but obtains the aggregate law using the simulation procedure proposed by Krusell and Smith [14] instead of explicit aggregation. The mother program becomes somewhat more complicated because it has to include a simulation procedure, but the structure of the program remains the same. We also consider an algorithm in which the aggregate laws of motion are still solved by explicit aggregation but the individual policy rules are obtained using a projection method. Given that there is no standard software (yet) to implement projection methods, the procedure described is obviously much more accessible than alternative algorithms.

The main contributions of this paper are to (*i*) establish the link between the algorithm of Preston and Roca [16] and the XPA algorithm of Krusell and Smith [14], (*ii*) describe how Dynare and Dynare++ can be used to implement these algorithms, and (*iii*) compare the different numerical solutions and assess their accuracy. Finally, one more important issue addressed in this paper is to describe a procedure to limit the number of state variables. In several models considered so far, the computational burden turns out to be low because only a small set of moments, typically just the mean, is sufficient to capture the predictive information in the distribution, a property referred to as *approximate aggregation*.[2] This is not likely to be a universal property. The key for the approximate aggregation to hold is that the individual policy functions are close to being linear in the relevant part of the state space.[3] Linear policy rules are unlikely to be a realistic property in general and it is (hopefully) only a matter of time until we will be using models in which nonlinearities matter for aggregation. But if nonlinearities matter, then additional moments have to be used as state variables. Following the logic of the XPA algorithm implies that each nonlinear basis term used in the individual policy function gives rise to an additional cross-sectional moment in the set of state variables. We show how to reduce the dimension of this set by using combinations of these state variables. This contribution is likely to be more important when the individual policy rules are solved with a projection procedure than with a perturbation procedure, given that keeping the cost of projection methods low with a large number of state

[2]See Krusell and Smith [15].

[3]For example, nonlinearities do not matter for the very poor agents because their behavior is not important for the aggregate anyway.

variables is tricky.

2 The Problem

The model applied in this paper is a standard heterogeneous agents model with aggregate productivity shocks similar to the one discussed in Aiyagari [2], Bewley [4] and Huggett [9]. It describes a simple exchange economy with incomplete markets, aggregate uncertainty and an infinite number of agents. The source of heterogeneity comes from the assumption that there are idiosyncratic income shocks, which are partially uninsurable.

Problem for the individual agent The economy is represented by a stochastic growth model with a continuum of individuals indexed by $i \in [0, 1]$. The individual agents are characterized as facing an idiosyncratic unemployment risk. All agents are ex ante identical, however there is ex post heterogeneity due to incomplete insurance markets and borrowing constraints. Every quarter, individuals differ from each other through their asset holdings and employment opportunities. In order to transfer their ressources over time, agents can only control their capital holdings. We simplify the model with respect to the one described in Preston and Roca [16], and assume that an employed agent earns a wage rate of w, while an unemployed agent has no income. However, agents can insure themselves, at least partially, against employment risks by building up a capital stock. To insure satisfaction of intertemporal budget constraints, capital holdings are restricted by a borrowing limit $b \geq 0$, ensuring the repayment of loans and the absence of Ponzi schemes.

Agent i's maximization problem is given by

$$\max_{\{c_{it},a_{i,t+1}\}} E_t \sum_{t=0}^{\infty} \beta^t \frac{c_{it}^{1-\gamma}-1}{1-\gamma}$$
$$\text{s.t. } c_{it} + a_{i,t+1} = r(k_t, l_t, z_t)a_{it} + w(k_t, l_t, z_t)e_{it}\bar{l} + (1-\delta)a_{it} \tag{1}$$
$$a_{it} \geq b$$

The utility function is characterized by a standard constant relative risk aversion (CRRA) utility function with risk aversion parameter $\gamma > 0$. This function is twice continuously differentiable, increasing, and concave in the level of consumption of household i, c_{it}. Let a_{it} be the agent's beginning-of-period asset holdings, $a_{i,t+1}$ is the next period level of asset constrained, $\bar{l}$ is the time endowment, β the subjective discount rate, $0 < \delta < 1$ is the depreciation rate. r_t and w_t is the interest rate and wage rate, respectively.

Each agent faces partially insurable labor market income risk and is endowed with one unit of time. This endowment is transformed into labor input according to $l_{it} = e_{it}\bar{l}$. The stochastic employment opportunity e_{it} follows an autoregressive process of the form

$$e_{i,t+1} = (1-\rho_e)\mu_e + \rho_e e_{it} + \varepsilon_{i,t+1}^e \ , \ \varepsilon_t^e \sim \mathcal{N}(0, \sigma_e^2) \tag{2}$$

where $0 < \rho_e < 1$, $\mu_e > 0$ and $\varepsilon_{i,t+1}^e$ a bounded *i.i.d.* disturbance with mean and variance $(0, \sigma_e^2)$.

Problem for the firm Markets are competitive and the production technology of the firm is characterized by a Cobb-Douglas production function. Let k_t and l_t stand for per capita capital and the employment rate, respectively. Per capita aggregate output takes as inputs the aggregate capital stock and labor supply

$$y_t = z_t k_t^{\alpha} l_t^{1-\alpha}$$

In the original model, as described in Krusell and Smith [14], aggregate productivity, z_t, is an exogeneous stochastic process that can take on two values, which are interpreted as the good and bad states of the economy. For reasons that will be detailed later, we transform the discrete space to a continuous support by assuming that the aggregate technology shock z_t, common to all households, satisfies

$$z_{t+1} = (1-\rho_z)\mu_z + \rho_z z_t + \varepsilon_{t+1}^z \tag{3}$$

where $0 < \rho_z < 1$, $\mu_z > 0$ and ε_{t+1}^z a bounded *i.i.d.* disturbance with mean and variance $(0, \sigma_z^2)$. We assume that firms rent their factors of production from households in competitive spot markets. These aggregate inputs imply market interest and wage rates equal to

$$r(k_t, l_t, z_t) = \alpha z_t \left(\frac{k_t}{l_t}\right)^{\alpha-1} \tag{4}$$

$$w(k_t, l_t, z_t) = (1-\alpha) z_t \left(\frac{k_t}{l_t}\right)^{\alpha} \tag{5}$$

In order to solve the optimization program given by expression (1) agents must forecast future prices. Under the maintained assumptions (l_t, z_t) follow an exogeneous stochastic process. Therefore in order to forecast future wage and rental rates, agents must know the stochastic process that describes the evolution of the aggregate capital stock. However, the stochastic properties of the aggregate capital stock depend on the distribution of capital holdings in the population. As a consequence, the whole capital distribution becomes a state variable. In a setup with a continuum of agents, capital distribution is a function, which cannot be used as an argument of the individual policy rules. Krusell and Smith [14] propose to summarize this distribution by a discrete and finite set of moments. For instance, if we take into consideration only the first order moments, the law of motion for aggregate capital, k_{t+1}, is given by

$$k_{t+1} = \zeta_0 + \sum_{i=1}^{I} \zeta_i M(i) + \zeta_{I+1} z_t \tag{6}$$

where $M(i)$ is the cross-sectional average of a^i with $k = M(1)$, and s is a vector containing the aggregate state variables.

Equilibrium An equilibrium for this benchmark model then consists of the following

- *Optimality:* given (4), (5), and (6) the household decision rules solve the maximization problems given in expression (1).

- *Factor prices:* wage and rental rates are factor marginal productivities and are determined by expressions (4) and (5), respectively.

- *Aggregation:* Factor inputs are generated by aggregation over agents with $k_t \equiv \int a_{it} di$ and $l_t \equiv \int l_{it} di$. A transition law for the cross-sectional distributions of capital, that is consistent with the investment policy function. Let k_t represent the beginning-of-period aggregate capital with the following transition law

$$k_{t+1} = \zeta_0 + \sum_{i=1}^{I} \zeta_i M(i) + \zeta_{I+1} z_t$$

This law of motion reveals an advantage of working with a continuum of agents. Because we apply a law of large numbers, we know that conditional on z_t, there is no uncertainty in determining k_{t+1}

Penalty Function Perturbation methods cannot directly be applied to models with occasionally-binding inequality constraints. In order to deal with the non-negativity constraint for asset holdings, we replace the borrowing constraint by a penalty function (Judd [10], pp. 123). The basic idea of this approach is that we allow anything to be feasible, but we change the objective function so that it has undesirable consequences if the constraints are violated. By using this approach we have converted the original problem, given in expression (1) into an optimization problem with only equality constraints, which allows us to apply a standard perturbation method. Recent contributions that use this approach for heterogeneous agents models include Kim et al. [11] and Preston and Roca [16]. There are many ways to represent the penalty function. We focus on three specifications from the recent literature on heterogeneous agents models. We will use the specification described in den Haan and de Wind [8]

$$\mathbf{P}(a_{i,t+1}) = \frac{\eta_1}{\eta_0} \exp\big(- \eta_0(a_{i,t+1} + b)\big) - \eta_2(a_{i,t+1} + b) \tag{7}$$

Alternative specification for the penalty function are discussed in the next chapter of the present thesis. These functions have the property that as individual asset holdings approach the borrowing constraint b the interior function approaches infinity. The first derivative with respect to $a_{i,t+1}$ of expression (7) is given by

$$p(a_{i,t+1}) \quad = \quad -\eta_1 \exp\big(-\eta_0(a_{i,t+1}+b)\big) - \eta_2 \tag{8}$$

$$\tag{9}$$

The modified model In order to solve this model by perturbation, we need to transform the optimization problem, given in expression (1), so that we get rid of the borrowing constraint. In fact, the inequality constraint is the reason of non-differentiability. One way of avoiding this problem is to replace the inequality constraint with a function, so that the individual problem will be

$$\max_{\{c_{it},a_{i,t+1}\}} E_t \sum_{t=0}^{\infty} \beta^t \frac{c_{it}^{1-\gamma}-1}{1-\gamma} - \phi \mathbf{P}(a_{i,t+1})$$
$$\text{s.t. } c_{it} + a_{i,t+1} = r(k_t, l_t, z_t)a_{it} + w(k_t, l_t, z_t)e_{it}\bar{l} + (1-\delta)a_{it} \tag{10}$$

This maximization problem faced by an individual in the economy can be represented as a dynamic programming problem in which a_{it}, e_{it}, Γ_t and z_t are the state variables and c_{it}, a_{it+1} are the decision variables. The variable Γ_t represents the cross-sectional distribution of assets in the economy. The optimality equation for this model is given by

$$
\begin{aligned}
V(a_i, e_i; z, \Gamma) \quad &= \quad \max_{\{c_i, a_i'\}} \left\{ \frac{c_i^{1-\gamma} - 1}{1-\gamma} + \beta E\big[V(a_i', e_i'; z', \Gamma') - \phi \mathbf{P}(a_i')\big] \right\} \\
\text{s.t} \quad & (1-\delta)a_i + r(k,l,z)a_i + w(k,l,z)e_i\bar{l} - c_i - a_i' \geq 0 \\
& z' = (1-\rho_z)\mu_z + \rho_z z + \varepsilon'^z \\
& e_i' = (1-\rho_e)\mu_e + \rho_e e_i + \varepsilon_i'^e \\
& \Gamma' = H(\Gamma, z)
\end{aligned}
\tag{11}
$$

where function $V(\cdot)$ is the value function of a type i household, r and w are functions that describe the factor prices, and Γ denotes the distribution of capital holdings in the population.

An important feature of the problem, as stated in expression (11), is that the cross-sectional distribution, Γ, is a state variable. Even if the economic agents will at some point require a limited set of information about this distribution, they will always be faced with the situation in which they will require an approximation of Γ. In the solution algorithm as proposed in Krusell and Smith [14], the individual agents use only the first order moments of Γ to compute current and future market prices. However, since they need an approximation of Γ to compute the first order moment, M_a, we cannot replace the cross-sectional distribution, Γ, with the first order moment, M_a, from the set of state variables.

The solution, as will be described later, differs from the traditional approach in this particular regard. Now we no longer rely on the entire distribution Γ, but use directly the approximated decision rules to compute the law of motion for the first order moment, M_a. As can be seen in expression (13), the use of the explicit aggregation algorithm, allows us to replace the cross-sectional distribution, Γ, with the first order moment, M_a, in the set of state variables.

The reader should also note that the dimension of the state space is endogeneously determined by the algorithm, unlike in the case of the traditional approach, where the order of cross-sectional moments that are used is determined by the economist. If the dynamic optimization problem is solved using a second order polynomial approximation (or a second order Taylor expansion), then the second order cross-sectional moments will become additional state variables. In order to illustrate this, we will present the two Bellman equations for the cases of first- and second order approximation of the decision rules. We have to keep in mind that simply substituting Γ by k is an abuse of notation, since the real model always depends on the evolution of the whole distribution Γ. However, in our implementation, the law of motion of the moments for the cross-sectional distribution are approximated by directly using the individual decision rules instead of the whole distribution, Γ. As a consequence, we can substitute variable Γ by the relevant moments in the list of state variables. This means that if we use a first order approximation to the decision rules, the original model (12) will be approximated by the model (13). By doing this, we have reduced the large dimensional state space $(a_i, e_i; z, \Gamma)$ to a more tractable 4-dimensional state space $(a_i, e_i; z, k)$. When we use a first order approximation, the problem can be represented as follows

$$\begin{aligned} V(a_i, e_i; z, k) \quad &= \quad \max_{\{c_i, a_i'\}} \left\{ \frac{c_i^{1-\gamma} - 1}{1-\gamma} + \beta E\big[V(a_i', e_i'; z', k') - \phi \mathbf{P}(a_i')\big] \right\} \\ &\text{s.t} \quad (1-\delta)a_i + r(k,l,z)a_i + w(k,l,z)e_i\bar{l} - c_i - a_i' \geq 0 \\ &\qquad z' = (1-\rho_z)\mu_z + \rho_z z + \varepsilon'^z \\ &\qquad e_i' = (1-\rho_e)\mu_e + \rho_e e_i + \varepsilon_i'^e \\ &\qquad k' = \zeta_0 + \zeta_1 k + \zeta_2 z \end{aligned} \tag{12}$$

If we use a second order approximation to the individual decision rules, the original problem (12) will be approximated by the model (13). In this case we have reduced a large dimensional state space to a 6-dimensional one. By doing this, we add the second order moments of the cross-sectional distribution to the list of state variables. The optimality equation for the second order approximated model is

$$V(a_i, e_i; z, k, M_{ae}, M_{a^2}) = \max_{\{c_i, a_i'\}} \left\{ \frac{c_i^{1-\gamma} - 1}{1-\gamma} + \beta E\big[V(a_i', e_i'; z', k', M_{ae}', M_{a^2}') - \phi \mathbf{P}(a_i')\big] \right\} \tag{13}$$

subject to the following

$$\begin{aligned}
c_i &= (1-\delta)a_i + r(k,l,z)a_i + w(k,l,z)e_i\bar{l} - a_i' \\
z' &= (1-\rho_z)\mu_z + \rho_z z + \varepsilon'^z \\
e_i' &= (1-\rho_e)\mu_e + \rho_e e_i + \varepsilon_i'^e \\
k' &= \zeta_0 + \zeta_1 k + \zeta_2 z + \zeta_3 M_{ae} + \zeta_4 M_{a^2} + \zeta_5 k^2 + \zeta_6 z^2 + \zeta_7 kz \\
M_{a^2}' &= \bar{\zeta}_0 + \bar{\zeta}_1 k + \bar{\zeta}_2 z + \bar{\zeta}_3 M_{ae} + \bar{\zeta}_4 M_{a^2} + \bar{\zeta}_5 k^2 + \bar{\zeta}_6 z^2 + \bar{\zeta}_7 kz \\
M_{ae}' &= \tilde{\zeta}_0 + \tilde{\zeta}_1 k + \tilde{\zeta}_2 z + \tilde{\zeta}_3 M_{ae} + \tilde{\zeta}_4 M_{a^2} + \tilde{\zeta}_5 k^2 + \tilde{\zeta}_6 z^2 + \tilde{\zeta}_7 kz
\end{aligned}$$

3 Recursion

Since there is no analytical solution to this model, we have to rely on numerical approximation methods. To give a general overview of the algorithm used to solve the present model, we start by assuming that the agents only use the first I moments of the wealth distribution in order to perceive current and future prices. This means that agents have access to the law of motion for the capital stock. Given this law of motion, each agent can compute his optimal choice. The iterative procedure used to approximate this aggregate law of motion will be decribed in the next section.

The general algorithm can be described as follows: (1) Select the order I of the approximation (and hence of the moments that are used as state variables), (2) choose an initial parameterization for the aggregate law of motion, (3) solve the individual problem, (4) use the individual decision rules to update the aggregate law of motion given in step (2), (5) iterate until convergence.

[Figure 1 about here.]

In order to study the sensitivity of the algorithm to the choice of various solution methods, we solve the individual problem using two methods: a local and a global method. Furthermore, we compare the updating procedure of the aggregate law of motion using two different approaches. These procedures will be explained in more detail in the following sections.

The overall algorithm is summarized in Figure 3, while the approximation for the parameters of the aggregate law of motion, the ζ's, are presented in Table 5.

[Table 1 about here.]

[Figure 2 about here.]

3.1 A standard solution method

The basic idea of the Krusell and Smith [14] algorithm relies on summarizing the cross-sectional distribution of capital and employment status with a limited set of moments. The algorithm specifies a law of motion for these moments and finds the approximating function using a simulation procedure. Given a set of individual policy rules, a time series of cross-sectional moments is generated and new laws of motion for the aggregate moments are estimated using the simulated data.

However, this iterative procedure relies on simulation which generates two types of sampling variation. One is related to the fact that we use a finite set of agents instead of a continuum of agents. The other type generates from the aggregate shock. Furthermore, since the method relies on simulated data to obtain numerical solutions, it has two disadvantages. First, by introducing sampling noise the policy functions themselves become stochastic. This effect can be reduced by using long time series, but sampling noise disappears at a slow rate. Second, the values of the state variables used to find the best fit for the aggregate law of motion are endogeneous and are typically clustered around their mean.

3.2 Explicit aggregation solution method

An alternative approach is to obtain the law of motion, which describes aggregate behaviour by explicitly aggregating the individual policy rule. This approach is simpler than the methods that rely on parameterization of the cross-sectional distribution (as in Algan, Allais and den Haan [3]), and it is not computationally intensive (as in Krusell and Smith [14] and den Haan [6]) as the methods that rely on simulation and regression. Another issue is the choice of the order of moments (of the cross-sectional distribution) that have to be included into the solution algorithm. In the previous method, this choice is made externally by the economist, depending on the forecasting power of the moments in question.

Instead of using the approximated decision rules for the construction of the simulated cross-sectional distribution, the approach that is adopted here relies on the decision rules for directly evaluating the moments of the distribution without simulating the entire distribution (den Haan and Rendahl [7]). This is done by *explicitly aggregating* the economy conditional on the properties of the approximated decision rules. As a consequence, the moments of the distribution that are included in the solution algorithm depend on the properties of the polynomial that approximates the decision rule. For instance, the aggregation of the first order (linear) approximation of the economy will only take into account the first order moments of the cross-sectional distribution. A second order (nonlinear) approximation, however, will take into account the first *and* second order moments of the distribution.

4 Explicit aggregation

The explicit aggregation algorithm for this particular model. is closely related to den Haan and Rendahl [7]. A detailed description of the algorithm is laid out below.

4.1 First-order approximation

In order to present the algorithm, we consider, for the moment, the first order approximation of a model with individual and aggregate uncertainty. Consider the following simple model in which all agents are identical, except for their initial capital stock, a_{i0}, and their employment status, e_{it}. The individual decision rule for next periods asset holding is given by

$$a_{i,t+1} = \theta_0 + \theta_1 a_{it} + \theta_2 e_{it} + \theta_3 z_t + \theta_4 M_{a,t} \tag{14}$$

From equation (14) it can be seen that the individual policy rule is assumed to be a polynomial of order 1 in a, e, z and M_a. Using monomials as in equation (14) simplifies the exposition; den Haan and Rendahl [7] show that other basis functions can be used as well, such as orthogonal polynomials or B-splines.

The key issue for heterogeneous agents is to establish a law of motion for aggregate capital, k. Given the expression in equation (14), this aggregate law of motion for k follows directly from aggregating the individual policy rule over i. That is,

$$\int_0^1 a_{i,t+1} di = \theta_0 + \theta_1 \int_0^1 a_{it} di + \theta_2 \int_0^1 e_{it} di + \theta_3 z_t + \theta_4 M_{a,t} \tag{15}$$

Note that in order to apply this method, we need an expression for the average level of the capital stock and not, for example, for the average of the log capital stock. Consequently, the left-hand side of equation (14) has to be equal to the level of a_{it+1}, as is the case in our example.

Lemma 1. *The economy-wide average for the employment status at a given time period is constant, with*

$$\int_0^1 e_i di = \mu_e \tag{16}$$

This is a direct consequence of the assumption that the employment status is an exogeneous autoregressive process of order 1.

Proof. Here is the proof for expression 16:

Assume that the law of motion for employment status follows an autoregressive process of order 1, such that

$$e_{i,t+1} \quad = \quad (1-\rho_e)\mu_e + \rho_e e_{it} + \epsilon_{i,t+1}$$

By integrating the previous expression over all individuals yields the economy-wide average for next periods employment status

$$\int_0^1 e_{i,t+1} di \quad = \quad (1-\rho_e)\mu_e + \rho_e \int_0^1 e_{it} di + \int_0^1 \epsilon_{i,t+1} di$$

Using $\int_0^1 \epsilon_i di \simeq 0$ and $\int_0^1 e_{i,t+1} di \simeq \int_0^1 e_{it} di$ the previous expressions reduces to

$$\int_0^1 e_i di = \mu_e$$

In order to motivate $\int_0^1 \epsilon_i di \simeq 0$, we could rewrite the previous expression in discrete time and use the law of large numbers

$$\text{As } N \to 0 \ , \ \frac{1}{N} \sum_{i=1}^{N} \epsilon_i \to 0 \text{ since } \epsilon_i \sim \mathcal{N}(0, \sigma_e^2)$$

□

Using expression (16) and the definition for the first order moment of the cross-sectional distribution for wealth, $M_a = \int_0^1 a_i di$, we have

$$M_{a,t+1} = (\theta_0 + \theta_2 \mu_e) + (\theta_1 + \theta_4) M_{a,t} + \theta_3 z_t \tag{17}$$

4.2 Second-order approximation

Equation (15) is obtained in a rather straightforward way as we have only first order terms in equation (14). As soon as we have higher order polynomials in equation (14), we will have higher order cross-sectional moments in equation (15). In other words, we need to include higher order terms as inputs for predicting k_{t+1}.

However, this raises additional problems, since it implies that including higher-order cross-sectional moments requires additional aggregate laws of motions to predict those. This is because these higher order terms will appear as arguments in next period's policy function. In order to illustrate this, let us take the second order approximation of the individual policy function, which is given by

$$\begin{aligned} a_{i,t+1} &= \theta_0 + \theta_1 a_{it} + \theta_2 e_{it} + \theta_3 z_t + \theta_4 M_{a,t} + \theta_5 M_{ae,t} + \theta_6 M_{a^2,t} \\ &\quad + \theta_7 a_{it}^2 + \theta_8 e_{it} a_{it} + \theta_9 e_{it}^2 + \theta_{10} z_t a_{it} + \theta_{11} z_t e_{it} \\ &\quad + \theta_{12} M_{a,t} a_{it} + \theta_{13} M_{a,t} e_{it} + \theta_{14} M_{a,t} z_t + \theta_{15} M_{a,t}^2 \end{aligned} \tag{18}$$

In equation (18), we see the appearance of second order terms. For this particular model, we are especially interested in a_{it}^2 and $a_{it}e_{it}$. As noted previously, the presence of these terms requires aggregate laws of motions to predict these moments.

Because of the cross-terms of the individual state variables a_i and e_i appear additional state variables, M_{ae} and M_{a^2}. By definition, the aggregation of these second-order terms are equal to the uncentered second-order moments

$$\int_0^1 a_i^2 di = M_{a^2} \tag{19}$$

$$\int_0^1 e_i a_i di = M_{ae} \tag{20}$$

One way to get a policy rule for a_{it+1}^2 is to use the one that is implied by the approximation of a_{it+1} given in equation (14). However by taking the square of equation (14), we will end up with a polynomial of order higher than 2, which means that additional moments would have to be added. Then additional policy rules would be needed to predict these moments, which in turn would introduce more state variables. Without modification, a solution based on explicit aggregation requires including an infinite number of moments as state variables whenever the order of approximation is higher than one.

The key approximating step of the algorithm described in den Haan and Rendahl [7] is to break this infinite regress problem and to construct separate approximations to the policy rules for a_{it+1}^2 and $a_{it+1}e_{it+1}$.

For a_{it+1}^2, we use an approximation, which has the same form as equation (18)

$$a_{it+1}^2 = \mathcal{P}_2(a_{it}, e_{it}, z_t, k_t, M_{ae}, M_{a^2}; \bar{\Theta}) \tag{21}$$

For $a_{it+1}e_{it+1}$, we have to be more careful

$$\begin{aligned} a_{it+1}e_{it+1} &= a_{it+1}\big((1-\rho_e) + \rho_e e_{it} + \varepsilon_{it+1}\big) \\ &= (1-\rho_e)a_{it+1} + \rho_e a_{it+1}e_{it} + a_{it+1}\varepsilon_{it+1} \end{aligned} \tag{22}$$

In order to compute the last expression, we need to approximate $a_{it+1}e_{it}$, and for doing this we use again a similar approximation as in equation (14)

$$a_{it+1}e_{it} = \mathcal{P}_2(a_{it}, e_{it}, z_t, k_t, M_{ae}, M_{a^2}, k_{t+1}; \tilde{\Theta}) \tag{23}$$

The coefficients of the approximating functions in equations (18), (21) and (23) can now be solved for using projection methods or perturbation methods. Once we obtain Θ, $\bar{\Theta}$ and $\tilde{\Theta}$, we can deduce, by explicit aggregation, the laws of motions for k_{t+1}, $M_{a^2,t+1}$ and $M_{ae,t+1}$.

Law of motion for M_{a^2}. In order to compute the law of motion for M_{a^2} given in expression (19), we introduce an auxiliary variable $\mathbf{q}_{i,t+1} = a^2_{i,t+1}$. The decision rule for this additional control variable will be approximated in a similar way as for the other control variables, which will yield an approximation for $a^2_{i,t+1}$:

$$\begin{aligned} \mathbf{q}_{i,t+1} &= \theta^{\mathbf{q}}_0 + \theta^{\mathbf{q}}_1 a_{it} + \theta^{\mathbf{q}}_2 e_{it} + \theta^{\mathbf{q}}_3 z_t + \theta^{\mathbf{q}}_4 M_{a,t} + \theta^{\mathbf{q}}_5 M_{ae,t} + \theta^{\mathbf{q}}_6 M_{a^2,t} \\ &\quad + \theta^{\mathbf{q}}_7 a^2_{it} + \theta^{\mathbf{q}}_8 e_{it} a_{it} + \theta^{\mathbf{q}}_9 e^2_{it} + \theta^{\mathbf{q}}_{10} z_t a_{it} + \theta^{\mathbf{q}}_{11} z_t e_{it} \\ &\quad + \theta^{\mathbf{q}}_{12} M_{a,t} a_{it} + \theta^{\mathbf{q}}_{13} M_{a,t} e_{it} + \theta^{\mathbf{q}}_{14} M_{a,t} z_t + \theta^{\mathbf{q}}_{15} M^2_{a,t} \end{aligned} \tag{24}$$

Law of motion for M_{ae}. For the aggregate law of motion in expression (20) we use the following

$$M_{ae,t+1} = (1-\rho_e)\mu_e M_{a,t+1} + \rho_e \int_0^1 \mathbf{p}_i di \tag{25}$$

where $\mathbf{p}_i$ is an auxiliary variable that approximates the following expression $\mathbf{p}_i = a_{i,t+1} e_{it}$:

$$\begin{aligned} \mathbf{p}_{i,t+1} &= \theta^{\mathbf{p}}_0 + \theta^{\mathbf{p}}_1 a_{it} + \theta^{\mathbf{p}}_2 e_{it} + \theta^{\mathbf{p}}_3 z_t + \theta^{\mathbf{p}}_4 M_{a,t} + \theta^{\mathbf{p}}_5 M_{ae,t} + \theta^{\mathbf{p}}_6 M_{a^2,t} \\ &\quad + \theta^{\mathbf{p}}_7 a^2_{it} + \theta^{\mathbf{p}}_8 e_{it} a_{it} + \theta^{\mathbf{p}}_9 e^2_{it} + \theta^{\mathbf{p}}_{10} z_t a_{it} + \theta^{\mathbf{p}}_{11} z_t e_{it} \\ &\quad + \theta^{\mathbf{p}}_{12} M_{a,t} a_{it} + \theta^{\mathbf{p}}_{13} M_{a,t} e_{it} + \theta^{\mathbf{p}}_{14} M_{a,t} z_t + \theta^{\mathbf{p}}_{15} M^2_{a,t} \end{aligned} \tag{26}$$

Proof. In order to compute the aggregate law of motions for the second order term

M_{ae} given in expression (20), we proceed as follows

$$\begin{aligned} a_{i,t+1}e_{i,t+1} &= a_{i,t+1}[(1-\rho_e)\mu_e + \rho_e e_{it} + \epsilon_{i,t+1}] \\ &= (1-\rho_e)\mu_e a_{i,t+1} + \rho_e a_{i,t+1}e_{it} + a_{i,t+1}\epsilon_{i,t+1} \end{aligned}$$

Integrating this over i yields

$$\int_0^1 a_{i,t+1}e_{i,t+1}di = (1-\rho_e)\mu_e \int_0^1 a_{i,t+1}di + \rho_e \int_0^1 a_{i,t+1}e_{it}di + \int_0^1 a_{i,t+1}\epsilon_{i,t+1}di$$

By using $\int_0^1 a_{i,t+1}e_{i,t+1}di = M_{ae,t+1}$, $\int_0^1 a_{i,t+1}di = M_{a,t+1}$, $\int_0^1 a_{i,t+1}\epsilon_{i,t+1}di = 0$ we can derive the law of motion for the second order moment M_{ae}

$$M_{ae,t+1} = (1-\rho_e)\mu_e M_{a,t+1} + \rho_e \int_0^1 \mathbf{p}_i di \tag{27}$$

where $\mathbf{p}_i$ is an auxiliary variable that approximates the following expression $\mathbf{p}_i = \int_0^1 a_{i,t+1}e_{it}di$. $\square$

Lemma 2. *The economy-wide average of the square of the employment status is a constant*

$$\int_0^1 e_i^2 di = \frac{1-\rho_e}{1-\rho_e^2}\left[(1-\rho_e)\mu_e^2 + 2\mu_e^2\rho_e + 2\mu_e M_\epsilon + M_{\epsilon^2}\right] \tag{28}$$

This is a direct consequence of the assumption that the employment status is an exogeneous autoregressive process of order 1.

Proof. Here is the proof for expression (28). Assume that the law of motion for employment status follows an autoregressive process of order 1, such that

$$e_{i,t+1} = (1-\rho_e)\mu_e + \rho_e e_{it} + \epsilon_{i,t+1}$$

Taking the square of the previous expression gives

$$\begin{aligned} e_{i,t+1}^2 &= \Big[(1-\rho_e)\mu_e + \rho_e e_{it} + \epsilon_{i,t+1}\Big]^2 \\ &= [(1-\rho_e)\mu_e]^2 + 2(1-\rho_e)\mu_e\rho_e e_{it} + 2\rho_e e_{it}\epsilon_{i,t+1} \\ &\quad + 2(1-\rho_e)\mu_e\epsilon_{i,t+1} + \rho_e^2 e_{it}^2 + \epsilon_{i,t+1}^2 \end{aligned}$$

Integrating the previous expression over i yields

$$\begin{aligned} \int_0^1 e_{i,t+1}^2 di &= [(1-\rho_e)\mu_e]^2 + 2(1-\rho_e)\mu_e\rho_e \int_0^1 e_{it} di + 2\rho_e \int_0^1 e_{it}\epsilon_{i,t+1} di \\ &\quad + 2(1-\rho_e)\mu_e \int_0^1 \epsilon_{i,t+1} di + \rho_e^2 \int_0^1 e_{it}^2 di + \int_0^1 \epsilon_{i,t+1}^2 di \end{aligned}$$

Using the following equalities, $\int_0^1 e_{it} di = \mu_e$, $\int_0^1 e_{it}\epsilon_{i,t+1} di = 0$, $\int_0^1 \epsilon_{i,t+1} di = M_\epsilon$, $\int_0^1 e_{it}^2 di = cst$ and $\int_0^1 \epsilon_{i,t+1}^2 di = M_{\epsilon^2}$, we get

$$\begin{aligned} \int_0^1 e_{i,t+1}^2 di &= [(1-\rho_e)\mu_e]^2 + 2(1-\rho_e)\mu_e^2\rho_e \\ &\quad + 2(1-\rho_e)\mu_e M_\epsilon + \rho_e^2 \int_0^1 e_{it}^2 di + M_{\epsilon^2} \end{aligned}$$

Assume $\int_0^1 e_{i,t+1}^2 di = \int_0^1 e_{it}^2 di$, we get

$$(1-\rho_e)\int_0^1 e_{it}^2 di = [(1-\rho_e)\mu_e]^2 + 2(1-\rho_e)\mu_e^2\rho_e + 2(1-\rho_e)\mu_e M_\epsilon + M_{\epsilon^2}$$

Hence

$$\int_0^1 e_i^2 di \quad = \quad \frac{1-\rho_e}{1-\rho_e^2}\left[(1-\rho_e)\mu_e^2 + 2\mu_e^2\rho_e + 2\mu_e M_\epsilon + M_{\epsilon^2}\right]$$

Since $M_\epsilon = 0$ and $M_{\epsilon^2} = \sigma_e^2$, we have[4]

$$\int_0^1 e_i^2 di \quad = \quad \frac{1-\rho_e}{1-\rho_e^2}\left[(1-\rho_e)\mu_e^2 + 2\mu_e^2\rho_e + \sigma_e^2\right]$$

□

5 Simulation-based approach

The simulation-based algorithm for this particular model. is closely related to Krusell and Smith [14]. A detailed description of the algorithm is laid out below.

5.1 First-order approximation

Using the solution to the decision rules from a first-order approximation we get

$$a_{i,t+1} = \theta_0 + \theta_1 a_{it} + \theta_2 e_{it} + \theta_3 z_t + \theta_4 M_{a,t}$$

This decision rule is used to simulate the economy for N agents and T time periods. Using the simulated time series a_{it} for $i = 1, \ldots, N$ and $t = 1, \ldots, T$, we compute

[4]Since $\epsilon \sim \mathcal{N}(0, \sigma_e^2)$. The moments of ϵ are given by $E(\epsilon_i) = \frac{1}{N}\int_0^N \epsilon_i di \to 0$ as $N \to \infty$. The second order moment is given by $E(\epsilon_i^2) = \frac{1}{N}\int_0^N \epsilon_i^2 di \to \sigma_e^2$ as $N \to \infty$.

the mean of the distribution

$$M_{a,t} = \frac{1}{N}\sum_{i=1}^{N} a_{it} \ , \forall t = 1, \ldots, T \tag{29}$$

Regressing $M_{a,t+1}$ on a constant, as well as on M_{at} and z_t, will yield updated values for the aggregate law of motion

$$M_{a,t+1} = \hat{\zeta}_0 + \hat{\zeta}_1 M_{a,t} + \hat{\zeta}_2 z_t \tag{30}$$

5.2 Second-order approximation

Using the solution to the decision rules from a second-order approximation we get

$$\begin{aligned} a_{i,t+1} \quad = \quad & \theta_0 + \theta_1 a_{it} + \theta_2 e_{it} + \theta_3 z_t + \theta_4 M_{a,t} + \theta_5 M_{ae,t} + \theta_6 M_{a^2,t} \\ & + \theta_7 a_{it}^2 + \theta_8 e_{it} a_{it} + \theta_9 e_{it}^2 + \theta_{10} z_t a_{it} + \theta_{11} z_t e_{it} \\ & + \theta_{12} M_{a,t} a_{it} + \theta_{13} M_{a,t} e_{it} + \theta_{14} M_{a,t} z_t + \theta_{15} M_{a,t}^2 \end{aligned} \tag{31}$$

In a way similar to the first-order solution, we will simulate the economy but this time using second-order approximation to the decision rules. Using the simulated time series for a_{it}, with $i = 1, \ldots, N$ and $t = 1, \ldots, T$, we compute the mean of the distribution

$$M_{a,t} = \frac{1}{N}\sum_{i=1}^{N} a_{it} \ , \forall t = 1, \ldots, T \tag{32}$$

Doing the same regression as for (30), we will find updated values for the coefficient of the aggregate law of motion.

6 Solution method

6.1 Perturbation method

In the perturbation method we compute the deterministic steady state of the model and then use an nth order Taylor expansion around this value.

Similarly, perturbation methods are typically performed on the optimality conditions, which are given by[5]

$$E_t f(y_{t+1}, y_t, y_{t-1}, \epsilon_t) = 0 \tag{33}$$

where $\epsilon_t = \sigma e_t$ and $E(\epsilon_t) = 0$, $E(\epsilon_t \epsilon_t') = \sigma^2 \Sigma_e$, and $E(\epsilon_t \epsilon_\tau') = 0$ for $t \neq \tau$. The policy function of this model is given by

$$y_t = h(y_{t-1}, \epsilon_t, \sigma)$$

For illustration purposes, we use the first order perturbation method to solve for the decision rules. This section follows closely Preston and Roca [16].

The equilibrium for this model is determined by the following optimality conditions:

$$\begin{aligned} c_{it}^{-\gamma} &= \beta E_t \Big[c_{i,t+1}^{-\gamma} \big(r(k_{t+1}, \bar{l}, z_{t+1}) + 1 - \delta \big) + p(a_{i,t+1}) \Big] \\ a_{i,t+1} &= (1-\delta) a_{it} + r(k_t, \bar{l}, z_t) + w(k_t, \bar{l}, z_t) \bar{l} - c_{it} \end{aligned}$$

[5] Notation used is in line with the documentation of the Dynare project.

and the exogeneous processes

$$\begin{aligned} z_{t+1} &= (1-\rho_z)\mu_z + \rho_z z_t + \epsilon^z_{t+1} \ , \ \epsilon^z_{t+1} \sim \mathcal{N}(0,\sigma_z^2) \\ e_{i,t+1} &= (1-\rho_e)\mu_e + \rho_e e_{it} + \epsilon^e_{i,t+1} \ , \ \epsilon^e_{i,t+1} \sim \mathcal{N}(0,\sigma_e^2) \end{aligned}$$

as well as the aggregate law of motion for capital

$$k_{t+1} = \zeta_0 + \zeta_1 k_t + \zeta_2 z_t$$

The previous expressions can be written as in (33), so that

$$\begin{aligned} E_t\Big[f(y_{t+1}, y_t, y_{t-1}, \epsilon_t)\Big] &= E_t \begin{bmatrix} c_{it}^{-\gamma} - \beta E_t\Big[c_{i,t+1}^{-\gamma}\big(r(k_{t+1}, \bar{l}, z_{t+1}) + 1 - \delta\big) - p(a_{i,t+1})\Big] \\ a_{i,t+1} - (1-\delta)a_{it} - r(k_t, \bar{l}, z_t) - w(k_t, \bar{l}, z_t)\bar{l} + c_{it} \\ z_{t+1} - (1-\rho_z)\mu_z - \rho_z z_t - \epsilon^z_{t+1} \\ e_{i,t+1} - (1-\rho_e)\mu_e - \rho_e e_{it} - \epsilon^e_{i,t+1} \\ k_{t+1} - \zeta_0 - \zeta_1 k_t - \zeta_2 z_t \end{bmatrix} \\ &= 0 \end{aligned} \tag{34}$$

where $y = (c \ \ k \ \ a \ \ z \ \ e)'$.

The solution to this model is given by the following decision rules

$$c_{it} = g^c(a_{it}, e_{it}, z_t, k_t) \tag{35}$$

$$a_{it} = g^a(a_{it}, e_{it}, z_t, k_t) \tag{36}$$

and

$$k_t = g^k(a_{it}, e_{it}, z_t, k_t) \tag{37}$$

$$z_t = g^z(a_{it}, e_{it}, z_t, k_t) \tag{38}$$

$$e_{it} = g^e(a_{it}, e_{it}, z_t, k_t) \tag{39}$$

where σ is a scale parameter and determines the degree of uncertainty in ϵ. However, $g^z(\cdot)$ and $g^e(\cdot)$ are known and determined by an autoregressive process of order 1.

Since (34) is a system of nonlinear equations, (36) give a non-linear mapping from the current state variables to the optimal allocations for consumption and next period values for individual asset holdings.

Perturbation methods are local solution methods, hence the approximation to the functions $g(\cdot)$ are always taken in the neighborhood of the deterministic steady state of the model $(\bar{a}, \bar{e}, \bar{z}, \bar{k})$.

Since the solution algorithm described in the previous sections relies on the decision rule for the next period's individual asset holding, we will focus on the first order approximation of $g^a(\cdot)$ around the steady state $(\bar{a}; \bar{e}, \bar{z}, \bar{k})$

$$\begin{aligned} a_i'(a_i, e_i, z, k; \sigma) &= g^a(a_i, e_i, z, k; \sigma) \\ &= g^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) + (a_i - \bar{a}) g_a^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) \\ &\quad + (e_i - \bar{e}) g_e^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) + (z - \bar{z}) g_z^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) \\ &\quad + (k - \bar{k}) g_k^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) + \sigma g_\sigma^a(\bar{a}_i, \bar{e}_i, \bar{z}, \bar{k}; 0) \end{aligned} \tag{40}$$

The unknowns in this Taylor expansion are given by the set of first order derivatives

$$g_a^a,\ g_e^a, g_z^a, g_k^a, g_\sigma^a \tag{41}$$

These unknown coefficients are solved in Dynare.

6.2 Projection method

Let us remind the equilibrium conditions for this model:

$$\begin{aligned}
c_{it}^{-\gamma} &= \beta E_t\Big[c_{i,t+1}^{-\gamma}\big(r(k_{t+1},\bar{l},z_{t+1})+1-\delta\big)+p(a_{i,t+1})\Big] \\
a_{i,t+1} &= (1-\delta)a_{it}+r(k_t,\bar{l},z_t)+w(k_t,\bar{l},z_t)\bar{l}-c_{it} \\
z_{t+1} &= (1-\rho_z)\mu_z+\rho_z z_t+\epsilon_{t+1}^z\ ,\ \epsilon_{t+1}^z\sim\mathcal{N}(0,\sigma_z^2) \\
e_{i,t+1} &= (1-\rho_e)\mu_e+\rho_e e_{it}+\epsilon_{i,t+1}^e\ ,\ \epsilon_{i,t+1}^e\sim\mathcal{N}(0,\sigma_e^2) \\
k_{t+1} &= \zeta_0+\zeta_1 k_t+\zeta_2 z_t
\end{aligned} \tag{42}$$

To solve the model using projection methods, it is useful to first represent it by an operator equation, $\mathcal{R}(f)$. There is some freedom in choosing which function to approximate by parametric forms and which equilibrium conditions to use as residual functions.

In the present application, we need to parameterize the decision rule for next periods' individual asset holdings and we use as residual functions the Euler equation in (42). By using the barrier method, the decision rule for individual asset holdings is relatively smooth with a limited degree of nonlinearity, and hence it can be reasonably well approximated by polynomials. Then, for a given parameterization of the decision rule for individual asset holdings, the remaining choices of variables can be computed using the remaining expressions from (42). For instance, by plugging the solution for $a_i'(a_i, e_i, z, k)$ into the budget constraint we can deduce c_i, which will allow us to compute the Euler equation and hence the residual function $\mathcal{R}(f)$. Again, we can compute the level of aggregate capital by explicitly aggregating the decision rule for individual asset holdings. This procedure involves numerical optimization routines in order to minimize the square residual $\mathcal{R}(f)$ over a finite set of points in the bounded state space.

In the present version of the model, agents differ only in the realizations of the shocks that determine their employment status. As a consequence, the policy rules for each agent will be the same. The first step in the projection method is to define a bounded state space. Let $\mathcal{P}_n(x)$ denote a polynomial of degree n on the vector x. In the projection method we replace next periods asset holding, a_{it+1}, by a function of the state variables of the agent that is, $\mathcal{P}_n(a, e, z, k; \Theta_n)$. We will choose $\mathcal{P}_n(\cdot)$, and Θ_n, the vector of parameters, to make the marginal utility of consumption $c(a, e, z, k; \Theta_n)^{-\gamma}$ as close as possible to the conditional expectation. Note that $\mathcal{P}_n(a, e, z, k; \Theta_n)$ has been used to compute $c(a, e, z, k; \Theta_n)^{-\gamma}$. Since the objective of this paper is to compare two numerical approximation methods up to order $n = 2$, we can use ordinary polynomials.

After fixing n, we need to find Θ_n. The algorithm starts with an initial guess of Θ_n, Θ_n^0. Starting from Θ_n^0, we choose a new Θ_n in order to minimize the residual function

$$\begin{aligned}\mathcal{R}(\cdot, \Theta_n) \equiv\ & c(a, e, z, k; \Theta_n)^{-\gamma} \\ & -\beta E_t\left[c'(a', e', z', k'; \Theta_n)^{-\gamma}\left(r'(a', e', z', k'; \Theta_n) + 1 - \delta\right) + p(a')\right]\end{aligned} \tag{43}$$

The expectation is computed using a deterministic integration method, more specifically the Gauss-Hermite method Abramowitz and Stegun [1]. In general, the algorithm starts with a first-order polynomial and then adds higher-order terms until the results do not change anymore. However, we have decided to stop at order $n = 2$, as the current version of Dynare is able to do second order approximations only[6] .

6.3 Perturbation vs. Projection

First order perturbation methods have been widely used since the work of Blanchard and Kahn [5], and is described in King and Watson [12], Klein [13] and Sims [18]. The perturbation principle differs from the projection method in several respects. For instance, first order perturbation linearizes the system around a point of the state space, such as the steady state of the economy. Next, we take first order Taylor expansions around this steady state value, which requires calculations of the first derivatives of the functions h and f. First order perturbation methods are fast and widely used.

However, this approximation may be insufficient when analyzing the impact risk has on the behaviour of the economy. For this reason we may want to solve the model using second order approximations. This method may produce locally accurate approximations for capturing the dynamics of the model without ignoring the nonlinear features of it, as well as the higher order effects. In this case, we compute a second order Taylor expansion of the model. The method is described in Schmitt-Grohé and Uribe [17]. First and second order perturbation methods are implemented in the open-source software Dynare; Dynare++ is available for higher

[6]For higher-order perturbation methods, we could use Dynare++

order perturbation methods.

6.4 Complete polynomials

In both solution methods we use complete polynomials as base functions (Judd [10]). There are two reasons for this choice: first, the version of perturbation method that we use, and which is implemented in Dynare, uses $2nd$-degree Taylor series expansion around the deterministic steady state. Using a first-degree Taylor series ($k = 1$) for our four-dimensional problem uses the linear functions

$$\mathcal{P}_1^4 = \{1, a_i, e_i, z, M_a\}$$

where the subscript gives the degree of the Taylor series expansion and the superscript gives the dimension of the state space. For ($k = 2$), Taylors theorem uses

$$\mathcal{P}_2^6 = \mathcal{P}_1^4 \cup \left\{a_i^2, e_i^2, z^2, M_a^2, M_{a^2}^2, M_{ae}^2, a_i e_i, a_i z, a_i M_a, a_i M_{a^2}, a_i M_{ae}, e_i z, e_i M_a, z M_a\right\}$$

The second reason for preferring complete polynomials over, for instance, tensor product collections is that the dimension with the former one grows only polynomially as the dimension increases.

7 Accuracy Checks

The maximal and average Euler equation errors for both procedures are given in Table 6. These numbers are the logarithm values. They represent the percentage cost in terms of steady state consumption due to the approximation. A value of -4, for example, implies a mistake of \$1 for every \$10 000 spent.

It is clear from the numbers reported in these tables that the perturbation and projection methods produce roughly the same level of accuracy. A closer look at these methods, however, suggests that the perturbation method performs slightly better in the case of low idiosyncratic uncertainty, while the projection method performs better as the level of idiosyncratic risk increases.

The sensitivity of the results to the number of Chebyshev nodes, and for the Gauss-Hermite nodes used in the numerical integration, is examined by increasing the limit from 10 to 20, and 10 to 50, respectively. These changes had a negligible effect on the results, and in order to keep the computational time low, we chose 10 nodes for each method.

In order to check for the accuracy of the numerical approximation, we compute the Euler equation errors, as described in Judd [10]

$$E(a, e, z, k; \Theta_n) = \frac{R(a, e, z, k; \Theta_n)}{\hat{c}(a, e, z, k; \Theta_n)^{-\gamma}}$$

This term is a dimension-free quantity that expresses optimization error as a fraction of current consumption. In economic terms, it tells us how irrational agents would be in using the approximating rule. For instance, in our benchmark model the maximum value of the error is found to be 0.0046, which is to say that this approximation implies that agents make 0.46 percent errors in their period-to-period consumption decisions. This value is very low (Table 6), and we can conclude that the approximation is very good. This is not surprising, as the calibration of our benchmark model implies low uncertainty and quasi-linear policy functions.

Table 6 shows the log Euler equation errors, where $\log ||E||_\infty$ and $\log ||E||_1$ represent respectively the maximum error and the average error in the bounded state space $[a_{min}, a_{max}] \times [e_{min}, e_{max}] \times [z_{min}, z_{max}] \times [k_{min}, k_{max}]$. Finally, the Euler equation errors and the least squares projection method is computed in a 4-D state space with 10000 nodes.

[Table 2 about here.]

8 Parameterization

The parameterization of the benchmark model is standard (Table 7). To briefly recapitulate, the time period is one quarter; the intertemporal discount factor β is set equal to 0.99, and the depreciation rate δ to 0.1. We choose for the benchmark model a value for γ equal to 1. Most studies estimate the relative risk aversion parameter γ to be between one and two. For this reason, we include different values within this range in our sensitivity analysis. The share of capital α is 0.33. The normalizing constant $\bar{l}$ is set equal to 1. The aggregate technology shock is specified by $\mu^z = 1$, $\rho^z = 0.9$ and $\sigma^z = 0.01$ to correspond the two state Markov process close to the one adopted in Krusell and Smith [14]. As mentioned earlier, the law of motion for individual's employment status is transformed to continuous support, as shown in equation (2). Labor market conditions therefore depend on the aggregate state. The individual's employment status is specified as $\mu^e = 1$, $\rho^e = 0.95$ and $\sigma^e = 0.005$.

The analysis assumes that agents are constrained to hold positive quantities of assets, so that the borrowing limit is set to $b = 0$. In other words, the credit limit is set at 0, which means that agents are not allowed to hold net debt. The parameter ϕ controls the sensitivity to the borrowing constraint in the modified utility function and it is set to $\phi = 0.05$. This ensures that no agent violates the borrowing constraint.

[Table 3 about here.]

9 Results

The selected quantitative results will focus on the dynamics of aggregate capital for studying model properties. Our comparison exercise has been done in various dimensions.

Simulation vs. explicit aggregation We first used linear approximation methods to solve the model. For this, we used four alternative algorithms obtained from the combination perturbation/projection and simulation-based/explicit aggregation. The number of iterations needed to reach convergence is the same for both the simulation-based algorithm or the explicit aggregation algorithm. The convergence path for the coefficients of the aggregate law of motion is given in Figure 2. However, each single iteration can be very slow in the simulation-based algorithm. This is obviously due to the time-consuming simulation of the economy composed of N agents and T time periods. We found that the values for T and N need to be relatively large in order to get accurate approximations of the stationary cross-sectional distribution. Another finding is that as we increase the number of agents and the length of the time period, the coefficients of the aggregate law of motion from the simulation-based method converges to those obtained from the explicit aggregation algorithm[7]. The results are presented in the first two columns in Table 5.

[7] In practice we find that the algorithm which uses explicit aggregation is much faster and requires a very small amount of RAM.

Global vs. local approximation When comparing the projection method to the perturbation method (Table 1), the results diverge as soon as we make the households more risk-averse and include additional idiosyncratic uncertainty. This is not surprising, given that the projection method approximates the model over a larger portion of the state space, while the perturbation methods use local approximations around a single point, which is the deterministic steady state. In this case, a high value for the risk aversion parameter and a larger deviation from the steady state may be ill-suited for the perturbation method.

The role of second order effects In this economy, there are two important reasons for going beyond a first order approximation of the decision rules. First, the presence of a borrowing constraint will add significant nonlinearities to the decision rules, and simple linear approximation of the model may in that case distort the dynamics of the economy. The second reason is related to the presence of idiosyncratic shocks and incomplete markets. In this economy, households cannot borrow and the only way to insure against a strain of bad employment status shocks is to accumulate assets. By holding a higher amount of capital than in the complete market setup, households will be able to smooth out consumption in case they are hit by bad idiosyncratic shocks. This means that households have to take uncertainty into account when shaping their decisions. Since first order approximations are subject to the certainty equivalence principle, uncertainty will not play any role. In order to include uncertainty into the households' decision rules, we have to use at least second order approximations of the model. In order to capture nonlinearities and to study the impact of uncertainty on aggregate wealth, we solve the model using second order perturbation methods and compute the aggregate capital of the economy. In Table 8 we compare the results from a first order solution to those of a second order solution. Clearly by taking into account second order effects (*column* 4 and 8), the aggregate level of asset holdings will increase, which is not the case for first order approximations where uncertainty is ignored (*column* 2 and 6).

[Table 4 about here.]

Conclusion

This paper presents a simple and fast algorithm that allows us to solve heterogeneous agent models with aggregate uncertainty and incomplete markets. We present and justify the modifications that have to be made to the original model in order to be able to use our algorithm.

Next, we compared various versions of the approximations in order to test the robustness and accuracy of our algorithm. Compared to global solutions, we find that local solution methods perform relatively well, as long as the idiosyncratic shocks that hit the households are not too large. Secondly, comparing first and second order approximations allows us to highlight the significance of second order effects and precautionary savings motives in this type of model. Finally, comparing simulation-based and explicit-aggregation algorithms highlights the speed and accuracy of the explicit aggregation algorithm.

As mentioned earlier, the basic properties (i.e., precautionary savings) of the original model are preserved in our approximation. One key component of the method described here is the specification of the borrowing constraint. The sensitivity of the solution to the choice of this specification is an important issue that will be the focus of the next chapter.

References

[1] Abramowitz, M. and I.A. Stegun *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.*, Dover (New York), 1964.

[2] Aiyagari, S.R. Uninsured Idiosyncratic Risk and Aggregate Saving *Quarterly Journal of Economics*, 109:659–684, 1994.

[3] Algan, Y., O. Allais and W.J. den Haan Solving heterogeneous-agent models with parameterized cross-sectional distributions *Journal of Economic Dynamics & Control*, 32:875–908, 2008.

[4] Bewley, T. Interest Bearing Money and the Equilibrium Stock of Capital *manuscript*, undated.

[5] Blanchard, O.J. and C.M. Kahn The Solution of Linear Difference Models under Rational Expectations *Econometrica* , 48:1305–1311, 1980.

[6] den Haan, W.J. Solving Dynamic Models With Aggregate Shocks And Heterogeneous Agents *Macroeconomic Dynamics* , 1:355–386, 1997.

[7] den Haan, W.J. and P. Rendahl Solving the Incomplete Markets Model with Aggregate Uncertainty using Explicit Aggregation *Journal of Economic Dynamics & Control*, 34(1):69–78, 2009.

[8] den Haan, W.J. and J. de Wind How well-behaved are higher-order perturbation solutions? *DNB Working Paper*, no. 240, 2010.

[9] Huggett, M. The Risk-Free Rate in Heterogeneous-Agents, Incomplete Markets Economies *Journal of Economic Dynamics & Control* , 17:953–969, 1993.

[10] Judd, K.L. *Numerical Methods in Economics* , The MIT Press (Cambridge, Massachusetts), 1993.

[11] Kim, S.H., R. Kollmann and J. Kim Solving the incomplete market model with aggregate uncertainty using a perturbation method *Journal of Economic Dynamics & Control*, 34(1):50–58, 2009.

[12] King, R.G. and M.W. Watson The Solution of Singular Linear Difference Systems under Rational Expectations *International Economic Review* , 39(4):1015–1026, 1998.

[13] Klein, P. Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model *Journal of Economic Dynamics & Control* , 24(10):1405–1423, 2000.

[14] Krusell, P. and A.A. Smith Income and wealth heterogeneity in the macroeconomy *Journal of Political Economy*, 106(5):867–896, 1998.

[15] Krusell, P. and A.A. Smith Quantitative Macroeconomic Models with Heterogeneous Agents *Econometric Society Monographs*, 41:298-340, 2006.

[16] Preston, B. and M. Roca Incomplete Markets, Heterogeneity and Macroeconomic Dynamics *NBER Working Paper Series*, 13260, 2007.

[17] Schmitt-Grohé, S. and M. Uribe Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function *Journal of Economic Dynamics & Control* , 28(4):755–775, 2004.

[18] Sims, C.A. Solving Linear Rational Expectations Models *Computational Economics* , 20(1-2):1–20, 2002.

Dynamic Accuracy Test

9.0.1 Accuracy test for the individual policy rules

This test consists of two parts, one dealing with the individual policy rules

1. Simulate the aggregate variables $\{k_t\}_{t=1}^T$ and $\{z_t\}_{t=1}^T$

2. Simulate the employment status $\{e_t\}_{t=1}^T$ for one agent

3. Simulate the individual asset holding $\{a_t\}_{t=1}^T$ for one agent using the policy function $\mathcal{P}(a_t, e_t, z_t, k_t)$. This part does not involve the numerical approximation of the integral associated with the expectation. Here we only use the policy function to simulate the individual time series

4. Simulate an alternative individual asset holding $\{\tilde{a}_t\}_{t=1}^T$ based on the same $\{z_t\}_{t=1}^T$, $\{k_t\}_{t=1}^T$, $\{e_t\}_{t=1}^T$ and e_0

   (a) Start by setting, $\tilde{a}_0 = a_0$

   (b) Compute $\tilde{a}_{temp} = \mathcal{P}(\tilde{a}_t, e_t, z_t, k_t)$. This $\tilde{a}_{temp}$ will only be used to compute the expectation. Since we have already simulated $\{k_t\}_{t=1}^T$ and $\{z_t\}_{t=1}^T$, we can compute $\{r_t\}_{t=1}^T$ and $\{w_t\}_{t=1}^T$. Using the budget constraint, we get

$$
\begin{aligned}
c' &= r'\tilde{a}_{temp} + w'e' + (1-\delta)\tilde{a}_{temp} - \mathcal{P}(\tilde{a}_{temp}, e', z', k') \\
c_t^{-\gamma} &= \beta E_t \left[c'^{-\gamma}(r' + 1 - \delta) + \mathbf{P}(\tilde{a}_{temp}) \right] \\
c_t &= \left(\beta \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \left[c'^{-\gamma}(r' + 1 - \delta) + \mathbf{P}(\tilde{a}_{temp}) \right] \frac{\omega_{1,j_1}}{\sqrt{\pi}} \frac{\omega_{1,j_1}}{\sqrt{\pi}} \right)^{-\frac{1}{\gamma}}
\end{aligned}
$$

Using the budget constraint we compute $\tilde{a}_{t=1}^{T}$ and throw away $\tilde{a}_{temp}$

$$\tilde{a}_{t+1} = r_t \tilde{a}_t + w_t e_t + (1-\delta)\tilde{a}_t - c_t$$

5. Once we have simulated $\{a_t\}_{t=1}^{T}$ and $\{\tilde{a}_t\}_{t=1}^{T}$, we plot and compare the two series. This will give us an idea on the regions (that do matter) where the accuracy is low

9.0.2 Accuracy test for the aggregate policy rules

This test assumes that the individual policy rule is accurate and we are going to test the accuracy of the aggregate law of motion.

1. Simulate $\{k_t\}_{t=1}^{T}$ and $\{z_t\}_{t=1}^{T}$ using the aggregate law of motion and the exogenous law of motion for z

2. At each period, we simulate a large number of individuals and we use the individual asset holdings for computing

$$\tilde{k}_t = \sum_i a_i$$

3. For next period, k_{t+1}, we use the policy function

$$a_{t+1} = \mathcal{P}(a_t, e_t, z_t, k_t)$$

and

$$\tilde{k}_{t+1} = \sum_i a_{t+1}$$

4. Compare $\{k_t\}_{t=1}^{T}$ and $\{\tilde{k}_t\}_{t=1}^{T}$

Matlab and Dynare Code

The **Matlab** code:

```
for idxIter = 1:iIter

 % Solve the model for given coeffs. of the aggregate law of motion (vZetaOld)
 dynare Dimension4PF noclearall ;

 vZetaNew(1) = vTheta(1) + vTheta(3);
 vZetaNew(2) = vTheta(2) + vTheta(5);
 vZetaNew(3) = vTheta(4);

 % Check convergence of coefficients
 dConv = fnConvergence(vZetaNew,vZetaOld,iTol);
 if dConv == 1
     break;
 end

 vZetaOld = dLambda * vZetaNew + (1-dLambda) * vZetaOld;

 pZeta0 = vZetaOld(1);
 pZeta1 = vZetaOld(2);
 pZeta2 = vZetaOld(3);
 delete InitParams.mat;
 save InitParams.mat pZeta0 pZeta1 pZeta2;

end
```

The **Dynare** mod-file:

```
var ... ; // declare endogeneous variables
varexo ...; // declare exogeneous variables
parameters  ... ; // declare parameters

load InitParams; // load coefficients for ALM
set_param_value('pZeta0',pZeta0);
set_param_value('pZeta1',pZeta1);
set_param_value('pZeta2',pZeta2);

load StructParams; // load structural parameters (sensitivity analysis)
set_param_value('pGamma',pGamma);
set_param_value('pSigmae',pSigmae);

model; ... end;  // model block

initval; ... end; // initial values for solver

shocks; ... end; // declare shocks
```

```
stoch_simul(order=1,nocorr,noprint,nomoments,IRF=0);

// Collecting parameters

mPolicy = [oo_.dr.ys'; oo_.dr.ghx'; oo_.dr.ghu']; // read coefficients of policy
  // functions
mPolA = mPolicy(:,2);
// Rearrange parameters
dTheta0 = mPolA(1)-mPolA(2)*mPolA(1)-mPolA(6)-mPolA(7)-mPolA(5)*mPolicy(1,5);
dTheta1 = mPolA(2);
dTheta2 = mPolA(6);
dTheta3 = mPolA(7);
dTheta4 = mPolA(5);

vTheta = [dTheta0 dTheta1 dTheta2 dTheta3 dTheta4];
```

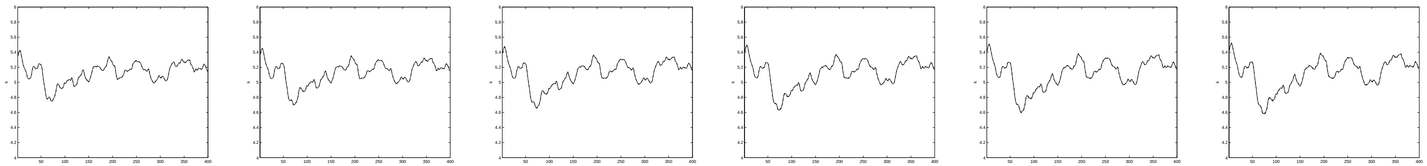
Table 1: Approximating Functions: Family of Monomials (2nd order projection method)

	cst	a	e	z	M_a	M_{ae}	M_{a^2}
cst	cst	a	e	z	M_a	M_{ae}	M_{a^2}
a	a	a^2	ae	az	aM_a	aM_{ae}	aM_{a^2}
e	e	ea	e^2	ez	eM_a	eM_{ae}	eM_{a^2}
z	z	za	ze	z^2	zM_a	zM_{ae}	zM_{a^2}
M_a	M_a	$M_a a$	$M_a e$	$M_a z$	M_a^2	$M_a M_{ae}$	$M_a M_{a^2}$
M_{ae}	M_{ae}	$M_{ae} a$	$M_{ae} e$	$M_{ae} z$	$M_{ae} M_a$	$M_{ae} M_{ae}$	$M_{ae} M_{a^2}$
M_{a^2}	M_{a^2}	$M_{a^2} a$	$M_{a^2} e$	$M_{a^2} z$	$M_{a^2} M_a$	$M_{a^2} M_{ae}$	$M_{a^2}^2$

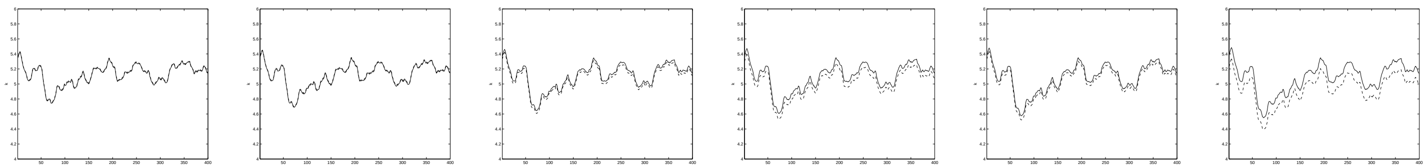
Table 2: Approximating Functions: Family of Monomials (2nd order perturbation method)

	cst	a	e_{-1}	z_{-1}	M_a	M_{ae}	M_{a^2}	ε^e	ε^z
cst	cst	a	e_{-1}	z_{-1}	M_a	M_{ae}	M_{a^2}	ε^e	ε^z
a	a	a^2	ae_{-1}	az_{-1}	aM_a	aM_{ae}	aM_{a^2}	$a\varepsilon^e$	$a\varepsilon^z$
e_{-1}	e_{-1}	$e_{-1}a$	e_{-1}^2	$e_{-1}z_{-1}$	$e_{-1}k$	$e_{-1}M_{ae}$	$e_{-1}M_{a^2}$	$e_{-1}\varepsilon^e$	$e_{-1}\varepsilon^z$
z_{-1}	z_{-1}	$z_{-1}a$	$z_{-1}e_{-1}$	z_{-1}^2	$z_{-1}M_a$	$z_{-1}M_{ae}$	$z_{-1}M_{a^2}$	$z_{-1}\varepsilon^e$	$z_{-1}\varepsilon^z$
M_a	M_a	$M_a a$	$M_a e_{-1}$	$M_a z_{-1}$	M_a^2	$M_a M_{ae}$	$M_a M_{a^2}$	$M_a\varepsilon^e$	$M_a\varepsilon^z$
M_{ae}	M_{ae}	$M_{ae}a$	$M_{ae}e_{-1}$	$M_{ae}z_{-1}$	$M_{ae}M_a$	$M_{ae}M_{ae}$	$M_{ae}M_{a^2}$	$M_{ae}\varepsilon^e$	$M_{ae}\varepsilon^z$
M_{a^2}	M_{a^2}	$M_{a^2}a$	$M_{a^2}e_{-1}$	$M_{a^2}z_{-1}$	$M_{a^2}M_a$	$M_{a^2}M_{ae}$	$M_{a^2}^2$	$M_{a^2}\varepsilon^e$	$M_{a^2}\varepsilon^z$
ε^e	ε^e	$\varepsilon^e a$	$\varepsilon^e e_{-1}$	$\varepsilon^e z_{-1}$	$\varepsilon^e M_a$	$\varepsilon^e M_{ae}$	$\varepsilon^e M_{a^2}$	$\varepsilon^e\varepsilon^e$	$\varepsilon^e\varepsilon^z$
ε^z	ε^z	$\varepsilon^z a$	$\varepsilon^z e_{-1}$	$\varepsilon^z z_{-1}$	$\varepsilon^z M_a$	$\varepsilon^z M_{ae}$	$\varepsilon^z M_{a^2}$	$\varepsilon^z\varepsilon^e$	$\varepsilon^z\varepsilon^z$

Figure 1: Aggregate law of motion for capital, M_a, approximated with **perturbation method** (first row) and **projection method** (second row) with $\sigma_e = 0.005$. First order approximations are represented through continuous line, while second order approximations are plotted in dashed line.

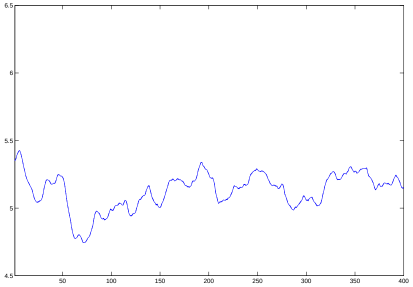


(a) $\gamma = 1.00$ (b) $\gamma = 1.50$ (c) $\gamma = 2.00$ (d) $\gamma = 2.50$ (e) $\gamma = 3.00$ (f) $\gamma = 3.50$

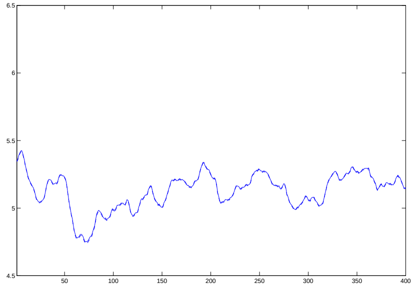


(g) $\gamma = 1.00$ (h) $\gamma = 1.50$ (i) $\gamma = 2.00$ (j) $\gamma = 2.50$ (k) $\gamma = 3.00$ (l) $\gamma = 3.50$

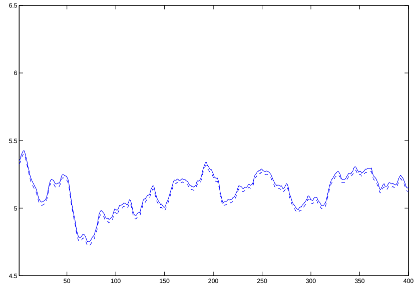
Table 3: Aggregate law of motion for capital, M_a, approximated with **first order perturbation method** (*continuous line*) and **second order perturbation method** (*dashed line*)



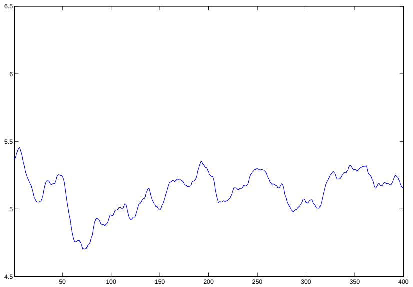
(a) $\gamma = 1.0$, $\sigma_e = 0.005$



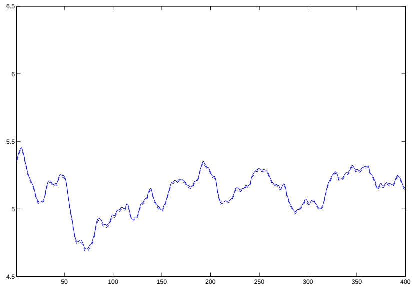
(b) $\gamma = 1.0$, $\sigma_e = 0.05$



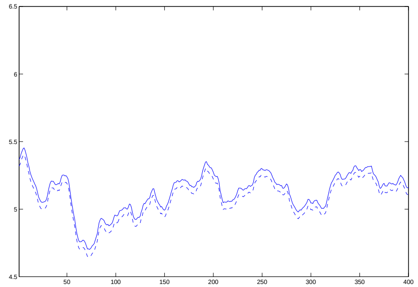
(c) $\gamma = 1.0$, $\sigma_e = 0.1$



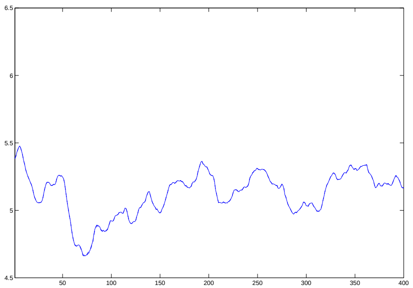
(d) $\gamma = 1.5$, $\sigma_e = 0.005$



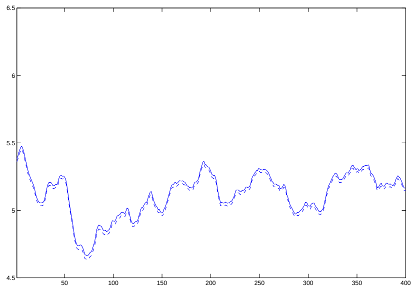
(e) $\gamma = 1.5$, $\sigma_e = 0.05$



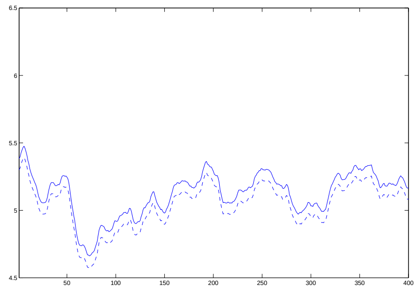
(f) $\gamma = 1.5$, $\sigma_e = 0.1$



(g) $\gamma = 2.0$, $\sigma_e = 0.005$



(h) $\gamma = 2.0$, $\sigma_e = 0.05$



(i) $\gamma = 2.0$, $\sigma_e = 0.1$

Table 4: Results from First Order Approximation for $\gamma = 1.0$, $\gamma = 1.5$, and $\gamma = 2.0$

		Method for ALM		Individual Policy Function					Coefficients for the ALM		
				θ_0	θ_1	θ_2	θ_3	θ_4	ζ_0	ζ_1	ζ_2
γ	σ_e			*const*	a	e	z	k	*const*	k	z
1.0	0.005	Xpa	Perturbation	-0.67961	0.91636	0.37787	1.03877	-0.05953	-0.30175	0.85683	1.03878
			Projection	-0.69772	0.91869	0.38446	1.04509	-0.06091	-0.31062	0.85763	1.04315
	0.05		Perturbation	-0.67961	0.91636	0.37787	1.03877	-0.05953	-0.30175	0.85683	1.03878
			Projection	-0.69612	0.91897	0.37788	1.05298	-0.06040	-0.31850	0.85868	1.05269
	0.1		Perturbation	-0.67961	0.91636	0.37787	1.03877	-0.05953	-0.30175	0.85683	1.03878
			Projection	-0.68900	0.92054	0.34559	1.08491	-0.05836	-0.34479	0.86245	1.08491
1.5	0.005	Xpa	Perturbation	-0.81449	0.92867	0.41797	1.01229	-0.04829	-0.39652	0.88038	1.01230
			Projection	-0.81655	0.93019	0.42395	1.00935	-0.05019	-0.40450	0.88102	1.01602
	0.05		Perturbation	-0.81449	0.92867	0.41797	1.01229	-0.04829	-0.39652	0.88038	1.01230
			Projection	-0.84207	0.93202	0.41792	1.03330	-0.04865	-0.41798	0.88267	1.03070
	0.1		Perturbation	-0.81449	0.92867	0.41797	1.01229	-0.04829	-0.39652	0.88038	1.01230
			Projection	-0.82472	0.93437	0.37466	1.07504	-0.04675	-0.46713	0.88910	1.08486
2.0	0.005	Xpa	Perturbation	-0.90545	0.93550	0.44355	1.00573	-0.04114	-0.46190	0.89436	1.00574
			Projection	-0.90958	0.93672	0.44777	1.00413	-0.04236	-0.46848	0.89480	1.00850
	0.05		Perturbation	-0.90545	0.93550	0.44355	1.00573	-0.04114	-0.46190	0.89436	1.00574
			Projection	-0.93137	0.93873	0.44007	1.03040	-0.04133	-0.48934	0.89719	1.02956
	0.1		Perturbation	-0.90545	0.93550	0.44355	1.00573	-0.04114	-0.46190	0.89436	1.00574
			Projection	-0.94895	0.94498	0.39268	1.10484	-0.03896	-0.57097	0.90732	1.11321

Figure 2: Convergence of the ALM coefficients during the updating process ($\gamma = 1.00$).

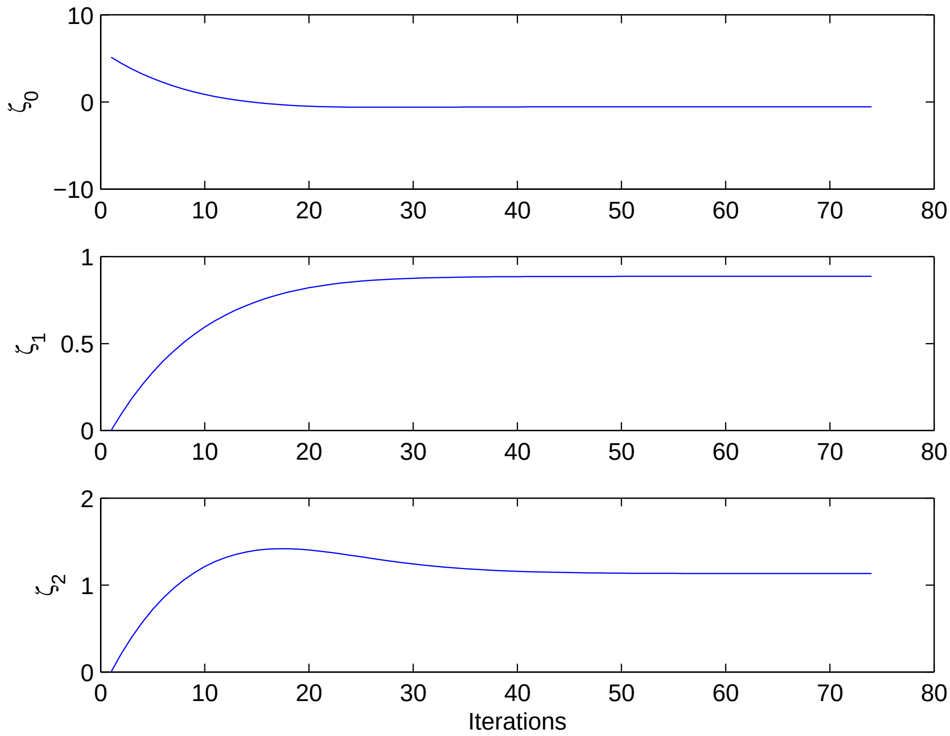


Figure 3: Approximation Aggregation - Second Order Approximation

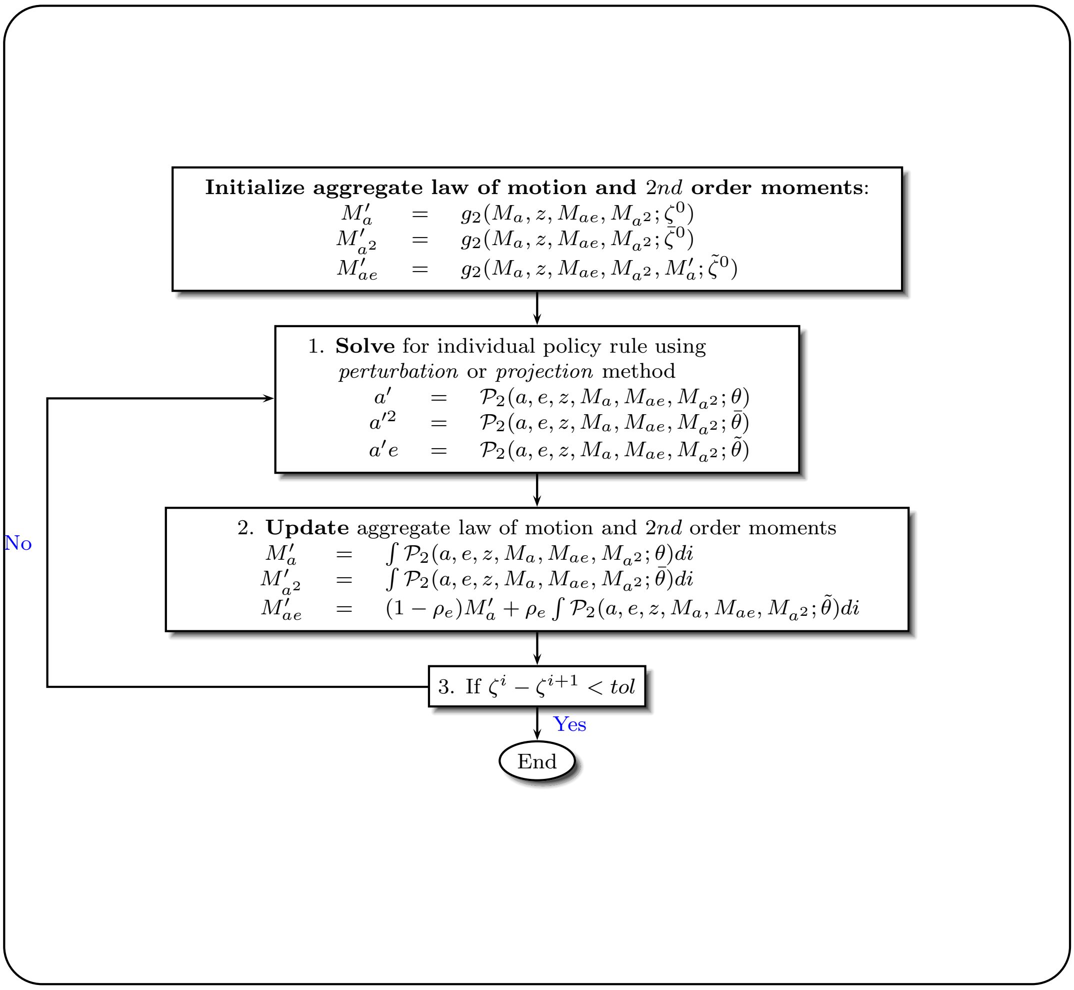


Table 5: Coefficients of the aggregate law of motion after convergence

	Simulation		Xpa	
	Perturbation	Projection	Perturbation	Projection
ζ_0	-0.30165	-0.30813	-0.30175	-0.31326
ζ_1	0.85681	0.85735	0.85683	0.85777
ζ_2	1.03873	1.04214	1.03878	1.04510
Iteration	-	-	-	-
Dampening λ	0.5	0.5	0.5	0.5
T periods	21000	21000	-	-
Burn-in	10000	10000	-	-
N agents	10000	10000	-	-
tol	10^{-5}	10^{-5}	10^{-5}	10^{-5}

Table 6: log Euler equation errors

		First Order Approximation			
		Projection		Perturbation	
γ	σ_e	$\|\|E\|\|_\infty$	$\|\|E\|\|_1$	$\|\|E\|\|_\infty$	$\|\|E\|\|_1$
1.0	0.005	-5.2119	-7.1647	-5.2734	-7.5850
	0.050	-5.1441	-7.1013	-4.9779	-6.6844
	0.100	-4.8109	-6.0265	-4.4006	-5.1563
1.5	0.005	-5.2679	-7.4768	-5.3530	-7.6463
	0.050	-5.3538	-7.1730	-4.9949	-6.5249
	0.100	-4.7907	-5.7823	-4.2882	-4.9477
2.0	0.005	-5.2555	-7.5171	-5.3215	-7.6430
	0.050	-5.1424	-7.0488	-4.9148	-6.3372
	0.100	-4.5529	-5.6149	-4.0674	-4.7447

Table 7: Parameterization of the Benchmark Economy

Parameter	α	β	δ	γ	μ^e	ρ^e	σ^e	μ^z	ρ^z	σ^z	$\bar{l}$	η_0	η_1	η_2
	.33	.99	.1	1	1	.95	.005	1	.9	.01	1	.1	.1	0

Table 8: Sample mean and standard deviation for aggregate capital ($T = 400$)

Method	Perturbation				Projection			
Order	1		2		1		2	
γ	Mean	Std.dev.	Mean	Std.dev.	Mean	Std.dev.	Mean	Std.dev.
1.00	5.1283	0.1330	5.1284	0.1337	5.1244	0.1340	5.1339	0.1333
1.50	5.1255	0.1496	5.1256	0.1506	5.1154	0.1505	5.1260	0.1501
2.00	5.1231	0.1638	5.1236	0.1652	5.1062	0.1645	5.0732	0.1643
2.50	5.1209	0.1759	5.1218	0.1779	5.0973	0.1762	5.0271	0.1762
3.00	5.1190	0.1862	5.1204	0.1888	5.0889	0.1859	5.0455	0.1877
3.50	5.1174	0.1950	5.1191	0.1983	5.0811	0.1938	4.9366	0.1937