Applying the Explicit Aggregation Algorithm to the Heterogeneous Agent Models with Aggregate Uncertainty in Continuous Time*

Masakazu Emoto[†] Takeki Sunakawa[‡]

September 2020

Abstract

This paper applies the explicit aggregation (XPA) algorithm proposed by den Haan and Rendahl (2010) to the standard heterogeneous agent model with aggregate uncertainty (Krusell and Smith, 1998) in continuous time. We find that the XPA algorithm is faster to solve the model than the Krusell-Smith (KS) algorithm as the XPA algorithm does not rely on simulations to solve the model. The XPA algorithm is more accurate than the perturbation method proposed by Reiter (2009) and Ahn et al. (2018) when aggregate uncertainty is large. Moreover, unlike Ahn et al. (2018), our algorithm can deal with nonlinear effects of aggregate uncertainty.

Keywords: Continuous Time, Heterogeneous Agent Models, Explicit Aggregation Algorithm.

JEL codes: C63; D52

*We thank Takashi Kamihigashi and Tamotsu Nakamura for comments and suggestions.

[†]Graduate School of Economics, Kobe University, 2-1 Rokko-dai, Nada, Kobe, 657-8501; Email: masakazu.emoto@gmail.com

[‡]Graduate School of Economics, Hitotsubashi University; Email: takeki.sunakawa@gmail.com.

1 Introduction

There are huge interests in heterogeneous-agent macro models ever than before. Recent studies such as Achdou et al. (2017) and Ahn et al. (2018) apply newly developed numerical methods to solve heterogeneous agent models in continuous time. In particular, Ahn et al. (2018) studies heterogeneous agent models with aggregate uncertainty in continuous-time and is able to solve the model very fast. However, there are some challenges to their approach. One, their method solves the model with a linear approximation and thus does not capture nonlinear effects of aggregate uncertainty. Two, due to the linearization, the accuracy of solving the model is significantly compromised when aggregate uncertainty is large.

We present an alternative numerical method to address the issues mentioned above. We introduce the explicit aggregation (XPA) algorithm (den Haan and Rendahl, 2010) into the standard heterogeneous agent model with aggregate shock (Krusell and Smith, 1998) in continuous time. Then we compare our algorithm in terms of the accuracy and efficiency with some of the existing algorithms, the Krusell-Smith (KS) algorithm using simulations and the Reiter-Ahn (REITER) algorithm using perturbation around the deterministic steady state. The latter is originally proposed by Reiter (2009) and recently adapted to continuous-time models Ahn et al. (2018).

We find that, compared with the KS algorithm, our algorithm is faster than and as accurate as the KS algorithm to solve the standard Krusell-Smith model. Compared with the REITER algorithm, our algorithm can solve the model as nearly fast as the REITER algorithm does, and is more accurate than the REITER algorithm, especially when aggregate uncertainty is large.

Our study is closely related to at least two areas of research. One is the literature on the XPA algorithm. den Haan and Rendahl (2010) is the first paper to apply this method to the standard heterogeneous agent model in Krusell and Smith (1998). Sunakawa (2020) applies their approach to some other heterogeneous agent models such as Khan and Thomas (2003, 2008) and Krueger et al. (2016) in discrete time. The present paper is the first to apply the XPA algorithm to the standard heterogeneous agent model with aggregate uncertainty in continuous time to the best of our knowledge.

The other is the research on the methods to solve heterogeneous agent models with aggregate shock in continuous time. The pioneering research in this area is Ahn et al. (2018). They adapt the perturbation method originally developed by Reiter (2009) to heterogeneous agent models with aggregate shock in continuous time.[1] Fernández-Villaverde et al. (2019) propose a neural-network algorithm to solve heterogeneous agent models with aggregate shock in continuous time. Our algorithm, unlike Fernández-Villaverde et al. (2019) and the standard KS algorithm, solves the model without using simulations. Furthermore, whereas the REITER algorithm solves the model with linear approximation, our algorithm solves the model nonlinearly so as to caputure nonlinear effects of aggregate uncertainty.[2] Also, as we will discuss later, the accuracy of solving the model is much higher than what is reported in Ahn et al. (2018), especially when aggregate uncertainty is large. Our results also hold for different degrees of the persistence of the aggregate shock.

The paper consists of the following sections. In Section 2, we apply the XPA algorithm, as well as the KS and REITER algorithms, to the Krusell and Smith (1998) model in continuous time. In Section 3, we compare the results of the three algorithms, XPA, KS, and REITER, in terms of accuracy and efficiency. Finally, Section 4 concludes.

2 Algorithms

We apply the XPA algorithm to Krusell and Smith (1998) model in continuous time studied by Ahn et al. (2018). We choose the Krusell-Smith model as it is known as one of the most popular heterogeneous agent models with aggregate uncertainty. Applying the XPA algorithm to other models is also straightforward. In the model, there are a representative firm, and a government, and heterogeneous households whose asset holdings and productivity are different from each other. As the model is well known, we defer the details of the model to Appendix A.

The XPA algorithm assumes the *approximate aggregation* (Young, 2005) so that the wealth

[1] Reiter (2010a); Ahn et al. (2018) further develop a method to reduce the dimension of the state space by projecting the distribution onto principal components. Bayer and Luetticke (2020) and Childers (2018) also suggest novel approaches using linearization.

[2] Reiter's (2010b) backward induction method can also be applied to solve heterogeneous agent models nonlinearly. The method is applied to stochastic overlapping generations models with aggregate uncertainty by Khan (2017); Kim (2018). Okahata (2018) demonstrates the method can also merge with the continuous–time methods in Ahn et al. (2018).

distribution is approximated by the mean as in the KS algorithm. In contrast with the KS algorithm, the XPA algorithm calculates the forecasting rules without simulations.

Both of the XPA and KS algorithms require two types of calculations, *the inner loop and the outer loop*. The inner loop calculation is common between the XPA and KS algorithms.[3] In continuous time models, the finite difference method is used to solve the Hamilton-Jacobi-Bellman (HJB) equation as in Achdou et al. (2017). Given the forecasting rule $\dot{K} = \Gamma(K, Z)$, we solve the HJB equation

$$\begin{aligned}\rho v(a, z, K, Z) = \max_c u(c) &+ v_a(a, z, K, Z)\dot{a} \\ &+ \lambda_z(v(a, z', K, Z) - v(a, z, K, Z)) + v_K(a, z, K, Z)\dot{K} \\ &+ v_z(a, z, K, Z)(-\mu Z) + \frac{\sigma^2}{2} v_{zz}(a, z, K, Z)\end{aligned}$$

for the policy function of household saving, $s(a, z, K, Z)$.

It is in the outer loop that the XPA and KS algorithms are different. In the outer loop, the policy function $s(a, z, K, Z)$ obtained in the inner loop is used to obtain the next period's aggregate capital as

$$\dot{K} = \sum_z \int_a s(a, z; K, Z) g(a, z) da.$$

In the XPA algorithm, we can obtain the forecasting rule by

$$\dot{K}(K, Z) = \sum_z \{s(K(z), z, K, Z) + \xi(z)\}\phi(z),$$

where $K(z)$ is capital conditioned on labor productivity z, $\phi(z) = \int g(a, z)da$ is the proportion of households with z, and $\xi(z)$ is for correcting the biases due to Jensen's inequality. These objects can be calculated when we compute the steady state in a fraction of time. That is, to obtain the forecasting rule, we just need to evaluate the policy function at $a = K(z)$. The value of $K(z)$ may not be on the grid of a, so we use linear interpolation. In Appendix B, we explain the details of

[3] Fernández-Villaverde et al. (2019) describes the KS algorithm in continuous time in details. In particular, they solve the standard Krusell-Smith model in continuous time using the KS algorithm.

the XPA algorithm following den Haan and Rendahl (2010) and Sunakawa (2020).

In the KS algorithm, we simulate the model to obtain the sequence of the TFP and the mean of the wealth distribution. Then the forecasting rule is obtained by estimating the following forecasting rule from the simulated sequence of $\{K_t, Z_t\}$.

$$\dot{K}(K, Z) = \beta_0 + \beta_1 \ln K + \beta_2 \ln Z.$$

The REITER algorithm differs from the XPA and KS algorithms in that the approximate aggregation does not hold. Instead, the REITER algorithm linearizes the model around the deterministic steady-state and uses the system of linearized equations to solve for the dynamics of the economy in the event of aggregate shocks. See Ahn et al. (2018) for more details.

3 Numerical results

We compare numerical results by three algorithms, the XPA algorithm, the KS algorithm, and the REITER algorithm. First, we illustrate the property of the forecasting rules obtained by XPA and KS. Then, we demonstrate the simulation results for XPA, KS, and REITER. After that, we discuss the computation time for each algorithm and the accuracy using the Denhaan Error proposed by den Haan (2010).[4]

Benchmark parameters

We use the parameters as in Table 1 for the benchmark case, following the ones used in Ahn et al. (2018). Later, we change the volatility and persistence of the TFP to see its effect on the accuracy of solving the model between different algorithms.

[4]We use the codes used in Fernández-Villaverde et al. (2019) for KS and the codes used in Ahn et al. (2018) for REITER. Both of the codes are modified to use the same set of parameters just for the purpose of comparison.

Table 1: Parameter Values

Parameters	Benchmark Value
γ : Relative Risk Aversion	1.0
ρ : Rate of Time Preference	0.01
α : Capital Share	0.36
δ : Rate of Capital Depreciation	0.025
τ : Tax Rate of Labor Income	0.011
b : Rate of Compensation	0.15
μ : Persistence of TFP	0.25
σ : Volatility of TFP	0.07
z : Idiosyncratic of Labor Productivity	$z_u = 0, z_e = 1$
λ : Probability of Labor Productivity	$\lambda_e = 0.50, \lambda_u = 0.03$

3.1 Forecasting rules

In Figure 1, we show the forecasting rule $\dot{K}(K, Z)$ in KS and XPA. As it is clear from the figure, there is no significant difference in the forecasting rules obtained by each algorithm. Each forecasting rule is characterized by a decreasing function with respect to the capital K and an increasing function with respect to the TFP Z. Also, the slope of the forecasting rule with regard to K is flatter than the 45-degree line as is in the standard neoclassical growth model. Thus, with respect to capital, if there is more (less) capital in the current period than in the steady state, $\dot{K}$ is negative (positive) and households will expect the future capital stock to decline (increase). Also, if the TFP is high (low), $\dot{K}$ is positive (negative) and households expect their next capital stock to increase (decrease).

3.2 Simulation paths

Next, we compare the capital paths obtained from the simulation results in KS, XPA, and REITER. In Figure 2, we show the results of the simulation path derived from the full model (i.e., the forecasting rule and household HJB equation) for 10000 periods in KS, XPA, and REITER. The blue line in the figure shows the capital path by XPA, the red line shows the capital path by KS, and the black line indicates the path by REITER. It is clear from the figure that the path of capital obtained by each algorithm is very close to each other, except for REITER when capital is low.

Figure 1: Forecasting rules

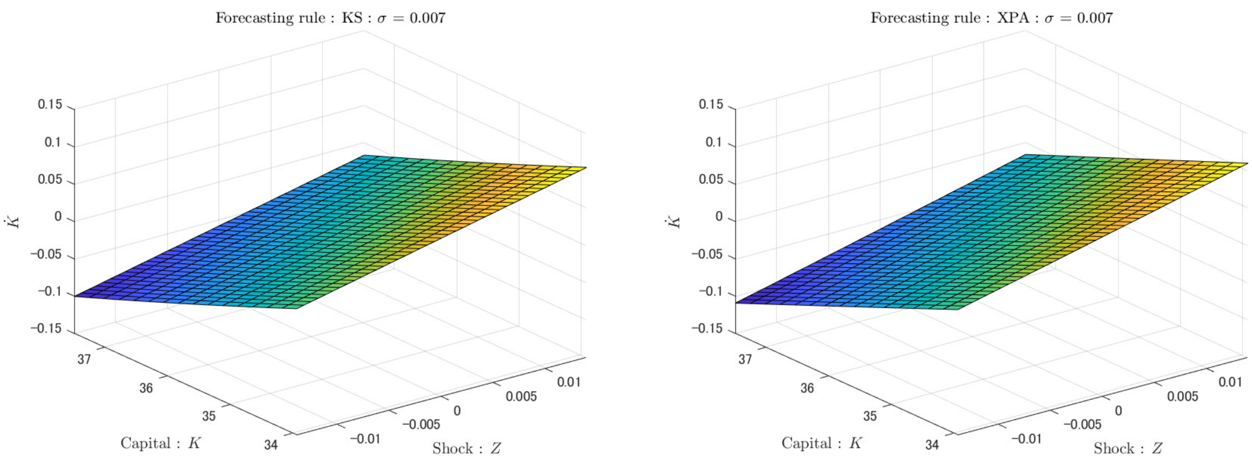
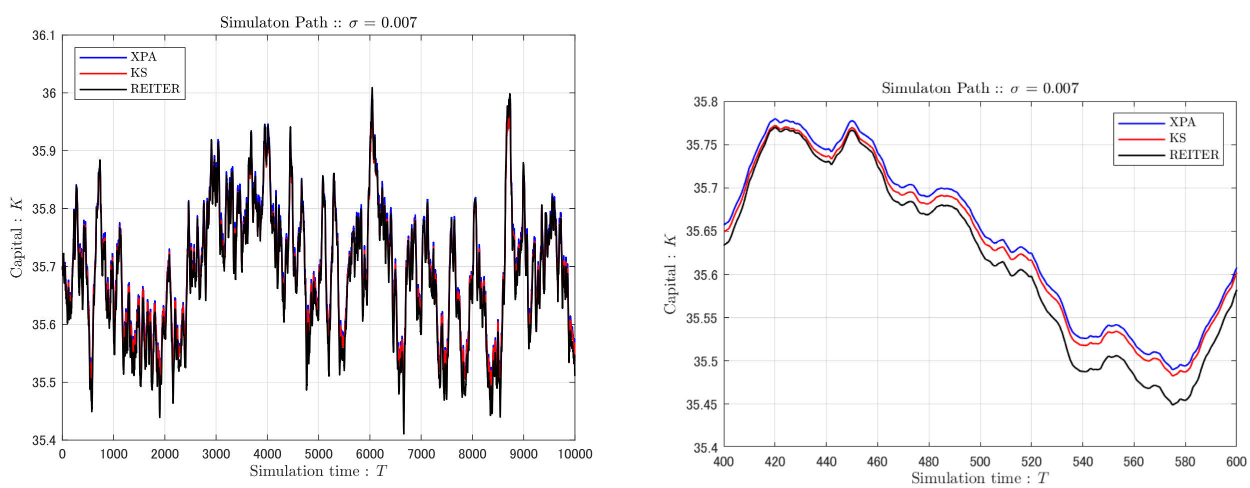


Figure 2: Simulation paths



XPA and KS solve the model nonlinearly, whereas REITER solve the model with linear approximation. As the law of motion for aggregate capital is concave, the error between the simulation results of the nonlinear methods (XPA and KS) and the linear method (REITER) is small when capital is high. On the other hand, when capital is low, the error between the simulation results of the nonlinear and linear methods is large. Thus, in Figure 4, we show that the simulation path of REITER diverges from those of XPA and KS as the value of capital decreases.

3.3 Efficiency

Table 2 summarizes the time that it takes to solve the model with each algorithm. Comparing the results of XPA and KS, we can see that XPA solves the model much faster than KS because it does not use simulations. Comparing the computation time of XPA and that of REITER, we can see that REITER is slightly faster than XPA. However, the difference between the two is not that large (XPA: 2.4 seconds vs. REITER: 0.5 seconds).[5]

Table 2: Computation time

Algorithm	Computation time
XPA (Our method)	2.478 sec
Krusell-Smith	122.101 sec
REITER (Ahn et al., 2018)	0.501 sec

Notes: Computations are done on a laptop with Intel Core i7-9750H and 16GB RAM using MATLAB R2019a.

3.4 Accuracy

The accuracy of solving the model is discussed using the Denhaan errors proposed by den Haan (2010). If the Denhaan errors are large in one algorithm, the model solution by using this method is not accurate because households are acting based on an erroneous forecasting rule. In this study, we simulate a 10000 period of time with each algorithm and use the results from $\{K_t^*\}_{t\in[0,T]}$ and

[5] Note that we use MATLAB and no parallelization, so the gap might be smaller when we use a faster language and/or parallelization.

$\{\tilde{K}_t\}_{t\in[0,T]}$ to measure the Denhaan error

$$\varepsilon_{DH}^{MAX} \equiv 100 \cdot \max_{t\in[0,T]} |\ln \tilde{K}_t - \ln K_t^*|,$$

$$\varepsilon_{DH}^{MEAN} \equiv 100 \cdot \frac{\sum_{t\in[0,T]} |\ln \tilde{K}_t - \ln K_t^*|}{T}.$$

Table 3 and Figure 3 summarize the Denhaan errors for each algorithm. It is clear from the table that when aggregate uncertainty is small, i.e., when σ is low, there is not that much difference in the Denhaan errors for each algorithm. However, when aggregate uncertainty is large, i.e., when σ is large, the Denhaan errors for REITER is considerably larger than those for XPA and KS. Therefore, it is clear that XPA is able to compute the model more accurately than REITER when aggregate uncertainty is large. Also, in the benchmark case, the Den Haan errors of XPA are smaller than those of KS when the volatility of TFP is $\sigma = 5.0\%$.[6]

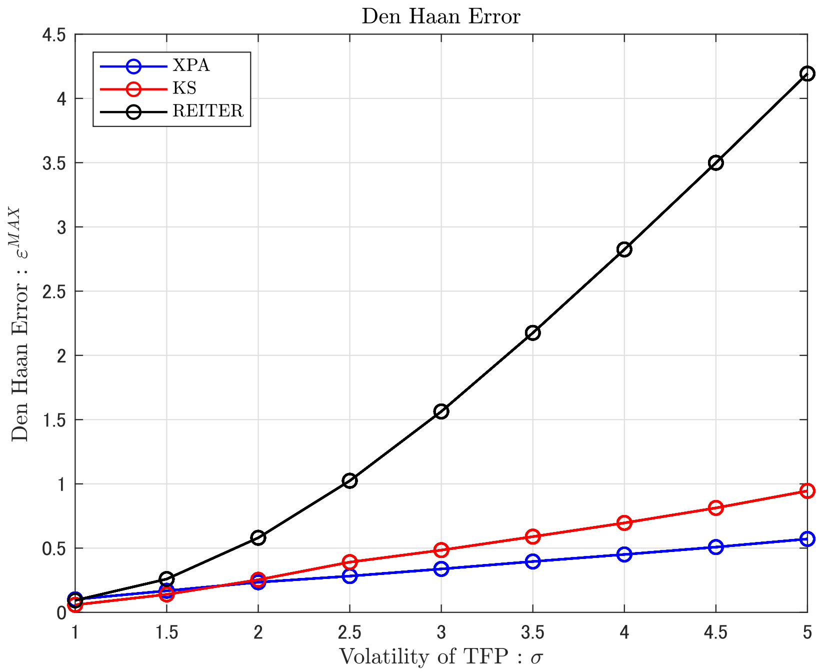
Table 3: Den Haan Errors

Agg Shock σ	0.01%	0.1%	0.7%	1.0%	5.0%
$\varepsilon_{\mathrm{DH\ XPA}}^{\mathrm{MAX}}$	0.000%	0.009%	0.071%	0.101%	0.571%
$\varepsilon_{\mathrm{DH\ XPA}}^{\mathrm{MEAN}}$	0.000%	0.002%	0.016%	0.024%	0.136%
$\varepsilon_{DH\ KS}^{MAX}$	0.000%	0.004%	0.035%	0.058%	0.945%
$\varepsilon_{DH\ KS}^{MEAN}$	0.000%	0.003%	0.023%	0.037%	0.690%
$\varepsilon_{DH\ REITER}^{MEAN}$	0.000%	0.001%	0.038%	0.078%	3.477%

Notes: For each value of σ, we adjust the values of the grid so that the maximum and minimum values of the capital stock obtained in the simulation are within the range of the grid of the capital stock.

[6] We also confirm that our results hold when the persistence of TFP, $1-\eta$, is lowered so that $\eta = 0.5$ or $\eta = 0.75$. See Appendix C.

Figure 3: Comparison of the Den Haan Error when aggregate uncertainty is large



4 Conclusion

In this paper, we apply the explicit aggregation (XPA) algorithm proposed by den Haan and Rendahl (2010) to the standard heterogeneous agent model with aggregate uncertainty (Krusell and Smith, 1998) in continuous time.

We find that, compared with the REITER algorithm that is popular to solve the heterogeneous agent model with aggregate uncertainty in continuous time, the XPA algorithm is able to solve the model as nearly fast as the REITER algorithm does, and more accurate than this algorithm in the case when aggregate uncertainty is large.

In our future tasks, we will apply our algorithm to the heterogeneous-agent New Keynesian (HANK) model (e.g., Bayer et al., 2019; Gornemann et al., 2016; Kaplan et al., 2018) in continuous time with the zero lower bound (ZLB) on nominal interest rates. Also, Terry (2017) shows that the model can be solved more accurately by using a projection method like our method than a perturbation method, when there is a dependency between aggregate uncertainty and idiosyncratic risk. Our algorithm can be applied when this is the case. In these applications, our algorithm will be even more interesting and useful than other algorithms.

References

Achdou, Y., J. Han, J.-M. L. Lasry, P.-L. L. Lions, and B. Moll (2017): "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach," NBER Working Papers 23732, National Bureau of Economic Research, Inc.

Ahn, S., G. Kaplan, B. Moll, , T. Winberry, and C. Wolf (2018): "When Inequality Matters for Macro and Macro Matters for Inequality," *NBER Macroeconomics Annual*, 1–75.

Bayer, C. and R. Luetticke (2020): "Solving heterogeneous agent models in discrete time with many idiosyncratic states by perturbation methods," *Quantitative Economics*, forthcoming.

Bayer, C., R. Luetticke, L. Pham-Dao, and V. Tjaden (2019): "Precautionary Savings, Illiquid Assets, and the Aggregate Consequences of Shocks to Household Income Risk," *Econometrica*, 87, 255–290.

Childers, D. (2018): "Solution of Rational Expectations Models with Function Valued States," Working paper.

den Haan, W. J. (2010): "Comparison of Solutions to the Incomplete Markets Model with Aggregate Uncertainty," *Journal of Economic Dynamics and Control*, 34, 4–27.

den Haan, W. J. and P. Rendahl (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty Using Explicit Aggregation," *Journal of Economic Dynamics and Control*, 34, 69–78.

Fernández-Villaverde, J., S. Hurtado, and G. Nuño (2019): "Financial Frictions and the Wealth Distribution," NBER Working Papers 26302, National Bureau of Economic Research, Inc.

Gornemann, N. M., K. Kuester, and M. Nakajima (2016): "Doves for the Rich, Hawks for the Poor? Distributional Consequences of Monetary Policy," Board of Governors of the Federal Reserve System International Finance Discussion Papers 1167.

Kaplan, G., B. Moll, and G. L. Violante (2018): "Monetary Policy According to HANK," *American Economic Review*, 108, 697–743.

Khan, A. (2017): "Large Recessions in an Overlapping Generations with Unemployment," 2017 Meeting Papers 1559, Society for Economic Dynamics.

Khan, A. and J. K. Thomas (2003): "Nonconvex Factor Adjustments in Equilibrium Business Cycle Models: Do Nonlinearities Matter?" *Journal of Monetary Economics*, 50, 331–360.

——— (2008): "Idiosyncratic Shocks and the Role of Nonconvexities in Plant and Aggregate Investment Dynamics," *Econometrica*, 76, 395–436.

Kim, H. (2018): "Inequality, Portfolio Choice, and the Business Cycle," Working Paper.

Krueger, D., K. Mitman, and F. Perri (2016): "Macroeconomics and Household Heterogeneity," in *Handbook of Macroeconomics*, ed. by J. B. Taylor and H. Uhlig, Elsevier, vol. 2, 843–921.

Krusell, P. and A. A. J. Smith (1998): "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, 106, 867–896.

Okahata, N. (2018): "An Alternative Solution Method for Continuous-Time Heterogeneous Agent Models with Aggregate Shocks," Mimeo.

Reiter, M. (2009): "Solving heterogeneous-agent models by projection and perturbation," *Journal of Economic Dynamics and Control*, 33, 649–665.

——— (2010a): "Approximate and Almost-Exact Aggregation in Dynamic Stochastic Heterogeneous-Agent Models," Economics Series 258, Institute for Advanced Studies.

——— (2010b): "Solving the incomplete markets model with aggregate uncertainty by backward induction," *Journal of Economic Dynamics and Control*, 34, 28–35.

Sunakawa, T. (2020): "Applying the Explicit Aggregation Algorithm to Heterogeneous Macro Models," *Computational Economics*, 55, 845–874.

Terry, S. J. (2017): "Alternative Methods for Solving Heterogeneous Firm Models," *Journal of Money, Credit and Banking*, 49, 1081–1111.

Young, E. R. (2005): "Approximate Aggregation," Mimeo, University of Virginia.

Appendix (not for publication)

A The Krusell-Smith model in continuous time

A.1 Environments

Households

Households face idiosyncratic uncertainty for labor productivity and the borrowing constraint $a_t \geq 0$. There are two states of labor productivity for each household, z_e and z_u, which follows the Poisson process with arrival rates λ_e and λ_u. z_e shows that the household is employed and z_u indicates that the household is unemployed.

If the household is employed, she/he receives the labor income after taxation $(1-\tau)w_t$. When the household is unemployed, she/he gets the unemployment insurance bw_t financed by the labor income tax.

Each household chooses their consumption in each period to maximize their expected life-time utility by taking the wage rate w_t and the interest rate r_t as given.

$$
\begin{aligned}
& v = \max \mathbb{E}_0 \left[\int_0^\infty e^{-\rho t} \frac{c_t^{1-\theta}}{1-\theta} dt \right], \\
\text{s.t.} \quad & da_t = (r_t a_t + (1-\tau) z_t w_t + (1-z_t) b w_t - c_t) dt, \ a_t \geq 0, \\
& z_t \in \{z_e, z_u\}, \ z_e = 1, \ z_u = 0.
\end{aligned}
$$

The instantaneous utility function is the form of the CRRA function, ρ is the rate of time preference, and θ is the degree of the relative risk aversion.

Firm

The representative firm produces the final good Y_t by inputs of capital K_t and labor L_t. The production function takes the form of Cobb-Douglas function

$$
Y_t = e^{Z_t} K_t^{\alpha} N_t^{1-\alpha},
$$

where α is capital share. Z_t is the logarithm of the total factor productivity (TFP) following the Ornstein-Uhlenbeck process

$$dZ_t = \eta(\bar{Z} - Z_t)dt + \sigma dW_t, \quad \bar{Z} = 0,$$

where dW_t follows the Wiener process. η is the persistence of the TFP and σ is the volatility of the TFP. This process is similar to the AR(1) process in discrete time.

The wage rate and the interest rate are obtained from the first-order conditions for the profit maximization problem as follows:

$$w_t = (1-\alpha)e^{Z_t}K_t^{\alpha}N_t^{-\alpha}, \;\; r_t = \alpha e^{Z_t}K_t^{\alpha-1}N_t^{1-\alpha} - \delta,$$

where δ is the depreciated rate of capital.

Government

The government imposes a tax on labor income to finance unemployment compensation. The government's budget is balanced as below

$$\tau w \mu_e = bw\mu_u.$$

That is, the government's tax revenue of labor income is equal to the government's expenditure to finance unemployment insurance. $\mu_e = \phi(z_e)$ is the share of employment and $\mu_u = \phi(z_u)$ is the share of unemployment in the economy.

A.2 Stationary equilibrium without aggregate uncertainty

We define the steady-state without aggregate uncertainty by setting $Z_t = 0$ for all $t \geq 0$. In the steady-state without aggregate uncertainty, the following equations are satisfied:

- The Hamilton-Jacobi-Bellman (HJB) equation and the policy function for households

$$\rho v(a,z) = \max_c u(c) + v_a(a,z)(ra + (1-\tau)zw + (1-z)bw - c) + \lambda_z(v(a,z') - v(a,z))$$

$$s(a,z) = ra + (1-\tau)zw + (1-z)bw - c(a,z)$$

- The Fokker-Planck equation

$$0 = \frac{\partial(s(a,z)g(a,z))}{\partial a} - \lambda_z g(a,z) + \lambda_{z'} g(a,z')$$

- The wage rate and the interest rate

$$w = (1-\alpha)K^{\alpha}L^{-\alpha},\ r = \alpha K^{\alpha-1}L^{1-\alpha} - \delta$$

- The government's budget constraint

$$\tau w \mu_e = bw\mu_u,\ \mu_e = \int g(a,z_e)da,\ \mu_u = \int g(a,z_u)da,$$

- The capital and the labor markets clear

$$K = \sum_z \int ag(a,z)da,\ L = \sum_z \int zg(a,z)da$$

B Details of the XPA algorithm

We explain the details of the XPA algorithm following den Haan and Rendahl (2010) and Sunakawa (2020). First, we rewrite the wealth distribution $g(a,z)$ using the conditional probability

$$\begin{aligned} g(a|z) &= \frac{g(a,z)}{\int g(a,z)da} = \frac{g(a,z)}{\phi(z)} \\ \Leftrightarrow g(a,z) &= g(a|z)\phi(z) \end{aligned}$$

where $\phi(z) = \int g(a,z)da$ is equal to the proportion of households with labor productivity z in the economy.[7] Then we can rewrite the forecasting rule using the conditional distribution of wealth as

$$\begin{aligned}\dot{K}(K,Z) &= \sum \int s(a,z;K,Z)g(a,z)da \\ &\approx \sum s\left(\int ag(a|z)da, z; K, Z\right)\phi(z) \\ &= \sum s(K(z), z; K, Z)\phi(z)\end{aligned}$$

where $K(z) = \int ag(a|z)da$ is the capital conditioned on the labor productivity z. We compute $K(z)$ by the following equations:

$$\begin{aligned}K(z) &= \psi(z)K_{ss}, \;\; \psi(z) \equiv \frac{K_{ss}(z)}{K_{ss}} = \frac{K_{ss}(z)}{\sum_z \int ag_{ss}(a,z)da}, \\ K_{ss}(z) &= \int ag_{ss}(a|z)da = \frac{\int ag_{ss}(a,z)da}{\phi(z)},\end{aligned}$$

where K_{ss} and g_{ss} are capital and the wealth distribution at the steady-state without aggregate uncertainty. Note that the ratio of the capital conditioned on z to the aggregate capital, $\psi(z) = K_{ss}(z)/K_{ss}$, can be easily obtained in the steady-state calculation.[8]

Moreover, following den Haan and Rendahl (2010), we do the bias correlation. In the explicit aggregation, we *assume* that the household's policy function $s(a,z;K,Z)$ is linear at $a = K(z)$ so that $\int s(a,z;K,Z)g(a|z)da \approx s\left(\int K(z), z; K, Z\right)$ holds. Therefore the forecasting rule may have biases from Jensen's inequality. We compute the steady-state counterparts to correct the biases:

$$\begin{aligned}\xi(z) &= \dot{K}_{ss}(z) - s_{ss}(K_{ss}(z), z), \\ \dot{K}_{ss}(z) &= \int s_{ss}(a,z)g_{ss}(a|z)da = \frac{\int s_{ss}(a,z)g_{ss}(a,z)da}{\phi(z)}.\end{aligned}$$

Again, $\xi(z)$ can be computed with a negligible cost in the steady state. Finally, we can write the forecasting rule as follows

[7] We assume the share of employment $\phi(z_e)$ $(= 1 - \phi(z_u))$ is time-invariant, although it is straightforward to make the employment measure be time-variant and depend on aggregate uncertainty as in Krusell and Smith (1998).

[8] We assume that $\psi(z)$ is constant even with aggregate uncertainty. The details of steady-state calculation are found in Appendix A.2.

$$\dot{K}(K,Z) = \sum_z \{s(K(z),z,K,Z) + \xi(z)\}\phi(z).$$

That is, to obtain the forecasting rule, we just need to evaluate the policy function at $a = K(z)$. The value of $K(z)$ may not be on the grid of a, so we use linear interpolation.

Summary of XPA Algorithm

In summary, we perform computations to solve the Krusell-Smith model with the XPA algorithm as follows. As mentioned above, the XPA algorithm is fast because it does not use any simulations.

1. Compute the deterministic steady state without aggregate uncertainty to obtain the conditional capital ratio $\psi(z)$ and the bias correlation term for correcting the forecasting rule $\xi(z)$.

2. (Inner loop) Solve the HJB equation for the policy function taking the forecasting rule as given.

3. (Outer loop) Compute the forecasting rule without simulations taking the policy function as given. The bias correction is also done.

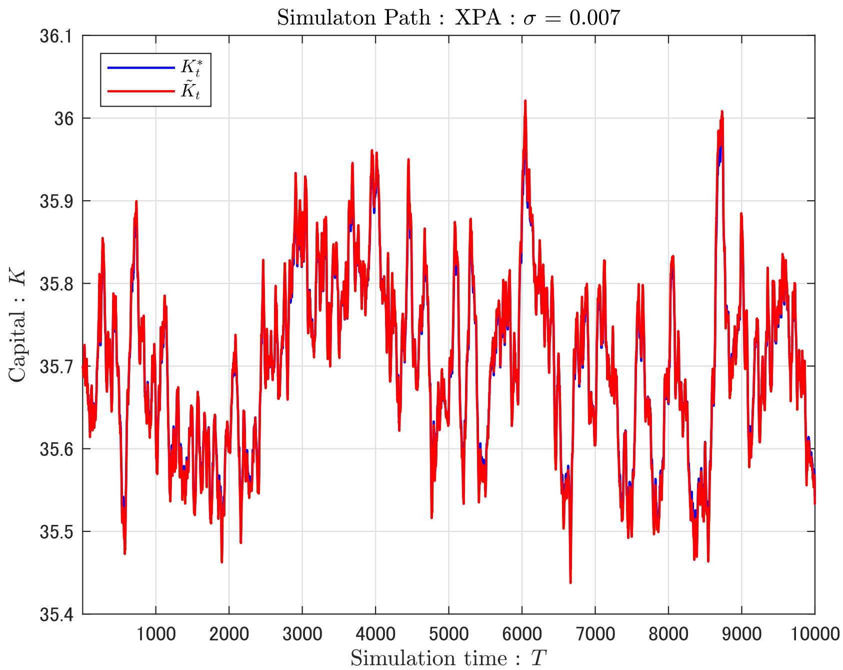
4. Repeat 2–3 until the forecasting rule converges.

C Further numerical results

In Figure 4, we show the results of the simulation path in XPA. The red line $\{\tilde{K}_t\}_{t\in[0,T]}$ in the figure shows the simulation obtained only from the forecasting rule for 10000 periods, while the bule line $\{K_t^*\}_{t\in[0,T]}$ shows the simulation obtained from the full model including the forecasting rule and the household HJB equation for 10000 periods.[9] This is known as the Denhaan's (2010)

[9]The time interval $dt = 0.25$ used in the simulation is the same as the time interval of Ahn et al. (2018). We use linear interporation for aggregate capital and TFP to calculate the wealth distribution at each point of time.

Figure 4: Simulation path in XPA (Denhaan's fundamental plot)



fundamental plot showing the accuracy of the solution. It is clear from the figure that the capital paths resulting from these simulations are very close.[10]

We also check the robustness of our results with respect to the persistence of the TFP. In Table 4, we show that the Denhaan errors of KS, XPA, and REITER when the persistence of TFP, $1-\eta$, is lowered so that $\eta = 0.5$ or $\eta = 0.75$. It is clear that, regardless of the persistence, the Denhaan errors of REITER are larger than those of XPA and KS when aggregate uncertainty is large.[11]

[10]If the red and blue lines are close, households approximately act on the correct forecasting rule. If not, households are working on a wrong forecasting rule. Therefore, the divergence between the two lines indicates that the model is not solved correctly based on rational expectations.

[11]However, unlike in the benchmark case of $\eta = 0.25$, the Den Haan error of XPA is larger than that of KS when the volatility of TFP is $\sigma = 5\%$ and $\eta = 0.5$ or $\eta = 0.75$.

Table 4: Den Haan Errors: Robustness

a. Case of $\eta = 0.50$

Agg Shock σ	0.01%	0.1%	0.7%	1.0%	3.0%	5.0%
$\varepsilon^{\text{MAX}}_{\text{DH XPA}}$	0.002%	0.019%	0.143%	0.198%	0.607%	1.051%
$\varepsilon^{\text{MEAN}}_{\text{DH XPA}}$	0.000%	0.005%	0.039%	0.054%	0.163%	0.301%
$\varepsilon^{MAX}_{DH\ KS}$	0.000%	0.002%	0.021%	0.034%	0.220%	0.432%
$\varepsilon^{MEAN}_{DH\ KS}$	0.000%	0.001%	0.007%	0.010%	0.049%	0.185%
$\varepsilon^{MEAN}_{DH\ REITER}$	0.000%	0.001%	0.055%	0.121%	1.953%	4.778%

b. Case of $\eta = 0.75$

Agg Shock σ	0.01%	0.1%	0.7%	1.0%	3.0%	5.0%
$\varepsilon^{\text{MAX}}_{\text{DH XPA}}$	0.003%	0.038%	0.276%	0.399%	1.110%	1.980%
$\varepsilon^{\text{MEAN}}_{\text{DH XPA}}$	0.001%	0.012%	0.089%	0.129%	0.378%	0.669%
$\varepsilon^{MAX}_{DH\ KS}$	0.000%	0.007%	0.059%	0.096%	0.439%	0.994%
$\varepsilon^{MEAN}_{DH\ KS}$	0.000%	0.002%	0.012%	0.018%	0.094%	0.212%
$\varepsilon^{MEAN}_{DH\ REITER}$	0.000%	0.002%	0.102%	0.264%	3.317%	7.407%