Model-Adaptive Approach to Dynamic Discrete Choice Models with Large State Spaces*

Ertian Chen†

January 26, 2025

Abstract

Estimating dynamic discrete choice models with large state spaces poses computational difficulties. This paper develops a novel model-adaptive approach to solve the linear system of fixed point equations of the policy valuation operator. We propose a *model-adaptive* sieve space, constructed by iteratively augmenting the space with the residual from the previous iteration. We show both theoretically and numerically that model-adaptive sieves dramatically improve performance. In particular, the approximation error decays at a superlinear rate in the sieve dimension, unlike a linear rate achieved using conventional methods. Our method works for both conditional choice probability estimators and full-solution estimators with policy iteration. We apply the method to analyze consumer demand for laundry detergent using Kantar's Worldpanel Take Home data. On average, our method is 51.5% faster than the conventional methods in solving the dynamic programming problem, making the Bayesian MCMC estimator computationally feasible. The results confirm the computational efficiency of our method in practice.

Keywords: Dynamic discrete choice; Adaptive sieve; Policy iteration; Storable goods.

JEL codes: C61, C63, C25, L66.

*I would like to thank Lars Nesheim, Tim Christensen and Dennis Kristensen for their continuous guidance and support. I am grateful to Karun Adusumilli, Victor Aguirregabiria, Jason Blevins, Austin Brown, Cuicui Chen, Ben Deaner, Hugo Freeman, Joao Granja, Jiaying Gu, Yao Luo, Angelo Melino, Bob Miller, Matthew Osborne, Aureo de Paula, Martin Pesendorfer, Eduardo Souza-Rodrigues, Ao Wang for helpful discussions and comments. I also thank seminar participants at UCL, LSE, IFS/LSE/UCL IO workshop, University of Bristol, University of Toronto, and conference audiences at 2024 Bristol Econometric study group, 2024 Midwest Econometrics Group Conference. The use of Kantar data does not imply the endorsement of Kantar in relation to the interpretation or analysis of the data. All errors are on my own.

†Department of Economics, University College London, CeMMAP and IFS. ertian.chen.19@ucl.ac.uk.

1 Introduction

Dynamic Discrete Choice (DDC) models with large state spaces have become increasingly popular due to their ability to capture decision-making processes in complex high-dimensional settings. However, estimating DDC models in these settings poses significant computational challenges. This paper introduces a novel model-adaptive approach to solve the linear system of equations of the policy valuation operator. Our goal is to provide a fast and easily implementable method to solve the equations within a pre-specified tolerance. As policy valuation steps are fundamental to both Conditional Choice Probability (CCP) estimators and full-solution estimators with policy iteration, our approach offers a useful numerical tool across various empirical applications, expanding the set of complex high-dimensional settings in which DDC models can be used.

The policy valuation operator, as described in Aguirregabiria and Mira [2002], involves solving for the value function implied by an arbitrary policy function, which may not necessarily be optimal. This value function represents the expected discounted utility if an individual behaves according to that policy function. CCP estimators (e.g., Hotz and Miller [1993], Aguirregabiria and Mira [2002, 2007], Pesendorfer and Schmidt-Dengler [2008], and Arcidiacono and Miller [2011]) use the policy valuation operator to solve for the value function given a consistent estimator of CCPs. In full solution estimators with policy iteration, the policy valuation operator is employed at each policy iteration step. However, in models with large state spaces, policy valuation remains computationally demanding, necessitating the use of numerical methods. The accuracy of these numerical methods is crucial for obtaining reliable estimates. Dubé et al. [2012] shows that loose tolerance thresholds can lead to bias in parameter estimates. Therefore, there is a clear need for fast and accurate numerical methods for DDC models with large state spaces.

This paper develops a model-adaptive approach to solve the linear system of fixed point equations of the policy valuation operator. The primary goal is to achieve a pre-specified tolerance while significantly reducing computational costs, enabling the use of policy valuation tools in a wide range of empirical applications. We call our approach model-adaptive as it designs the sieve space based on the model primitives (such as the transition density and utility function), and the algorithm itself selects the sieve dimension. At each step, our approach augments the sieve space with residuals from the previous iteration and projects the value function onto the augmented sieve space. It achieves a faster decay rate of the approximation error than conventional methods, such as Successive Approximation (SA) and Temporal Difference (TD). Formally, we show that the approximation error decays at a

superlinear rate in the sieve dimension (number of iterations), while SA achieves a linear rate and TD fails to achieve a superlinear rate. Furthermore, the sieve space and its dimension are automatically constructed by the algorithm, eliminating the need for researchers to design the sieve space and choose its dimension. Consequently, our method is easy to implement and converges faster than conventional methods, offering substantial computational savings.

The main computational cost depends on the number of iterations and matrix-vector multiplication operations. Our approach attains superlinear convergence on the approximation error, which substantially reduces the number of iterations required to reach a desired level of tolerance. Formally, our approach achieves an approximation error upper bound of $O((\frac{C_1}{\sqrt{k}})^k)$, where k is the number of iterations and C_1 is a constant. Furthermore, the bound can be improved to $O((\frac{C_2}{k})^k)$ if the transition density has continuous partial derivatives. Notably, only the constants in the upper bound depend on the dimension of the state space and the discount factor. Therefore, our method is well-suited for models with large state spaces and large discount factors, such as dynamic consumer demand models (e.g., Hendel and Nevo [2006a] and Wang [2015]). Moreover, it works particularly well for models with smooth transition densities, such as autoregressive (AR) processes, which are commonly used in empirical applications (e.g., Sweeting [2013], Huang and Smith [2014], Kalouptsidi [2014], Grieco et al. [2022], and Gerarden [2023]).

We provide implementations for both discrete and continuous state spaces. For discrete state spaces, the implementation requires only matrix operations as the system of equations takes the form of a finite-dimensional linear system. For continuous state spaces, we employ numerical integration to approximate the integral of the policy valuation step. Thus, there is a trade-off between simulation error and computational cost: increasing the number of grid points reduces simulation error at the cost of solving a larger linear system within a given tolerance. The computational cost of matrix-vector multiplication operations increases with the number of grid points. Therefore, we analyze the impact of the number of grid points on the number of iterations required to achieve a desired tolerance. We show that the number of iterations required for convergence remains approximately the same for all sufficiently large numbers of grid points. As a result, we can expect the number of iterations to be independent of the number of grid points for numerical integration. Therefore, our method allows researchers to reduce simulation error by increasing the number of grid points up to the computational limits of matrix-vector multiplication. Fast matrix-vector multiplication algorithms can further accelerate the computation (e.g., Rokhlin [1985], Greengard and Rokhlin [1987], and Hackbusch and Nowak [1989]). In addition, matrix-vector multiplication is amenable to GPU acceleration, which can further reduce the.

We illustrate the performance of our approach using three Monte Carlo simulations. We first simulate the bus engine replacement problem to visualize the convergence behavior of our method. The plot of our approximation solution shows that the method uses a few iterations to find a good sieve space. After that, it converges rapidly to the true solution.

Second, we analyze a model for dynamic consumer demand for storable goods similar to Hendel and Nevo [2006a]. Comparing five solution algorithms, we demonstrate that our method is 51.5% faster than conventional methods such as successive approximation and an exact equation solver. It is also 100 times faster than value function iteration methods. These substantial computational savings open the door to the use of Bayesian MCMC estimators Chernozhukov and Hong [2003], which are well-suited as the likelihood function is not differentiable in utility parameters.

Third, we examine a dynamic firm entry and exit problem in Aguirregabiria and Magesan [2023]. We solve the dynamic programming problem by policy iteration using our model-adaptive approach to solve the linear system of fixed point equations. We vary the discount factor and number of grid points for numerical integration to evaluate the performance. The computational times confirm that our method improves the computational efficiency of policy iteration. The results show that the numbers of iterations required for convergence are approximately the same regardless of the numbers of grid points. Moreover, the number of iterations only slightly increases as the discount factor approaches one. We also compare the combination of our method with the Nested Pseudo-Likelihood (NPL) algorithm with the "Full-Solution with Euler Equation" (FSEE) method from Aguirregabiria and Magesan [2023]. We find our approach is twice as fast as the FSEE method. Finally, we compare our method with TD and SA methods to assess the performance of the different methods. The simulation results show that our method outperforms SA in terms of computational time and TD in terms of approximation error.

We apply our method to a dynamic consumer demand model for laundry detergent using Kantar's Worldpanel Take Home data. For each household size, we separately estimate the dynamic parameters using the Bayesian MCMC estimator Chernozhukov and Hong [2003]. At each MCMC step, we solve the dynamic programming problem by policy iteration with our model-adaptive approach. The results confirm the computational efficiency of our method in practice. We also simulate the long-run elasticities, which reveal the heterogeneous substitution patterns across different household sizes.

1.1 Related Literature

Sieve approximation methods have been used to solve dynamic programming problems (e.g., Norets [2012], Arcidiacono et al. [2013], and Wang [2015]). They have been widely used to solve the linear equations in the policy valuation step. Applications include Hendel and Nevo [2006a], Sweeting [2013], and Bodéré [2023]. Recent work approximates the solution to the linear equation by temporal difference (see Adusumilli and Eckardt [2024]). However, those methods require researchers both to design the space of basis functions (e.g., polynomials, splines, or neural networks) and choose the sieve dimension (e.g., the degree of polynomials, the number of knots, and the number of hidden layers). The best choice for each application is almost always unclear. In contrast, our model-adaptive approach constructs a model-adaptive sieve space using the algorithm itself to design that space. The sieve dimension (i.e., the number of iterations) is also determined by the algorithm. The method is guaranteed to achieve the pre-specified tolerance. Finally, we show that the approximation error decays at a superlinear rate in the sieve dimension. Other methods like TD do not achieve this rate.

Successive Approximation (see Kress [2014]), also known as fixed point iteration Judd [1998], is an iterative method to solve the fixed point equations of the policy valuation operator. The computational cost depends on the number of iterations and the number of matrix-vector multiplication operations. The convergence of SA relies on the β-contraction property where β is the discount factor. Therefore, the number of iterations of SA increases significantly as the discount factor approaches one, making the methods computationally demanding. In contrast, the superlinear convergence of our method does not rely on the contraction property, making it particularly well-suited for models with large discount factors such as consumer demand models (e.g., Hendel and Nevo [2006a] and Wang [2015]). Moreover, our method can outperform SA even for small discount factors as SA achieves linear convergence while our method achieves superlinear convergence.

For continuous state spaces, we employ numerical integration to approximate the integral of the policy valuation operator, which implicitly discretizes the state space. Discretization is widely used in economics (e.g., Sweeting [2013], Kalouptsidi [2014], Huang et al. [2015], and Bodéré [2023]). Rust [1997a,b] study the simulation error from numerical integration and assume the discretized equation can be solved exactly. However, there is a trade-off between simulation error and computational cost of solving the discretized equation. Increasing the number of grid points reduces the simulation error while increasing the size of the linear system to be solved; potentially making it computationally infeasible to solve the system exactly. Sieve approximation is used to solve the discretized equation (e.g., Hendel and Nevo

[2006a], Sweeting [2013], and Bodéré [2023]). Instead, we propose to solve the discretized equation within a given tolerance while minimizing computational cost. The computational cost of our method depends on the number of iterations and the number of matrix-vector multiplication operations. We can expect that the number of iterations is small and independent of the number of grid points. Therefore, our method enables researchers to reduce simulation error up to the computational limits of matrix-vector multiplication.

Outline: The remainder of the paper is organized as follows. Section 2 reviews DDC models and the policy valuation operator. Section 3 presents the model-adaptive approach, its implementation and computational cost. Section 4 describes the theoretical properties. Section 5 reports results from three Monte Carlo simulations. Section 6 applies our method to a consumer demand for storable goods. Section 7 concludes. The proofs are in Appendix A. Online Appendix B contains details of algorithms of simulations and empirical application, and the supporting lemmas.

Notation: Let $\mathbb{X}$ be the support of x, and $\mathbb{A} := \{0, 1, \cdots, A-1\}$. $\mathbb{X}$ can be discrete or continuous. For a probability distribution μ on $\mathbb{X}$, which in the continuous case is assumed to be absolutely continuous with respect to Lebesgue measure, let $L^2(\mathbb{X}, \mu)$ denote the space of square-integrable functions on $\mathbb{X}$. Let $\langle \cdot, \cdot \rangle_\mu$ and $\| \cdot \|_\mu$ denote the inner product and norm induced by μ. Let ν_{Leb} denote the Lebesgue measure, and let $\langle \cdot, \cdot \rangle$, $\| \cdot \|$, and $L^2(\mathbb{X})$ denote the inner product, norm, and L^2-space, respectively. We suppress ν_{Leb} for notational simplicity. Let $\| \cdot \|_\infty$ denote the sup-norm of a vector. For a linear operator, $\mathcal{A} : L^2(\mathbb{X}) \longmapsto L^2(\mathbb{X})$, denote by $\|\mathcal{A}\|_{op} := \sup_{h \in L^2(\mathbb{X}), \|h\|=1} \|\mathcal{A}h\|$ the operator norm. Let $\mathcal{I}$ be the identity operator. Denote by $\asymp$ the weak equivalence of sequence, i.e., $a_n \asymp b_n$ iff $c_1 \leq \frac{a_n}{b_n} \leq c_2$ for some positive constants c_1, c_2. Let $\mathbb{S}_k := \text{span}\{\phi_1, \cdots, \phi_k\}$ be a sieve space where $(\phi_1, \cdots, \phi_k)$ are basis functions. Denote by $\Phi_k := (\phi_1, \cdots, \phi_k)$ the vector of basis functions. Let $\Pi_{\mathbb{S}_k}$ be the projection operator onto $\mathbb{S}_k$, i.e., $\Pi_{\mathbb{S}_k}(V) := \text{argmin}_{h \in \mathbb{S}_k} \|V - h\|_\mu$.

2 Model

2.1 Framework

We study infinite horizon stationary dynamic discrete choice models as in Rust [1994]. In each discrete period $t \leq +\infty$, an individual chooses $a_t \in \mathbb{A}$ to maximize her discounted expected utility:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \left[u(x_t, a_t) + \varepsilon_t(a_t)\right] \middle| x_0, \varepsilon_0\right]$$

where $\beta \in (0,1)$ is the discount factor, $u(x_t, a_t)$ is the utility function, $x_t \in \mathbb{X}$ is the observable state (to researchers) that follows a first-order Markov process with a transition density $f(x_{t+1}|x_t, a_t)$, and $\varepsilon_t \in \mathbb{R}^A$ is a vector of unobservable i.i.d. type I extreme value with Lebesgue density $g(\varepsilon_t)$. $\varepsilon_t(a_t)$ is the element of ε_t corresponding to action a_t.

Under regularity conditions (see Rust [1994]), the utility maximization problem has a solution and the optimal value function $V_{opt}(x, \varepsilon)$ is the solution to the *Bellman equation*:

$$V_{opt}(x, \varepsilon) = \max_{a \in \mathbb{A}} \left\{ u(x, a) + \varepsilon(a) + \beta \int V_{opt}(x', \varepsilon') f(x'|x, a) g(\varepsilon') dx' d\varepsilon' \right\}$$

where (x', ε') denotes the next period's state and utility shock. Under the i.i.d. assumption on utility shocks, integrating the utility shocks out, the *integrated Bellman equation* has the following form:

$$V_{opt}(x) = \int \max_{a \in \mathbb{A}} \left\{ v^*(x, a) + \varepsilon(a) \right\} g(\varepsilon) d\varepsilon$$

where $V_{opt}(x)$ is the integrated value function and $v^*(x, a)$ is the *conditional value function* defined as:

$$v^*(x, a) := u(x, a) + \beta \int V_{opt}(x') f(x'|x, a) dx'$$

The *conditional choice probability* (CCP) is the probability that action a is optimal conditional on observable state x defined by:

$$p^*(a|x) := \int \mathbb{1} \left\{ a = \operatorname*{argmax}_{a \in \mathbb{A}} \{ v^*(x, a) + \varepsilon(a) \} \right\} g(\varepsilon) d\varepsilon$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. Under the distributional assumption on utility shocks, CCPs take the following form:

$$p^*(a|x) = \frac{\exp(v^*(x, a))}{\sum_a \exp(v^*(x, a))}$$

The following map is derived in Arcidiacono and Miller [2011] as a corollary of *Hotz-Miller Inversion Lemma* by Hotz and Miller [1993], for $\forall\ (x, a) \in \mathbb{X} \times \mathbb{A}$:

$$V_{opt}(x) = v^*(x, a) - \log p^*(a|x) + \kappa \tag{2.1}$$

where κ is the Euler constant. The equation (2.1) establishes a crucial link between the integrated value function $V_{opt}(x)$, conditional value function $v^*(x, a)$, and conditional choice probabilities $p^*(a|x)$. It provides a powerful tool for the policy valuation step, which is

essential for both policy iteration and CCP estimators.

2.2 Policy Valuation Operator

The *policy valuation operator* (see Aguirregabiria and Mira [2002]) maps an arbitrary policy function to the value function using (2.1). The value function represents the expected discounted utility of an individual if she behaves today and in the future according to that policy, which is not necessarily optimal. Aguirregabiria and Mira [2002] shows that the value function is obtained by solving the following equation for V given a policy function p:

$$V(x) = \sum_a p(a|x) \left[u(x,a) + \kappa - \log p(a|x) + \beta \mathbb{E}_{x'|x,a} V(x') \right] \tag{2.2}$$

In this paper, we focus on solving (2.2) for a given policy function. Therefore, we introduce the following notations and suppress the dependence on p:

Definition 1. *For a given policy function p, define:*

(i) $f(x'|x) := \sum_a p(a|x) f(x'|x,a)$ *and* $\mathcal{T}V(x) := \beta \int f(x'|x) V(x') dx'$.

(ii) $u(x) = \sum_a p(a|x) u(x,a) + \kappa - \sum_a p(a|x) \log p(a|x)$.

Using this notation, we can rewrite (2.2) as a linear system of fixed point equations:

$$V = u + \mathcal{T}V \tag{2.3}$$

Both CCP estimators and the policy iteration method (see Howard [1960]) require solving equation (2.3) for V. For CCP estimators, p is replaced with its consistent estimator $\hat{p}$. At iteration i of the policy iteration, V_i is solved for p_i from the previous iteration. Subsequently, the *policy improvement step* updates the policy function as follows:

$$p_{i+1}(a|x) = \frac{\exp(v_i(x,a))}{\sum_a \exp(v_i(x,a))}$$

where $v_i(x,a) = u(x,a) + \beta \mathbb{E}_{x'|x,a} V_i(x')$. This process iterates until convergence of policy function is achieved, e.g., $\sup_{x,a} |p_{i+1}(a|x) - p_i(a|x)| \leq 10^{-5}$.

3 Model-Adaptive Approach

This section presents the model-adaptive approach, its implementation and computational cost. Our method employs the Conjugate Gradient (CG) method, an iterative approach for solving large linear systems. The CG method is commonly attributed to Hestenes et al. [1952]. For comprehensive textbooks, see Kelley [1995] and Han and Atkinson [2009].

The CG method was originally designed for solving linear systems with self-adjoint operators. However, the operator $\mathcal{T}$ is not necessarily self-adjoint. If $\mathcal{T}$ were self-adjoint with respect to the inner product space $\langle \cdot, \cdot \rangle_\mu$, then the Markov chain would be time-reversible, which is a strong assumption in many practical settings. Therefore, instead of solving $(\mathcal{I} - \mathcal{T})V = u$ for V directly, we propose to solve the following equation for y:

$$(\mathcal{I} - \mathcal{T})(\mathcal{I} - \mathcal{T}^*)y = u \tag{3.1}$$

and set $V = (\mathcal{I} - \mathcal{T}^*)y$ to solve $(\mathcal{I} - \mathcal{T})V = u$, where $\mathcal{T}^*$ is the adjoint operator of $\mathcal{T}$.[1]

For discrete state spaces, (3.1) boils down to a finite-dimensional linear system. For continuous state spaces, we will use deterministic numerical integration to approximate the integral in (3.1).

The adjoint operator $\mathcal{T}^*$ is defined with respect to an inner product space. The choice of the inner product does affect the convergence rate of the approximation error. Nevertheless, approximation solutions on different spaces all converge superlinearly under regularity conditions. To achieve the fastest decay of the approximation error when using CG, we propose to solve (3.1) on $L^2(\mathbb{X})$ and define $\mathcal{T}^*$ by the inner product $\langle \cdot, \cdot \rangle$. Lemma 9 formally discusses the convergence rate. Theorem 1(i) shows the existence and uniqueness of the solution to (3.1) on $L^2(\mathbb{X})$, denoted by y^*. Moreover, Theorem 1(ii) proves $V^* = (\mathcal{I} - \mathcal{T}^*)y^*$ where V^* is the solution to (2.3) on $L^2(\mathbb{X}, \mu)$. The *model-adaptive approach* is as follows:

Definition 2 (Model-adaptive Approach)**.** *Let $y_0 = 0$ be the initial guess and $r_0 = s_0 = u$ where r_0 is the initial residual and s_0 is the initial direction for updating y.*[2] *The sequence of the model-adaptive approximation solution to* (2.3)*, $\{V_k^{ma}\}_{k \geq 1}$, is given by:*

$$V_k^{ma} := (\mathcal{I} - \mathcal{T}^*)y_k$$

[1] The adjoint operator is similar to matrix transpose in the finite-dimensional case. For formal definition, see for example Han and Atkinson [2009] Chapter 2.6. For the specific adjoint operator used in my model-adaptive approach see Definition 2 below.

[2] If an alternative initial guess y_0 is available, we replace $r_0 = s_0 = u - (\mathcal{I} - \mathcal{T})(\mathcal{I} - \mathcal{T}^*)y_0$. All the subsequent steps remain unchanged.

where $\{y_k\}_{k\geq 1}$ *is iteratively defined by:*

$$\begin{aligned} y_k &= y_{k-1} + \alpha_{k-1} s_{k-1} & \alpha_{k-1} &= \frac{\|r_{k-1}\|^2}{\|(\mathcal{I}-\mathcal{T}^*)s_{k-1}\|^2} \\ r_k &= u - (\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^*)y_k & & \\ s_k &= r_k + \beta_{k-1} s_{k-1} & \beta_{k-1} &= \frac{\|r_k\|^2}{\|r_{k-1}\|^2} \end{aligned} \tag{3.2}$$

and $\mathcal{T}^*$ *is given by* $\mathcal{T}^*V(x) = \beta \int V(x')f(x|x')dx'$.

$$\mathcal{T}^*V(x) = \beta \int V(x')f(x|x')dx'$$

The algorithm stops if the norm of residuals $\|r_k\| \leq tol$ *where tol is a pre-specified tolerance.*

The core idea behind our approach is, at each iteration, to augment the sieve space with the residual from the previous iteration. By construction, the updates are orthogonal to previous updates. And, it is easy to show that $y_k \in \text{span}\{r_0, r_1, \cdots, r_{k-1}\}$ where $\{r_i\}_{i\leq k-1}$ is the sequence of residuals produced by previous iterations. In other words, $\{r_i\}_{i\leq k-1}$ is the model-adaptive sieve space after iteration k. In Theorem 2(ii), we show that y_k minimizes $\|(\mathcal{I}-\mathcal{T}^*)y - V^*\|$ over the model-adaptive sieve space, and Theorem 3 shows the superlinear convergence of the approximation error. The next section discusses the implementation and computational cost of our method.

3.1 Implementation

Discrete State Spaces: For discrete state spaces, (3.1) reduces to a finite-dimensional linear system as:

$$(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^T)y = u$$

where $\mathcal{I}$ is the identity matrix and $\mathcal{T}$ is the discounted transition matrix. Therefore, (3.2) only involves matrix-vector multiplications. The algorithm is as follows[3]:

Algorithm 1 (Model-adaptive Approach for Discrete State Spaces)**.**

- *Step 1:* For a given $f(x'|x)$, generate the matrix: $(\mathcal{I}-\mathcal{T})$.
- *Step 2:* Generate the matrix: $(\mathcal{I}-\mathcal{T}^T)$.
- *Step 3:* Given a tolerance, iterate algorithm (3.2) until convergence.

[3]We briefly review the SA algorithm in Section 4.1. We refer to Judd [1998] for other iterative methods such as Gauss-Jacobi and Gauss-Seidel algorithms.

Continuous State Spaces: For continuous state spaces, our method has to use numerical integration. We propose to use a deterministic numerical integration rule such as a Quasi-Monte Carlo rule to approximate the integral in (3.2). Let $\mathbb{M} := \{x_1, \cdots, x_M\}$ be the set of deterministic grid points used to approximate the integral. Note that implementing (3.2) on $\mathbb{M}$ (with the transition density normalized) implicitly solves the following equation:

$$(\mathcal{I}_M - \hat{\mathcal{T}}_M)(\mathcal{I}_M - \hat{\mathcal{T}}_M^T)y_M = u_M \tag{3.3}$$

where $\hat{\mathcal{T}}_M$ is the matrix whose (i,j)-th element is $\beta\tilde{f}(x_j|x_i)$, $\tilde{f}(x_i|x) := \frac{f(x_i|x)}{\sum_j f(x_j|x)}$ is the normalized transition density assuming the denominator is non-zero, u_M is an M-dimensional vector with $u_i = u(x_i)$ and $\mathcal{I}_M$ is an $M \times M$ identity matrix. Note that after solving (3.3) at iteration $\hat{k}_M$ for all $x \in \mathbb{M}$, we can interpolate $\widehat{V}^{ma}_{\hat{k}_M}(x)$ for $x \in \mathbb{X}\backslash\mathbb{M}$ using:

$$\widehat{V}^{ma}_{\hat{k}_M}(x) := u(x) + \beta \sum_{x_i \in \mathbb{M}} \tilde{f}(x_i|x)\widehat{V}^{ma}_{\hat{k}_M}(x_i) \tag{3.4}$$

where $\widehat{V}^{ma}_{\hat{k}_M}(x_i)$ for $x_i \in \mathbb{M}$ is the approximate solution to (3.3).

The continuous state-space algorithm is as follows:

Algorithm 2 (Model-adaptive Approach for Continuous State Spaces)**.**

- *Step 1:* For a given $f(x'|x)$ and $\mathbb{M} = \{x_1, \cdots, x_M\}$, generate the matrix $\hat{\mathcal{T}}_M$ whose (i,j)-th element is $\beta\tilde{f}(x_j|x_i)$ and u_M an M-dimensional vector with $u_i = u(x_i)$.
- *Step 2:* Generate the matrix $(\mathcal{I}_M - \hat{\mathcal{T}}_M)$ where $\mathcal{I}_M$ is an $M \times M$ identity matrix.
- *Step 3:* Generate the matrix $(\mathcal{I}_M - \hat{\mathcal{T}}_M^T)$.
- *Step 4:* For a given tolerance, iterate (3.2) until convergence.
- *Step 5:* Compute $\widehat{V}^{ma}_{\hat{k}_M}(x)$ for $x \in \mathbb{X}\backslash\mathbb{M}$ by (3.4).

3.2 Computational Cost

For discrete state spaces, the total computational cost of our method is $O(\hat{k}|\mathbb{X}|^2)$ where $\hat{k}$ is the number of iterations required for convergence and $O(|\mathbb{X}|^2)$ is the cost of matrix-vector multiplication. At each iteration, three matrix-vector multiplications are performed: two for $(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^T)y_k$ and one for $(\mathcal{I}-\mathcal{T}^T)s_k$. Due to the superlinear convergence, the number of iterations is expected to be small.

For continuous state spaces, the total computational cost is $O(\hat{k}_M M^2)$ where $\hat{k}_M$ can vary with M. As discussed before, there is a trade-off between simulation error and computational time. A large M leads to a small simulation error but a higher computational cost of solving the equation within the same tolerance. For different M, the cost of matrix-vector multiplication is determined by $O(M^2)$. The algorithm still converges superlinearly as shown in Theorem 5. Therefore, $\hat{k}_M$ is still expected to be small. Moreover, we will show that $\hat{k}_M$ is approximately the same for all sufficiently large M, which suggests that $\hat{k}_M$ is independent of M. Therefore, increasing M primarily affects the computational cost through matrix-vector multiplication rather than $\hat{k}_M$. This property offers a significant computational advantage as it primarily relies on matrix-vector multiplication that is amenable to GPU acceleration and fast matrix-vector multiplication methods mentioned in the introduction.

4 Theoretical Properties of Model-Adaptive Approach

This section discusses the theoretical properties of the model-adaptive approach. We will show the superlinear convergence of the approximation error. We compare our method with TD and SA. For continuous state spaces, we will consider the simulation error from the numerical integration and prove the number of iterations is approximately the same for all sufficiently large grids. We first impose the following regularity conditions:

Assumption 1. *For some positive constants $C_{\mu,1}$, $C_{\mu,2}$, C_u, assume:*

(i) $\mathbb{X}$ *is discrete or* $\mathbb{X} = [0,1]^d$.

(ii) *The Markov Chain $f(x'|x)$ has a unique stationary distribution μ. In the continuous state space case, this stationary measure is absolutely continuous with respect to Lebesgue measure.*

(iii) $\sup_x |u(x)| \leq C_u$.

(iv) *In the discrete state space case, $C_{\mu,1} \leq \mu(x) \leq C_{\mu,2}$ for $\forall\ x \in \mathbb{X}$. In the continuous state space case, $C_{\mu,1} \leq d\mu(x) \leq C_{\mu,2}$ for $\forall\ x \in \mathbb{X}$ where $d\mu$ is the density of μ.*

If $\mathbb{X} = [0,1]^d$, assume for a positive constant C_f:

(v) $\sup_{x',x} f(x'|x) \leq C_f$.

Under Assumption 1, $\mathcal{T}$ maps $L^2(\mathbb{X},\mu)$ to itself. Moreover, $\mathcal{T}$ is a β-contraction with respect to $\|\cdot\|_\mu$ (see for example Bertsekas [2015]). Consequently, (2.3) has a unique solution on $L^2(\mathbb{X},\mu)$. Assumption 1(iii) ensures that the solution is uniformly bounded by $\frac{C_u}{1-\beta}$. As discussed before, approximation solutions on different inner product spaces all converge superlinearly under regularity conditions. To achieve the fastest convergence rate of the approximation error when using CG, we propose to solve (3.1) on $L^2(\mathbb{X})$. Assumption 1(iv) and 1(v) are used to prove the existence and uniqueness of the solution to (2.3) on $L^2(\mathbb{X})$. Moreover, it μ-almost surely equals the solution on $L^2(\mathbb{X},\mu)$. The following theorem establishes the properties of the solution on $L^2(\mathbb{X})$:

Theorem 1. *Under Assumption 1, we have:*

(i) $(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^*)y = u$ *has a unique solution* y^* *on* $L^2(\mathbb{X})$.

(ii) $(\mathcal{I}-\mathcal{T}^*)y^* = V^*$, μ*-a.s., where* V^* *is the unique solution to* $(\mathcal{I}-\mathcal{T})V = u$ *on* $L^2(\mathbb{X},\mu)$.

Our approach first enjoys the following nice property:

Theorem 2. *(i) The sequence* $\{r_k\}_{k\geq 1}$ *generated by* (3.2) *is an orthogonal sequence.*

(ii) Under Assumption 1, the sequence $\{y_k\}_{k\geq 1}$ *generated by* (3.2) *is the optimal approximation in the following sense:*

$$y_k = \underset{y\in span\{r_0,r_1,\cdots,r_{k-1}\}}{\operatorname{argmin}} \|(\mathcal{I}-\mathcal{T}^*)y - V^*\|$$

Theorem 2 shows that $V_k^{ma} := (\mathcal{I}-\mathcal{T}^*)y_k$ minimizes the approximation error over $y \in \text{span}\{r_0, r_1, \cdots, r_{k-1}\}$. The orthogonality of the basis functions implies that the approximation error $\|V_k^{ma} - V^*\|$ decreases monotonically. As the residuals are informative about the solution, the projection onto the adaptive-sieve space can lead to a faster convergence rate of the approximation error than conventional methods.

Before establishing the convergence rate of our method, we define the concept of superlinear convergence. The concept of R-convergence, which is in analogy to the Cauchy root test for the convergence of series, quantifies the convergence rate of a sequence of approximation solutions:

Definition 3 (R-Convergence Ortega and Rheinboldt [2000])**.** *Let* $\{V_k\}_{k\geq 1}$ *be a sequence such that* $\lim_{k\to\infty}\|V_k - V^*\| = 0$. *Let* $R := \limsup_{k\to+\infty}\|V_k - V^*\|^{\frac{1}{k}}$ *be the root-convergence factors. The convergence is (i) superlinear for* $R = 0$, *(ii) sublinear for* $R = 1$, *and (iii) linear for* $0 < R < 1$.

Using this definition, the following theorem establishes the convergence rate of our method:

Theorem 3 (Convergence of Model-Adaptive Approach)**.** *Under Assumption 1, the sequence* $\{\|V_k^{ma} - V^*\|\}_{k\geq 1}$ *converges to zero monotonically and the sequence* $\{V_k^{ma}\}_{k\geq 1}$ *converges to* V^* *superlinearly. It also satisfies:*

$$\|V_k^{ma} - V^*\| = O((c_k)^k)$$

where, for some positive constants C_1*,* C_2*,* c_k *satisfies:*

$$\frac{C_1}{k} \leq c_k \leq \frac{C_2}{\sqrt{k}}.$$

Moreover, the sequence $\{\|r_k\|\}_{k\geq 1}$ *converges to zero superlinearly:*

$$\|r_k\| = O((c_k)^k)$$

In the continuous case, the rates can be improved if $f(x'|x)$ *has continuous partial derivatives of order up to* l*. In that case, there exists a constant* $C(l)$ *such that:*

$$c_k \leq \frac{C(l)}{k}$$

Finally, for discrete state spaces, the algorithm converges to V^* *in at most* $|\mathbb{X}|$*-steps.*

Theorem 3 establishes the superlinear convergence of the residual and the approximation error. It also shows the monotonic convergence of the approximation error. The decay rate of c_k is at most $\frac{1}{k}$ and at least $\frac{1}{\sqrt{k}}$. Those two bounds hold for all inner product spaces under regularity conditions, while the lower bound is achieved by solving the equation on $L^2(\mathbb{X})$ with the continuity assumption on the partial derivatives of the transition density. As $c_k \to 0$, it is straightforward to show that algorithm (3.2) will achieve the tolerance after a finite number of iterations. Therefore, our algorithm also selects the correct number of iterations (sieve dimension) itself.

4.1 Comparison with TD and SA

This section compares the convergence rate of the approximation error of our method with TD and SA. Denote V_k^{sa} and V_k^{td} the SA and TD approximation solutions.[4] To analyze

[4]For TD, see Tsitsiklis and Van Roy [1996] and Dann et al. [2014]. For SA, see Kress [2014].

the convergence properties of TD, we impose the following assumption on the projection bias:

Assumption 2. *There exist $C_{td,2} > C_{td,1} > 0$ such that for each k:*

$$C_{td,1}k^{-\frac{\alpha}{d}} \leq \|V^* - \Pi_{\mathbb{S}_k} V^*\|_\mu \leq C_{td,2}k^{-\frac{\alpha}{d}}$$

Assumption 2 imposes upper and lower bounds on the projection bias. The upper bound is standard in the literature. Similar lower bounds on function approximation by neural nets can be found in Yarotsky [2017] Lemma 3. The following theorem establishes the convergence rate of SA and TD:

Theorem 4. *(i) Under Assumption 1 and 2, the sequence $\{V_k^{td}\}_{k\geq 1}$ does not converge to V^* superlinearly.*

(ii) Under Assumption 1, the sequence $\{V_k^{sa}\}_{k\geq 1}$ converges to V^* linearly.

Theorem 4 shows that the SA method converges linearly and the TD method fails to converge superlinearly. It implies that to achieve a pre-specified tolerance, the number of iterations required for SA and the sieve dimension for TD can be much larger than for our method as it converges superlinearly.

4.2 Continuous State Space

This section shows that the number of iterations required to achieve a pre-specified tolerance is approximately the same for all sufficiently large grids. We also derive the simulation error caused by numerical integration in Algorithm 2. Note that the asymptotics in this section apply to the simulation error from numerical integration.

The use of numerical integration introduces an additional source of bias. While we can reduce the simulation error by increasing the number of grid points, this comes at the cost of solving a larger linear system. As discussed in Section 3.2, the computational cost of our method depends on the number of iterations and number of matrix-vector multiplication operations. Since the cost of the matrix-vector multiplication increases, we study the effect of the number of grid points on the number of iterations. We first quantify the simulation error. Define the α-Hölder ball with radius C as:

$$\mathcal{W}_C^\alpha([0,1]^d) := \left\{ f : [0,1]^d \mapsto \mathbb{R} \middle| \max_{|\boldsymbol{k}|<\alpha} \sup_x \left|D^{\boldsymbol{k}} f(x)\right| + \max_{|\boldsymbol{k}|=\lfloor\alpha\rfloor} \sup_{x\neq y} \frac{|D^{\boldsymbol{k}} f(x) - D^{\boldsymbol{k}} f(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \leq C \right\}$$

where we used the multi-index notation $\boldsymbol{k} := (k_1, \cdots, k_d) \in \mathbb{N}^d$, $|\boldsymbol{k}| := \sum_i k_i$, $D^{\boldsymbol{k}} f(x) := \frac{\partial^{|\boldsymbol{k}|} f(x)}{\partial x_1^{k_1} \cdots \partial x_d^{k_d}}$ and $\|x\|_\infty := \max_i |x_i|$. We review two results on the numerical integration:

Lemma 1 (Novak [2006]). *We have the following union bound on numerical integration:*

$$\inf_{\{x_i\}_{i \leq M}} \sup_{h \in \mathcal{W}_C^\alpha([0,1]^d)} \left| \frac{1}{M} \sum_{i=1}^{M} h(x_i) - \int h(x) dx \right| \asymp M^{-\frac{\alpha}{d}}$$

where the upper bound can be achieved by using a tensor product of the regular grid: $\{\frac{1}{2M}, \frac{3}{2M}, \cdots, \frac{2M-1}{2M}\}$.

Lemma 1 shows that the union bound suffers from the curse of dimensionality. However, in practical applications, the transition density often enjoys stronger smoothness. For example, autoregressive processes are often used to model the transition of state variables (e.g., Erdem et al. [2003], Hendel and Nevo [2006a], Aguirregabiria and Mira [2007], Aw et al. [2011], and Gowrisankaran and Rysman [2012]). The Koksma-Hlawka inequality allows us to bound the simulation error of Quasi-Monte Carlo methods by the discrepancy of the grids:

Lemma 2 (Koksma-Hlawka Inequality Hlawka [1961]). *Let $V_{HK}(h)$ be the total variation of h in the sense of Hardy–Krause, then:*

$$\left| \frac{1}{M} \sum_{i=1}^{M} h(x_i) - \int h(x) dx \right| \leq V_{HK}(h) D^*(x_1, \cdots, x_M)$$

where $D^*(x_1, \cdots, x_M)$ is the discrepancy of $\{x_1, \cdots, x_M\}$.

Lemma 2 shows that the upper bound on the simulation error is the product of two terms. $V_{HK}(h)$ measures the difficulty to integrate the function h. The smoother the function h is, the smaller the value of $V_{HK}(h)$. $D^*(x_1, \cdots, x_M)$ measures the quality of the points. For example, the discrepancy of Hammersley points is $D^*(x_1, \cdots, x_M) = O(\frac{(\log M)^{d-1}}{M})$ (see Hammersley and Handscomb [1964]). We introduce some notations before stating the next theorem. Let $1:d := \{1, 2, \cdots, d\}$. For a set $m \subset 1:d$, define $-m := 1:d \setminus m$. For $x \in [0,1]^d$, let $x_m : 1_{-m}$ be the point $z \in [0,1]^d$ with $z_j = x_j$ if $j \in m$ and $z_j = 1$ otherwise. Let the mixed partial derivative of $h(x_m, 1_{-m})$ taken once with respect to each x_j for $j \in m$ be denoted as $D^m h(x_m : 1_{-m})$. We define the following function class:

Definition 4. *Let $\mathcal{HK}_C := \{h | D^m h(x_m, 1_{-m})$ is continuous for $\forall\ m \subset 1:d, V_{HK}(h) \leq C\}$.*

Assumption 3. *If the low-discrepancy grid is used, for a positive constant C assume:*

(i) $u \in \mathcal{HK}_C$ *and* $f(x'|\cdot) \in \mathcal{HK}_C$ *for* $\forall\ x'$.

(ii) $f(\cdot|x) \in \mathcal{HK}_C$ *for* $\forall\ x$.

If the regular grid is used, for a positive constant C *assume:*

(iii) $u \in \mathcal{W}_C^\alpha([0,1]^d)$ *and* $f(x'|\cdot) \in \mathcal{W}_C^\alpha([0,1]^d)$ *for* $\forall\ x'$.

(iv) $f(\cdot|x) \in \mathcal{W}_C^\alpha([0,1]^d)$ *for* $\forall\ x$.

Assumption 3 imposes smoothness conditions on the utility function and the transition density. Assumption 3(i) and 3(iii) are used to prove that the solution V^* also belongs to $\mathcal{HK}_C$, or $\mathcal{W}_C^\alpha([0,1]^d)$ where the constant C may change. Assumption 3(ii) and 3(iv) are then used to control the simulation error from the numerical integration. The following theorem establishes both the simulation error and the approximation error:

Theorem 5. *Suppose Assumption 1 and 3 hold. Let* $\{1-\lambda_{j,M}\}_{j\leq M}$ *be the ordered eigenvalue of* $(\mathcal{I}_M - \hat{\mathcal{T}}_M)(\mathcal{I}_M - \hat{\mathcal{T}}_M^T)$, $\Delta_M := \max_j\{1-\lambda_{j,M}\}$, *and* $\delta_M := \min_j\{1-\lambda_{j,M}\} > 0$.[5] *Then,* $\{\hat{V}_k^{ma}\}_{k\geq 1}$ *generated by Algorithm 2 satisfies:*

- *If the low-discrepancy grid is used, then:*

$$\|\hat{V}_k^{ma} - V^*\| = O(\underbrace{\frac{(\log M)^{d-1}}{M}}_{\textit{Simulation Error}} + \underbrace{(c_{1,k,M})^k}_{\textit{Approximation Error}})$$

- *If the regular grid is used, then:*

$$\|\hat{V}_k^{ma} - V^*\| = O(\underbrace{M^{-\frac{\alpha}{d}}}_{\textit{Simulation Error}} + \underbrace{(c_{2,k,M})^k}_{\textit{Approximation Error}})$$

where $c_{1,k,M} \leq \frac{C_{1,M}}{k}$ *and* $c_{2,k,M} \leq \frac{C_{2,M}}{k}$ *for some positive constants* $C_{1,M}$ *and* $C_{2,M}$.

Theorem 5 consists of two parts. The simulation error arises from replacing $\mathcal{T}$ by $\hat{\mathcal{T}}_M$. The approximation error upper bound also depends on $\hat{\mathcal{T}}_M$. However, as $c_{1,k,M}$ and $c_{2,k,M}$ converge to zero at the rate of $\frac{1}{k}$, we can expect that the number of iterations will still be small. To compare $c_{1,k,M}$ and $c_{2,k,M}$ with c_k, we impose the following assumption used in Atkinson [1975]:

Assumption 4. *Let* $\hat{\mathcal{K}}_M$ *be the numerical integral operator defined as* $\hat{\mathcal{K}}_M y(x) := \hat{\mathcal{T}}y(x) + \hat{\mathcal{T}}^*y(x) - \hat{\mathcal{T}}\hat{\mathcal{T}}^*y(x)$, *where* $\hat{\mathcal{T}}y(x) := \beta\sum_{i=1}^M \tilde{f}(x_i|x)y(x_i)$, $\hat{\mathcal{T}}^*y(x) := \beta\sum_{i=1}^M \tilde{f}^*(x_i|x)y(x_i)$, *and* $\hat{\mathcal{T}}\hat{\mathcal{T}}^*y(x) := \beta\sum_{i=1}^M \tilde{f}(x_i|x)\hat{\mathcal{T}}^*y(x_i)$. *Assume:*

[5] We have $\delta_M > 0$ as $(\mathcal{I}_M - \hat{\mathcal{T}}_M)(\mathcal{I}_M - \hat{\mathcal{T}}_M^T)$ is positive definite.

(i) $\hat{\mathcal{K}}_M$ *maps* $L^2(\mathbb{X})$ *to itself for* $M \geq 1$.

(ii) *The sequence of operators* $\{\hat{\mathcal{K}}_M\}_{M\geq 1}$ *is collectively compact, i.e.,* $\{\hat{\mathcal{K}}_M y | M \geq 1, \|y\| \leq 1$ *and* $y \in L^2(\mathbb{X})\}$ *has compact closure in* $L^2(\mathbb{X})$.

Recall we aim to solve the equation within a given tolerance, and Theorem 3 guarantees it can be achieved within a finite number of iterations. Therefore, we impose the following assumption for a finite upper bound on the number of iterations:

Assumption 5. *Let* p *be any given positive integer and* $\{\phi_{i,k}\}_{i\leq J_k}$ *be a basis for null*$(\lambda_k \mathcal{I} - \mathcal{K})$ *where* λ_k *is of multiplicity* J_k. *For all* $k \leq p$, $i \leq J_k$, *assume for some positive constant* C:

(i) *If the low-discrepancy grid is used,* $\phi_{i,k} \in \mathcal{HK}_C$.

(ii) *If the regular grid is used,* $\phi_{i,k} \in \mathcal{W}_C^{\alpha}([0,1]^d)$.

Theorem 6. *Let* p *be any given positive integer. Under Assumption 1, 4 and 5, for sufficiently large* M *and any* $k \leq p$, *we have:*

- *If the low-discrepancy grid is used, then:*

$$(c_{1,k,M})^k = O(\frac{(\log M)^{d-1}}{M} + (c_k)^k)$$

- *If the regular grid is used, then:*

$$(c_{2,k,M})^k = O(M^{-\frac{\alpha}{d}} + (c_k)^k)$$

Theorem 6 implies that the needed iterations $\hat{k}_M$ to solve the equation within a given tolerance will be approximately the same for all sufficiently large M. Moreover, it is also approximately the same as the number of iterations to solve the original integral equation within a given tolerance. This phenomenon is called the mesh independence principle (see Atkinson [1997]), and has also been observed experimentally in Flores [1993]. Let $p := k^*$ where k^* is the needed iterations for convergence of the original equation. Then, $\hat{k}_M$ will be approximately the same as k^* for all sufficiently large M. In other words, the algorithm itself will choose the correct number of iterations (sieve dimension), which is expected to be independent of M. Combining Theorem 5 and 6 gives the following corollary:

Corollary 1. *Let* p *be any given positive integer. Under Assumption 1, 3, 4 and 5, for sufficiently large* M *and any* $k \leq p$, *we have:*

- *If the low-discrepancy grid is used, then:*

$$\|\hat{V}_k^{ma} - V^*\| = O(\underbrace{\frac{(\log M)^{d-1}}{M}}_{Simulation\ Error} + \underbrace{(c_k)^k}_{Approximation\ Error})$$

- *If the regular grid is used, then:*

$$\|\hat{V}_k^{ma} - V^*\| = O(\underbrace{M^{-\frac{\alpha}{d}}}_{Simulation\ Error} + \underbrace{(c_k)^k}_{Approximation\ Error})$$

5 Monte Carlo Simulations

This section presents three Monte Carlo simulations. First, we simulate a bus engine replacement model to visualize the convergence behavior of our method. Second, we analyze a model of consumer demand for storable goods similar to Hendel and Nevo [2006a]. We compare the computational time of five solution algorithms. The substantial performance gains from our method show that our method opens the door to the use of Bayesian MCMC estimators for such models. Finally, we examine a single-firm entry and exit problem described in Aguirregabiria and Magesan [2023]. We show that our method can improve the computational efficiency of policy iteration. We compare our approach (combination of the model-adaptive algorithm with Nested Pseudo-Likelihood (NPL)) with the Full-Solution with Euler Equation (FSEE) method from Aguirregabiria and Magesan [2023]. We also compare the performance of our approach against: SA and TD.

5.1 Bus Engine Replacement

This section simulates a well-known bus engine replacement problem to visualize the convergence behavior of our method. We adapt the setting in Arcidiacono and Miller [2011].

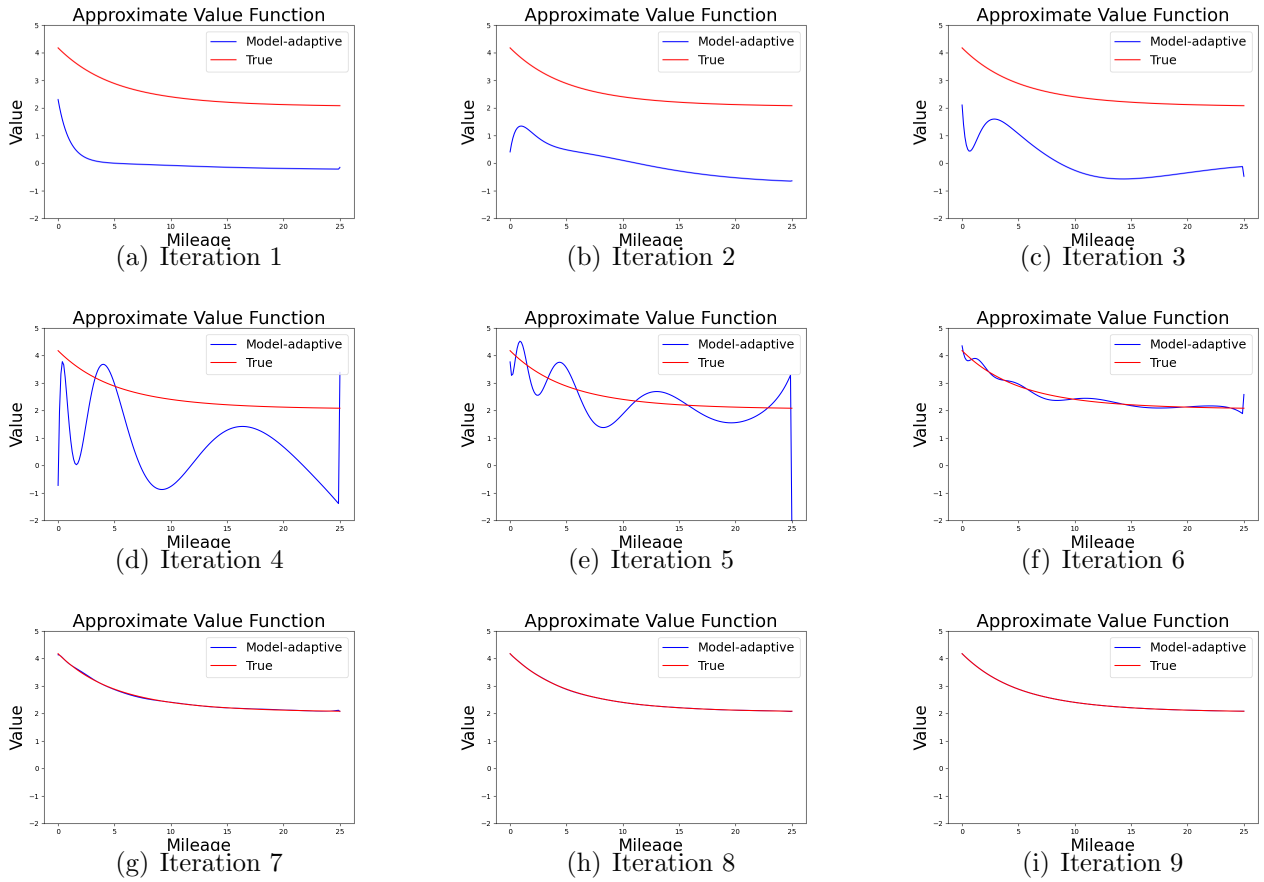
At each period $t \leq \infty$, an agent chooses to maintain $a_t = 1$ or replace $a_t = 0$ the engine. The replacement cost is RC. The maintenance cost is linear in mileage with accumulated mileage up to 25, i.e., $u(x_t, 1) = \theta_1 \min\{x_t, 25\}$, where x_t is the mileage of the engine. Moreover, mileage accumulates in increments of 0.125. The period utility of the agent is $u(x_t, a_t, \varepsilon_t) = (1 - a_t)(RC + \varepsilon_t(0)) + a_t(\theta_1 \min\{x_t, 25\} + \varepsilon_t(1))$ where $(\varepsilon_t(0), \varepsilon_t(1))$ are i.i.d

extreme value type I distributed shocks. The transition probability of x_t is specified as:

$$f(x_{t+1}|x_t, a_t) = \begin{cases} \exp(-\theta_2 x_{t+1}) - \exp(-\theta_2(x_{t+1} + 0.125)) & \text{if } a_t = 0, 0 \leq x_{t+1} \leq 25 \\ \exp(-\theta_2 x_{t+1}) & \text{if } a_t = 0, x_{t+1} = 25 \\ \exp(-\theta_2(x_{t+1} - x_t)) - \exp(-\theta_2(x_{t+1} + 0.125 - x_t)) & \text{if } a_t = 1, x_t \leq x_{t+1} \leq 25 \\ \exp(-\theta_2(25 - x_t)) & \text{if } a_t = 1, x_{t+1} = 25 \end{cases}$$

where we set $\theta_1 = -0.15, \theta_2 = 1, RC = -2$, and the discount factor $\beta = 0.9$. To visualize, we solve the DP problem and use the true CCPs to construct the linear systems.

Our method takes 15 iterations to solve the equation. Figure 1 visualizes our approximation solution for the first 9 iterations. Each panel shows the impact on the approximate solution of adding one additional sieve basis function to the previous value. Figure 9 in Online Appendix B.1 plots the basis functions at each iteration.



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

(d) Iteration 4 (e) Iteration 5 (f) Iteration 6

(g) Iteration 7 (h) Iteration 8 (i) Iteration 9

Figure 1: Convergence Behavior of Model-Adaptive Approach

Figure 2 plots the L_2-norm of the residuals and the sup-norm of the approximation error.

The approximation solution is close to the true solution after iteration 6, which implies that our algorithm constructs a good sieve space using 6 iterations. After that, the approximation converges to the true solution rapidly. This finding aligns with Figure 2 as the approximation error dramatically decreases at iteration 6-7. Other norms also decay dramatically after 6 iterations. Moreover, the L_2-norm of the approximation errors decreases monotonically, consistent with Theorem 3.

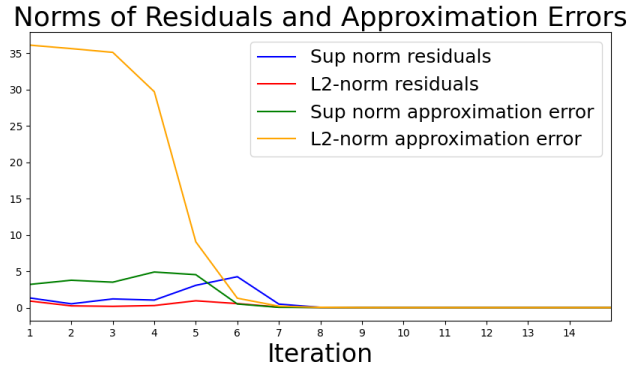


Figure 2: Norms of residuals and approximation error for Bus Engine Replacement Model

5.2 Consumer Demand for Storable Goods

In this section, we analyze a model of consumer demand for storable goods similar to Hendel and Nevo [2006a]. We aim to show that our method can substantially improve computational efficiency and open the door to use of Bayesian MCMC estimation methods Chernozhukov and Hong [2003].

At each period, given prices $\boldsymbol{p}$ and inventories I, a household of size n decides which brand k to purchase, how much to purchase j and how much to consume C.[6] Let $\omega_j(\boldsymbol{p})$ be the indirect utility from brand choice. Given the assumptions in Hendel and Nevo [2006a], brand choice is purely a static problem and the consumer's value function $V(x)$ for the dynamic problem is the solution to:

$$V(x) = \log \sum_j \exp(U(C(x,j), I, j; \theta) + \omega_j(\boldsymbol{p}) + \beta \mathbb{E}[V(x')|x, C(x,j), j])$$

where $C(x,j) := \text{argmax}_{c_{min} \leq c \leq c_{max}} \Big[U(c, I, j; \theta) + \omega_j(\boldsymbol{p}) + \beta \mathbb{E}[V(x')|x, c, j] \Big]$ and $x := \{\boldsymbol{\omega}, I\}$.

[6]Further details of the model, are presented in Section 6. For each household size, parameter values are set equal to the estimated values reported in Section 6

We compare five solution algorithms[7]. First, we solve the DP problem by policy iteration using our model-adaptive approach. Specifically, for a given consumption function, we update the value function by policy iteration with our method. Then, we update the consumption function and iterate until convergence. The second and third algorithms solve the linear system by SA and by using an exact linear equation solver[8], respectively. The fourth algorithm updates the value function by value iteration for a given consumption function. The fifth algorithm updates the value function by one-step value iteration given a consumption function. Then, it iterates over both the value function and the consumption function until convergence. The one-step updating was used in Osborne [2018] to implement the Bayesian estimator proposed by Imai et al. [2009]—henceforth, IJC. At each MCMC step, rather than iterate to convergence, the algorithm updates both the value function and the consumption function only by one-step.

Table 1 compares the computational time and number of iterations required by each of the five solution algorithms across simulations for different household sizes. The first algorithm, which uses our model-adaptive approach with policy iteration, requires approximately 2.5 minutes across all household sizes. In comparison, the second algorithm requires around 3.7 minutes, about 1.5 times longer than ours. Exact solution takes about 6.6 minutes, roughly 2.6 times longer than ours. These results demonstrate that our model-adaptive approach substantially improves the computational efficiency of policy iteration. Finally, our method is at least 100 times faster than both value iteration and one-step value iteration.

Table 1: Performance of Solution Algorithms

	Time in mins					Number of Iterations				
Household Size	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Model-Adaptive	2.4	2.6	2.4	2.5	2.4	6	7	6	6	6
Successive Approx.	3.6	3.9	3.6	3.8	3.7	6	7	6	6	6
Exact Solution	6.6	6.1	6.2	6.6	7.3	6	7	6	6	6
Value Iteration	316.4	372.0	339.1	304.9	327.8	6	7	6	6	6
One-step Value Iteration	260.2	261.2	262.1	263.8	348.9	1621	1639	1649	1650	1646

Note: The computational time is the real time in minutes. The code runs on an Intel Xeon Gold 6240 CPU (2.60GHz) with 192GB RAM.

Our method opens the door to use of Bayesian MCMC estimators. To simulate 10,000 MCMC steps, Table 1 and Table 9 suggest that our method would require around 7 days

[7]For details of the algorithms, see Online Appendix B.2. For the first three algorithms, the stopping rules are $\|p^{i+1} - p^i\|_\infty \leq 10^{-4}$, $\|r_k\|_\infty \leq 10^{-8}$, and $\|C^{i+1} - C^i\|_\infty = 0$. For the last two algorithms, the stopping rules are $\|V^{i+1}(x) - V^i(x)\|_\infty \leq 10^{-8}$, and $\|C^{i+1} - C^i\|_\infty = 0$.

[8]We use scipy.sparse.linalg.spsolve.

while SA and exact solution would require around 11 and 18 days, respectively.[9]

An alternative estimator to the Bayesian MCMC estimator is the IJC approach. However, the performance of the one-step value iteration suggests that the IJC approach may not be the most suitable estimator for this problem. While the IJC cost-per-iteration is about 0.33 minutes[10] (compared with 1 minute for our approach), in Table 1, the IJC algorithm requires more than 1600 iterations to converge even when the true parameters are known. Consequently, due to the extremely large number of iterations required, we expect the total computational time using the IJC approach would be substantially higher than our proposed approach.

5.3 Single Firm Entry and Exit

This section examines a single-firm entry and exit problem described in Aguirregabiria and Magesan [2023]. Our analysis focuses on comparing computational time and number of iterations for different number of grid points for numerical integration and discount factors. We also compare computational time of the NPL-algorithm with our method and the Full-Solution with Euler Equation (FSEE). Finally, we compare our model-adaptive approach against successive approximation and temporal difference in terms of computational time and approximation error in solving the linear equation.

5.3.1 Design of the Simulation

At each period $t \leq +\infty$, a firm decides whether to exit ($a_t = 0$) or enter ($a_t = 1$) the market. For an active firm, the profit is $\pi(1, x_t)+\varepsilon_t(1)$. $\pi(1, x_t)$ equals the variable profit VP_t minus fixed cost FC_t, and minus entry cost EC_t. For an inactive firm, $\pi(0, x_t)$ is normalized to be 0, and the profit is $\varepsilon_t(0)$. The variable profit is $VP_t = (\theta_0^{VP} + \theta_1^{VP} z_{1t} + \theta_2^{VP} z_{2t}) \exp(w_t)$ where w_t is the productivity shock, z_{1t} and z_{2t} are exogenous state variables that affect price-cost margin. The fixed cost is $FC_t = \theta_0^{FC} + \theta_1^{FC} z_{3t}$, and the entry cost is $EC_t = (1 - a_{t-1})(\theta_0^{EC} + \theta_1^{EC} z_{4t})$ where $(1 - a_{t-1})$ indicates that the entry cost is paid if the firm is inactive at the previous period ($a_{t-1} = 0$). Continuous state variables follow AR(1) process: $z_{jt} = 0.6 z_{jt-1} + \epsilon_{jt}$, $w_t = 0.2 + 0.6 w_{t-1} + \epsilon_t$, where ϵ_{jt}, ϵ_t follows i.i.d standard normal. The true parameters θ^* are chosen to be $\theta_0^{VP} = 0.5, \theta_1^{VP} = 1, \theta_2^{VP} = -1, \theta_0^{FC} = 1.5, \theta_1^{FC} = 1, \theta_0^{EC} = 1, \theta_1^{EC} = 1$. For each of 5 continuous state variables, we use Tauchen's method

[9]For the simulation, the initial guess is zero, while each MCMC step uses the previous value function as its initial guess, thereby reducing the computational time to approximately one minute per iteration.

[10]Note that each MCMC step in the IJC approach requires two iterations to update the value function and consumption function of both the current and candidate parameters.

(Tauchen [1986]) to discretize the exogenous state variables to obtain the transition matrix for each exogenous state variable with M-point support where $M = \{7, 8, 9, 10\}$. Moreover, we set the discount factor $\beta = \{0.95, 0.975, 0.980, 0.985, 0.990, 0.995, 0.999\}$.

5.3.2 Computational Time and Number of Iterations

We first compare the performance of our method for different M and β. We use policy iteration[11] with our approach[12] to solve the dynamic programming problem. The policy iteration converged after 4 iterations for all M and β. Note that the transition matrix is a dense matrix. Table 2 presents the average number of iterations and average computational time across 4 policy iteration steps required by our model-adaptive method. These results provide empirical support for Theorem 6. As M increases, the number of iterations remains relatively stable for all β. The computational time scales approximately as $O(|\mathcal{X}_M|^2)$, corresponding to the matrix size. Additionally, there is a slight increase in the number of iterations (from around 40 to 50) as β increases. As it only takes around 7.5 minutes to solve the linear equation with $|\mathcal{X}_M| = 200,000$, our method can be used to improve the computational efficiency of policy iteration where the main computational cost is solving the linear system of equations.

Table 2: Monte Carlo Simulation: Performance for Different β and M

	Avg Number of Iterations				Avg Time in secs			
β	$M=7$	$M=8$	$M=9$	$M=10$	$M=7$	$M=8$	$M=9$	$M=10$
0.950	39	38	37	37	12.1	44.6	137.4	392.6
0.975	42	41	40	39	13.5	48.2	149.5	426.5
0.980	42	41	40	40	13.3	47.8	149.6	425.1
0.985	43	42	41	40	13.9	49.5	152.8	432.8
0.990	44	43	42	41	13.7	50.5	158.0	443.1
0.995	45	45	44	43	14.1	52.4	166.5	456.2
0.999	51	50	49	49	15.9	59.5	185.5	522.9
$\lvert\mathcal{X}_M\rvert$	33,614	65,536	118,098	200,000	33,614	65,536	118,098	200,000

Note: $|\mathcal{X}_M| := 2 \times M^5$ is the cardinality of the state space. All dynamic programming problems converge after 4 policy iterations. Avg number of iterations and avg time refer to the average number of iterations and time of the model-adaptive approach to solve the linear system of equations of the policy valuation operator. The computational time is the real time in seconds. The code runs on an Intel Xeon Gold 6240 CPU (2.60GHz) with 1.5TB RAM.

[11] An alternative approach to solve the DP by the Euler-Equation method Aguirregabiria and Magesan [2023]. However, it works for models where the only endogenous state variable is the previous action (see their Definition 1). This feature is satisfied in the simulation, though it is restrictive in general.

[12] The stopping rules are $\|p_{k+1} - p_k\|_\infty \leq 10^{-5}$, and $\|r_k\|_\infty \leq 10^{-7}$. The initial guess are $p_0 = \frac{1}{2}$, and $y_0 = 0$.

5.3.3 Comparison between NPL Algorithm and Full-solution Estimator

This section compares the computational time of NPL-algorithm with our approach and the FSEE method. FSEE requires solving the DP problem for each candidate parameter guess during the optimization of the likelihood. In contrast, NPL-algorithm only needs to solve the linear system once if the utility function is linear in parameters. Specifically, to solve:

$$V(x) = \sum_a p(a|x) \left[\boldsymbol{g}(x,a)^T \theta + \kappa - \log p(a|x) + \beta \mathbb{E}_{x'|x,a} V(x') \right] \tag{5.1}$$

where $\boldsymbol{g}(x,a)$ is the vector of features and θ is the utility parameter, we can separately solve the following $(d_\theta + 1)$ equations:

$$V_i(x) = \sum_a p(a|x) \left[\boldsymbol{g}_i(x,a) + \beta \mathbb{E}_{x'|x,a} V_i(x') \right]$$
$$V_p(x) = \sum_a p(a|x) \left[\kappa - \log p(a|x) + \beta \mathbb{E}_{x'|x,a} V_p(x') \right]$$

where $\boldsymbol{g}_i(x,a)$ is the i-th element of $\boldsymbol{g}(x,a)$, and set $V(x) = \boldsymbol{V}(x)^T \theta + V_p(x)$ to solve (5.1) where $\boldsymbol{V}(x) := (V_1(x), \ldots, V_{d_\theta}(x))$.

The NPL-algorithm requires solving $(d_\theta+1)$ linear systems at each NPL iteration. Therefore, the total computational time is $O(k_{NPL}(d_\theta + 1)\bar{k}|\mathcal{X}|^2)$ where k_{NPL} is the number of NPL iterations, $\bar{k}$ is the average number of iterations to solve the linear system of fixed point equations, and $|\mathcal{X}|$ is the cardinality of the state space. The FSEE requires solving the DP problem for each candidate parameter guess during the optimization of the likelihood. The computational time is $O(k_{EE}\bar{k}_{EE}(|\mathbb{A}| - 1)|\mathcal{X}|^2)$ where k_{EE} is the number of iterations for optimization and $\bar{k}_{EE}$ is the average number of EE iterations.

Table 3 compares the computational time for NPL-algorithm with our model-adaptive approach and FSEE[13] for all β and $M \in \{5, 6, 7\}$. All NPL-algorithm converged after 3 iterations. 1-NPL-algorithm is about two times faster than the FSEE. On average, 2-NPL-algorithm is as fast as the FSEE. Furthermore, the computational time can be reduced by parallelizing these $(d_\theta + 1)$ linear equations. Therefore, for large state spaces where the full-solution method is computationally infeasible, the CCP estimator (1-NPL-algorithm) may still be applied.

[13]For NPL-algorithm, we estimate CCPs by a flexible logit model.The convergence criteria are $\|p^{i+1} - p^i\|_\infty \leq 10^{-4}$ for NPL-algorithm, $\|r_k\|_\infty \leq 10^{-7}$ for model-adaptive approach, and $\|\tilde{v}^{i+1} - \tilde{v}^i\|_\infty \leq 10^{-7}$ for EE method where $\tilde{v}$ is the conditional value function difference, i.e., $\tilde{v}(x,a) := v(x,a) - v(x,0)$.

Table 3: Comparison: NPL and Full-solution Estimator

	1-NPL Time in secs			2-NPL Time in secs			3-NPL Time in secs			EE Time in secs		
β	$M=5$	$M=6$	$M=7$	$M=5$	$M=6$	$M=7$	$M=5$	$M=6$	$M=7$	$M=5$	$M=6$	$M=7$
0.950	11.7	41.4	175.9	19.7	75.9	346.7	27.7	111.0	515.8	19.6	96.2	382.4
0.975	13.2	45.0	189.1	22.6	85.8	368.0	31.7	127.0	544.7	19.8	99.6	377.0
0.980	13.1	44.3	186.4	22.3	84.9	366.9	30.9	126.0	546.2	18.5	98.0	377.0
0.985	13.4	46.1	193.4	22.7	87.7	377.0	31.9	129.4	560.3	18.4	99.5	378.9
0.990	13.7	47.3	198.5	23.5	90.3	386.4	32.7	133.0	575.0	18.2	96.8	376.7
0.995	13.7	48.8	204.1	23.7	93.6	400.8	33.4	137.9	595.9	18.0	99.3	376.7
0.999	14.5	52.0	219.8	25.3	101.8	437.5	35.7	151.2	654.7	18.3	99.0	379.1
$\lvert\mathcal{X}_M\rvert$	6,250	15,552	33,614	6,250	15,552	33,614	6,250	15,552	33,614	6,250	15,552	33,614

Note: $|\mathcal{X}_M| := 2 \times M^5$ is the cardinality of the state space. All NPL algorithms converge after 3 iterations. The max bias defined by $\|\hat{\theta} - \theta\|_2^2$ for all estimators are 5×10^{-5} across β and M. The time for NPL is the total time of the NPL-algorithm. The code runs on an i9-13980HX CPU and 64GB RAM..

5.3.4 Comparison with SA and TD

This section compares our method with TD and SA for $M = 5$ in terms of computational time and approximation error. We solve the DP problem and use the true CCPs to construct the linear equations. For the model-adaptive approach and SA, the convergence criterion and the initial guess are the same as before. For TD, we use polynomials of degree k with an intercept and the second-order interaction terms. Our method and SA are guaranteed to converge. Therefore, we only report the computational time. To evaluate the approximation error of TD for a fixed k, we compute $\|(\mathcal{I}_M - \hat{\mathcal{T}}_M)V_k^{td} - u_M\|_\infty$ where V_k^{td} is the TD approximation solution. Then, we report minimum norm of residuals for degree $k \leq 10$ and the average computational time across $k \leq 10$.

Table 4 shows that in terms of the number of iterations and computational time, our method outperforms SA for all β. Moreover, as β increases, the number of iterations of our method only slightly increases, while that of SA increases significantly. This observation can be explained by the approximation error upper bound $O(\beta^k)$ of SA. In contrast, the approximation error upper bound of our method is $O((\frac{C_2}{k})^k)$ where only C_2 depends on β. For the norm of residuals, both methods outperform TD. This arises from the ad-hoc sieve space of TD that may not be good enough to approximate the value function.

In Figures 3 to 6, we visualize the convergence of our method. We plot the sup-norm of residuals, $\|r_k\|_\infty$, L_2-norm of the residuals, $\|r_k\|_2$, the sup-norm of the approximation error, $\|V_k - V^*\|_\infty$, and L_2 norm of the approximation error, $\|V_k - V^*\|_2$. All figures are plotted as functions of the iteration count k for different β. Figures 3 to 5 exhibit a similar pattern: they increase initially, peak between iterations 20-30, and then decrease rapidly.

Table 4: Monte Carlo Simulation: Performance Comparison

β	**Model-Adaptive**		**Successive Approx.**		**Temporal Diff.**	
	# Iter	**Time (s)**	**# Iter**	**Time (s)**	**Min Res**	**Time (s)**
0.950	50	0.35	349	0.74	110.06	0.05
0.975	53	0.29	706	1.28	110.75	0.04
0.980	54	0.31	885	1.86	110.89	0.04
0.985	54	0.29	1,182	2.08	111.03	0.05
0.990	57	0.31	1,777	3.42	104.21	0.04
0.995	58	0.31	3,561	6.89	98.21	0.04
0.999	63	0.35	17,833	33.96	93.96	0.04

Note: For TD, Min Res reports $\min_{k\leq 10} \|(\mathcal{I}_M - \hat{\mathcal{T}}_M)V_k^{\text{td}} - u_M\|_\infty$ where $M = 5$. The code runs on an Intel Xeon Gold 6240 CPU (2.60GHz) with 1.5TB RAM.

Figure 6 shows the L_2-norm of approximation error decreases monotonically consistent with Theorem 3. Notably, while $\beta = 0.999$ initially results in larger residuals and approximation errors compared to smaller discount factors, it still achieves convergence after approximately 30 iterations. This demonstrates the ability of our method to handle large discount factors, which is challenging for SA.

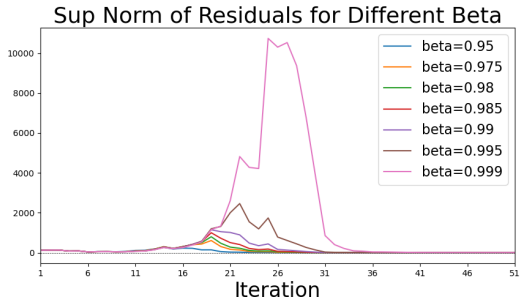


Figure 3: Sup-norm of Residuals

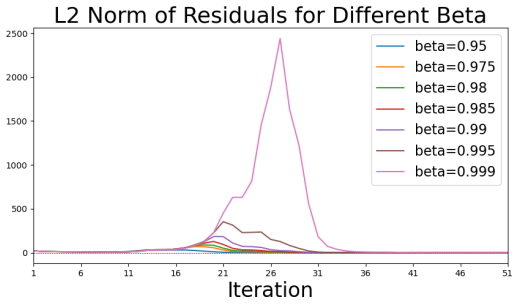


Figure 4: L_2-Norm of Residuals

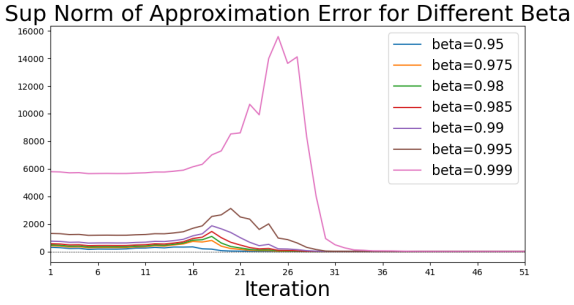


Figure 5: Sup-norm of Approximation Error

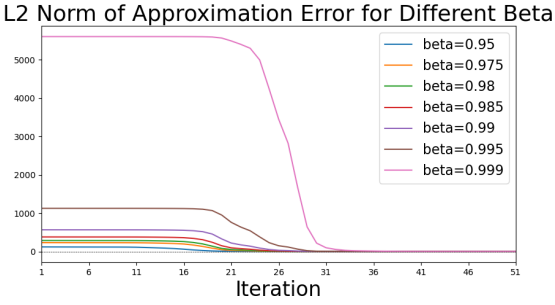


Figure 6: L_2-Norm of Approximation Error

6 Application to Consumer Demand for Storable Goods

This section applies our model-adaptive approach to a dynamic consumer demand model for laundry detergent using Kantar's Worldpanel Take Home data. We first describe the data, model, and estimation procedure. We also discuss the implication of the results.

6.1 Data

The analysis of the Great Britain laundry detergent industry is based on the data from 1st January 2017 until 31st December 2019. It captures detailed information on a representative sample of British households' purchase of fast moving products, including food, drink, and laundry detergents. The data has been used in previous studies such as Dubois et al. [2014, 2020]. Households use barcode scanners to record all their grocery purchases. For each purchase, the data includes key information such as price, quantity, product characteristics, and purchase date.

We consider the market for laundry detergent. A laundry detergent product is defined by its quantity, brand, and chemical properties (bio/non-bio). Laundry detergents are available in various formats such as liquid, powder, and gel, each with different dosage metrics. To standardize quantity across formats, we define the quantity purchased by the number of washes. Table 5 presents the top 10 brands and bio/non-bio combinations, which account for 76.82% of the total observed purchases. We restrict our analysis to these top 10 brand and bio/non-bio combinations, and assume they are available to all consumers.

Table 5: Market Share of Top 10 Brands and Bio/Non-Bio

Brand + Bio	1	2	3	4	5	6	7	8	9	10
# of Purchases	41,427	40,954	33,164	32,604	29,213	26,087	21,768	17,371	17,334	15,945
Cumulative Share (%)	11.54	22.94	32.18	41.26	49.39	56.65	62.72	67.55	72.38	76.82

Note: Number of Purchases is the total number of purchases observed for each brand and bio/non-bio combination between 1st January 2017 and 31st December 2019, using Kantar's Worldpanel Take Home data.

In Figure 7 we show the histogram of quantities purchased[14], which suggests three natural clusters corresponding to small, medium, and large sizes. Therefore, we use 3-means clustering to aggregate the quantities for each brand and bio/non-bio combination into small (23), medium (40), and large (64) sizes. Table 6 summarizes the number of observations for each cluster. We then calculate the weekly average transaction price for each product cluster, and use it as the price index in our model.

[14]We restricted our analysis to products with number of washes between 10 and 100.

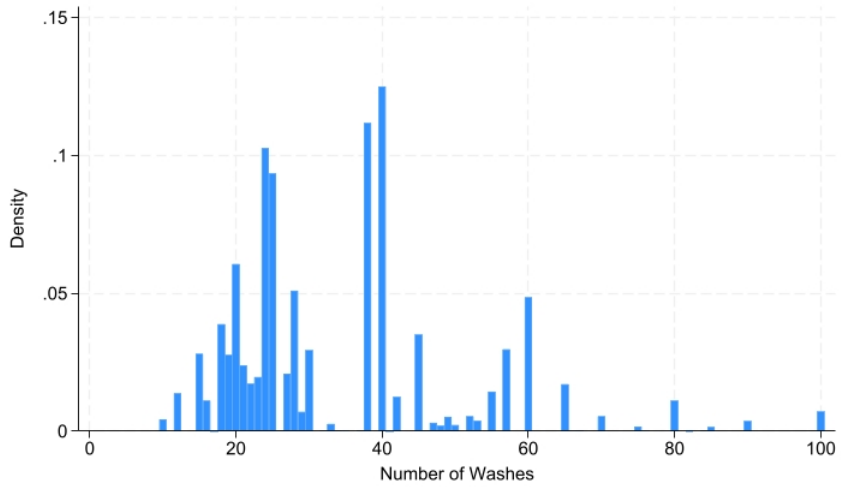


Figure 7: Histogram of Quantities Purchased

Table 6: 3-means Clustering of Quantities Purchased

	Brand + Bio									
Number of Washes	1	2	3	4	5	6	7	8	9	10
23	20,928	24,946	20,066	28,564	12,134	6,370	5,832	15,554	10,896	6,280
40	13,265	11,938	9,370	3,789	10,075	11,802	9,634	1,720	5,052	7,783
64	7,234	4,070	3,728	0	7,004	7,915	6,302	0	1,386	1,882

Note: Number of observations for 3-means clustering of quantities purchased for each brand and bio/non-bio combination between 1st January 2017 and 31st December 2019, using Kantar's Worldpanel Take Home data.

In Table 7 we describe purchase statistics by household size. We restrict our analysis to households who purchase laundry detergent at least 3 times and at most 15 times each year for household sizes 1-4, and 3 to 20 times each year for household size 5. The average size of laundry detergent purchased increases with household size, ranging from 32.2 washes per purchase for single-person households to 38.1 washes for households with four or five. The average price per wash, conditional on purchase, shows a slight decreasing trend as household size increases, from £0.152 for single-person households to £0.142 for the largest households. The average number of weeks between purchases decreases as household size increases, with single-person households making purchases every 9.7 weeks on average, while households with five members purchase every 7.4 weeks. The average quantity of laundry detergent purchased per year increases with household size, ranging from 165 washes per year for single-person households to 256 washes for the largest households. Table 7 indicates heterogeneity in purchasing behavior across different household sizes. Therefore, we separately estimate the model for each household size.

Table 7: Laundry Detergent Purchase Statistics by Household Size

Household Size	**1**	**2**	**3**	**4**	**5**
Avg. Size conditional on purchase	32.2	35.2	36.6	38.1	38.1
Avg. Price per wash conditional on purchase	0.152	0.146	0.145	0.143	0.142
Avg. Weeks between purchases	9.7	8.6	7.8	7.6	7.4
Avg. Quantity purchased per Year	165	203	232	247	256
Number of households	235	772	351	373	117

Note: The purchase statistics are based on Kantar's Worldpanel Take Home data between 1st January 2017 and 31st December 2019.

6.2 Model

In the model, laundry detergent is storable, and households derive utility from both consumption and purchase. As in Hendel and Nevo [2006a], we assume that laundry detergents are perfect substitutes in consumption which implies that the unobserved state variable, inventory, is one-dimensional. At week $t \leq +\infty$, consumer i chooses discrete consumption c_t and purchase decision a_{jklt}, where $j \in \{0, 23, 40, 64\}$ is the quantity measured by the number of washes, k refers to the brand, and l is bio/non-bio. Let $\boldsymbol{a}_t$ be a vector of purchase decisions with $\sum_{j,k,l} a_{jklt} = 1$, where $a_{jklt} = 1$ indicates the purchase of quantity j of brand k and bio/non-bio l, and $a_{jklt} = 0$ otherwise. The period utility[15] is given by:

$$u(\boldsymbol{a}_t, c_t, I_t, \boldsymbol{p}_t, \varepsilon_t; \theta) = \theta_1 c_t + \theta_2 c_t^2 + \theta_3 I_{t+1}^2 + \theta_4 (\sum_{j>0,k,l} a_{jklt} > 0) + \sum_{j,k,l} a_{jklt}(\theta_5 p_{jklt} + j\xi_k + j\delta_l + \varepsilon(a_{jklt}))$$

where I_t is the inventory at the beginning of t and $\boldsymbol{p}_t$ is a vector of prices. $\varepsilon(a_{jklt})$ is the choice-specific utility shock following an i.i.d. Type I Extreme Value distribution. (θ_1, θ_2) capture the marginal utility of consumption. θ_3 captures the carrying cost that depends on the inventory at next period: $I_{t+1} := I_t + j_t - c_t$. The fixed cost of making a purchase is θ_4. We impose an upper bound on the inventory $I_t \in \{0, 1, \cdots, I_{max}\}$ where[16] $I_{max} = 80$. The consumption is bounded from below and above by $c_{min,t} := \max\{0, I_t + j_t - I_{max}\}$ and $c_{max,t} := I_t + j_t$, respectively. θ_5 captures the price sensitivity. ξ_k and δ_l are brand and bio/non-bio fixed effects per wash.

[15] Our model differs from Hendel and Nevo [2006a] as we do not include a taste shock for consumption. Osborne [2018] also found that it is difficult to identify its distribution.

[16] We impose the upper bound of 80 washes because it approximately equals one-third of the average annual washes (256) for the largest household size. This bound helps maintain computational feasibility while still capturing consumers' stockpiling.

A household maximizes expected life-time utility:

$$\begin{aligned}\max_{\{\boldsymbol{a}_t, c_t\}_{t=0}^{\infty}} \mathbb{E}\left[\sum_{t=0}^{+\infty} \beta^t u(\boldsymbol{a}_t, I_t, \boldsymbol{p}_t, \varepsilon_t; \theta) | I_0, \boldsymbol{p}_0\right] \\ \text{s.t.} \quad c_t \in [c_{min,t}, c_{max,t}] \\ I_{t+1} := I_t + j_t - c_t\end{aligned}$$

where $\beta = 0.99$. The state space is defined by $s_t := (I_t, \boldsymbol{p}_t)$.

6.3 Estimation Overview

This section presents the estimation procedure. We mainly follow the three-step procedure in Hendel and Nevo [2006a]. We assume $\boldsymbol{p}_t$ follows an exogenous first-order Markov process. Therefore, inventory I_t is the only endogenous state variable. This allows us to decompose the decisions into dynamic decisions (c_t, j_t) and static decisions (k_t, l_t), as the evolution of I_t is determined only by (c_t, j_t). Moreover, as shown in Hendel and Nevo [2006a], the choice probability of purchasing brand k and bio/non-bio l conditional on purchasing quantity j has the conditional logit model form:

$$Pr(k, l | \boldsymbol{p}, j) = \frac{\exp(\theta_5 p_{jkl} + j\xi_k + j\delta_l)}{\sum_{k,l} \exp(\theta_5 p_{jkl} + j\xi_k + j\delta_l)}$$

allowing for simple estimation of $(\theta_5, \xi_k, \delta_l)$.

Then, the inclusive value of purchasing quantity j is defined by:

$$\omega_{j,t} = \log\left[\sum_{k,l} \exp(\theta_5 p_{jklt} + j\xi_k + j\delta_l)\right]$$

where $\omega_{j,t}$ is the indirect utility of purchasing quantity j at period t. We impose the inclusive value sufficiency (IVS) assumption as in Hendel and Nevo [2006a]:

Assumption 6 (Inclusive Value Sufficiency)**.** *$F(\boldsymbol{\omega_t} | \boldsymbol{p_{t-1}})$ can be summarized by $F(\boldsymbol{\omega_t} | \boldsymbol{\omega_{t-1}})$ where F is the conditional distribution function.*

Given Assumption 6, the lagged inclusive value $\boldsymbol{\omega_{t-1}}$ is a sufficient statistic to forecast $\boldsymbol{\omega_t}$. As a result, the state space can be reduced from $s_t := (I_t, \boldsymbol{p}_t)$ to $x_t := \{I_t, \boldsymbol{\omega}_t\}$. We forecast the inclusive values using a VAR(1) and then discretize it into 5^3 bins.[17] The simplified

[17] The discretization is implemented by quantecon.markov.approximation.discrete_var, which is based on

dynamic programming problem based on the IVS assumption is:

$$V(x,\varepsilon) = \max_{j, c_{min} \le c \le c_{max}} \Big[U(c, I, j; \theta) + \omega_j + \varepsilon(j) + \beta \mathbb{E}[V(x')|x, c, j] \Big]$$

where $U(c, I, j; \theta) = \theta_1 c + \theta_2 c^2 + \theta_3 I'^2 + \theta_4 (\sum_{j>0,k,l} a_{jkl} > 0)$, ε is a 4-dimensional vector of i.i.d Type I Extreme Value distribution, and $V(x) = \mathbb{E}_\varepsilon V(x, \varepsilon)$.

We decompose the joint optimization problem into sequential optimization problems; first choose purchase quantity j, then choose consumption. For the second problem, given state x, purchase quantity j, and $V(x)$, define the consumption function as:

$$C(x, j) := \operatorname*{argmax}_{c_{min} \le c \le c_{max}} \Big[U(c, I, j; \theta) + \omega_j + \beta \mathbb{E}[V(x')|x, c, j] \Big] \tag{6.1}$$

For the first problem, the choice probability of purchasing quantity j is:

$$p(j|x) = \frac{\exp(v(x, j))}{\sum_{j'} \exp(v(x, j'))}$$

where $v(x, j) := U(C(x, j), I, j; \theta) + \omega_j + \beta \mathbb{E}[V(x')|x, C(x, j), j]$. Combining these two steps, we have:

$$V(x) = \log \sum_j \exp(U(C(x, j), I, j; \theta) + \omega_j + \beta \mathbb{E}[V(x')|x, C(x, j), j]) \tag{6.2}$$

For a given consumption function, we solve (6.2) by policy iteration with our model-adaptive approach. Then, we update the consumption function using (6.1), and iterate until convergence (see Algorithm 3).

As the argmax operator in Algorithm 3 for discrete outcomes is not differentiable, we estimate the dynamic parameters by the Bayesian MCMC estimator Chernozhukov and Hong [2003]. The estimation details are described in Online Appendix B.3.

6.4 Results and Implications

We separately estimate the model for different household sizes. Table 8 presents the results of the conditional logit model. The coefficient θ_5 measures the price sensitivity, which is negative and statistically significant at 5% level across all household sizes. Single-person households exhibit the highest price sensitivity at -0.501, followed by households with

simulation. The state space is of cardinality $5^3 \times 81 = 10125$.

three members at -0.416, households with two members at -0.345, and households with four members at -0.307. Five-person households have the lowest price sensitivity at -0.223. The estimates also suggest the heterogeneity in price sensitivity across different household sizes.

Table 8: Conditional Logit: Brand and Bio Choice Conditional on Quantity

Household Size	**1**	**2**	**3**	**4**	**5**
θ_5	-0.501 (0.029)	-0.345 (0.014)	-0.416 (0.020)	-0.307 (0.018)	-0.223 (0.030)
Brand FE * Quantity	Yes	Yes	Yes	Yes	Yes
Bio FE * Quantity	Yes	Yes	Yes	Yes	Yes

Note: Standard errors are in parentheses. Estimates of a conditional logit model. We use the transaction price for the observed purchase and the price index for other choices.

In Table 9 we report the estimates of dynamic parameters and computational times. All estimates have the expected signs and are statistically significant at the 5% level. The utility from consumption, determined jointly by the linear term (θ_1) and quadratic term (θ_2), shows varying patterns across household types. The linear coefficient θ_1 ranges from its lowest value of 1.241 for four-person households to its peak of 3.531 for three-person households. Five-person households have the lowest magnitude for the quadratic term θ_2 (-8.283). Single-person households have the lowest fixed cost of making a purchase ($\theta_4 = -4.184$), while five-person households have a relatively high fixed cost ($\theta_4 = -5.336$). The average time ranges from 0.89 to 1.47 minutes per Metropolis-Hastings step. The total time of the MCMC estimator ranges from 6.2 to 10.2 days, which confirms the computational efficiency of our model-adaptive approach in practice.

To investigate the model fit, we compare the simulated and observed purchase behavior. For each household size, we simulate consumption and purchase behavior for the same number of households as observed in the data over 156 weeks (3 years).[18] The simulated market shares in Table 10 reasonably match the observed data. Figure 8 plots the hazard rates and their confidence intervals. We also estimate the static model without the carrying cost.[19] All household sizes exhibit a similar pattern, with the hazard rates initially increasing until stabilizing at the highest level. For single-person and two-person households, the hazard

[18]The first 30% periods of both the simulated and observed data are discarded as it is used to simulate the initial inventory for the estimation.

[19]For the static model, we assume households consume the entire pack when a purchase is made. Therefore, we have $U(\omega_t, j_t, \varepsilon_t) := \theta_1^{static} j_t + \theta_2^{static} j_t^2 + \omega_{jt} + \theta_4^{static} \mathbb{1}(j_t > 0) + \varepsilon(j_t)$. The hazard rate is then the probability of making a purchase.

Table 9: Estimates of Dynamic Parameters

Household Size	**1**	**2**	**3**	**4**	**5**
θ_1	2.020	2.658	3.531	1.241	1.843
	(0.369)	(0.187)	(0.262)	(0.224)	(0.517)
θ_2	-13.768	-14.819	-13.979	-11.019	-8.283
	(0.902)	(0.477)	(0.563)	(0.476)	(0.802)
θ_3	-3.197	-2.235	-3.188	-2.375	-4.141
	(0.217)	(0.081)	(0.149)	(0.117)	(0.815)
θ_4	-4.184	-4.866	-4.913	-5.117	-5.336
	(0.078)	(0.041)	(0.061)	(0.054)	(0.120)
Avg Time (mins)	1.18	0.91	0.99	0.89	1.47
Total Time (days)	8.2	6.3	6.9	6.2	10.2

Note: The first 8,000 of 10,000 MCMC draws are discarded as burn-in. The means are taken over last 2,000 MCMC draws. The standard errors (in parentheses) are the standard deviation of the MCMC draws. The computational time is the average time per Metropolis-Hastings step. The code runs on an Intel Xeon Gold 6240 CPU (2.60GHz) with 192GB RAM.

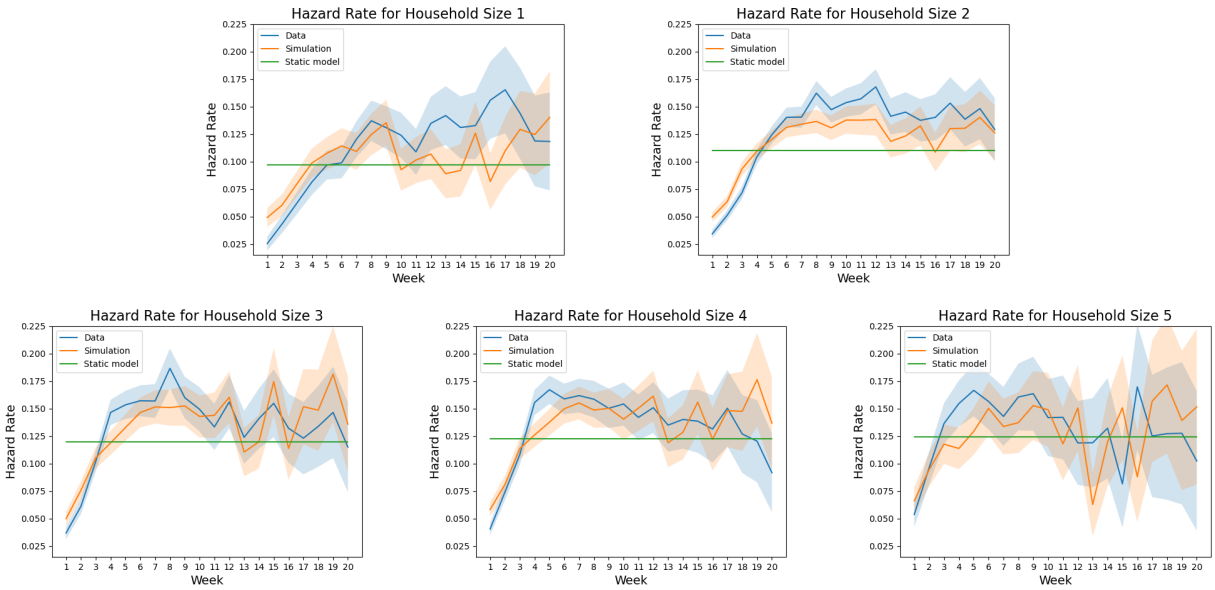
rates increase until around 8 weeks, while the hazard rates increase until around 5 weeks for larger households. The hazard rates all peak around 15%. The dynamic model outperforms the static model in capturing the hazard rates as the static model, by construction, cannot capture the increasing hazard rate observed in the data.

Table 10: Model Fitting for Different Household Sizes: Market Share

Household Size		**Small**	**Medium**	**Large**
1	Observed	5.76 (0.14)	2.96 (0.11)	1.09 (0.06)
	Simulated	5.50 (0.14)	3.23 (0.11)	0.93 (0.06)
2	Observed	5.38 (0.08)	3.90 (0.07)	1.78 (0.05)
	Simulated	5.19 (0.08)	4.13 (0.07)	1.60 (0.04)
3	Observed	5.27 (0.11)	4.66 (0.10)	2.14 (0.07)
	Simulated	5.15 (0.11)	4.65 (0.11)	2.08 (0.07)
4	Observed	4.94 (0.11)	4.66 (0.10)	2.78 (0.08)
	Simulated	4.83 (0.11)	4.70 (0.10)	2.71 (0.08)
5	Observed	4.90 (0.19)	4.90 (0.19)	2.71 (0.14)
	Simulated	4.81 (0.19)	4.71 (0.19)	2.56 (0.14)

Note: Values represent percentages. We simulate the same number of households as observed in the data over 156 weeks and discard the first 30% periods of both the simulated and observed data. The standard errors of the simulated and observed data are in parentheses.

Long-run elasticities measure the effects of permanent price changes on market shares. Table 11 simulates the long-run elasticities for different household sizes. Own-price elastici-



Note: For each household size, we simulate the same number of households as observed in the data over 156 weeks and discard the first 30% periods of both the simulated and observed data. The light blue lines in each figure represent the 95% confidence intervals of the hazard rates of the observed data. The light orange lines represent the 95% confidence intervals of the hazard rates of the simulated data.

Figure 8: Hazard Rates of Purchase Across Different Household Sizes

ties are larger in absolute value for larger pack sizes. They are all greater than 1 in absolute value, except the small pack for household size 5 (-0.744), indicating that the demand is generally elastic. For single-person households, the own-price elasticity for large packs is -4.072, meaning that if the prices of all large packs increase by 1%, the market shares of large pack will decrease by 4.072%. As household size increases, there is a trend towards lower price elasticities, suggesting that larger households are less sensitive to price changes. This trend can partially be explained by the price coefficients (θ_5) as larger households are more price sensitive, except for household size 2, as shown in Table 8.

The demand for each size is elastic in general, indicating consumers are likely to substitute out of the pack size that has increased in price and to other pack sizes or not purchase at all. For single-person households, the cross-price elasticity between medium and large packs is 0.627, higher than the cross-price elasticity between medium and small packs (0.251). This pattern continues across household sizes - for 2-person households, the medium-large cross-price elasticity is 0.382, for 3-person households it increases to 0.635, showing stronger substitution effects for larger sizes. The cross-elasticities between medium and small packs remain relatively stable across household sizes (ranging from 0.251 to 0.269 for households of 1-3 persons), while the substitution effects involving large packs tend to be stronger. These findings suggest that consumers are more likely to substitute between larger pack sizes when

prices change, particularly in households with 2-3 members, though this effect diminishes somewhat for the largest households.

Table 11: Long Run Elasticities for Different Household Sizes

Household Size		Small	Medium	Large
1	**Small**	**-1.635** (0.062)	**0.228** (0.059)	**0.438** (0.095)
	Medium	**0.251** (0.077)	**-2.437** (0.099)	**0.627** (0.154)
	Large	0.107 (0.056)	**0.256** (0.084)	**-4.096** (0.196)
2	**Small**	**-1.136** (0.047)	**0.217** (0.034)	**0.164** (0.050)
	Medium	**0.255** (0.054)	**-1.596** (0.050)	**0.382** (0.060)
	Large	**0.173** (0.043)	**0.229** (0.058)	**-2.598** (0.105)
3	**Small**	**-1.464** (0.057)	**0.272** (0.041)	**0.240** (0.054)
	Medium	**0.269** (0.062)	**-1.987** (0.056)	**0.635** (0.067)
	Large	**0.186** (0.062)	**0.367** (0.077)	**-2.985** (0.137)
4	**Small**	**-1.110** (0.060)	**0.136** (0.047)	**0.178** (0.042)
	Medium	**0.143** (0.056)	**-1.396** (0.056)	**0.292** (0.053)
	Large	**0.172** (0.058)	**0.282** (0.074)	**-2.204** (0.077)
5	**Small**	**-0.746** (0.042)	0.039 (0.031)	**0.110** (0.039)
	Medium	0.101 (0.073)	**-1.037** (0.060)	**0.176** (0.038)
	Large	**0.126** (0.064)	**0.157** (0.064)	**-1.652** (0.071)

Note: We increase the price of each pack size by 1% to simulate the long-run elasticities. For each of the 2000 MCMC draws, we solve the dynamic programming problem, simulate 1000 households over 2000 weeks and discard the first 30% periods. The means are taken over the 2000 MCMC draws. The standard errors (in parentheses) are the standard deviations of simulated elasticities across 2000 MCMC draws. Estimates in bold are statistically significant at the 5% level.

7 Conclusion

In this paper, we propose a novel model-adaptive approach to solve the linear system of fixed point equations of the policy valuation operator. Our method adaptively constructs the sieve space and chooses its dimension. It converges superlinearly, while conventional methods do not. We demonstrate through simulations that our model-adaptive sieves dramatically improves the computational efficiency of policy iteration and opens the door to the use of Bayesian MCMC estimator for DDC models.

We apply our method to analyze consumer demand for laundry detergent using Kantar's Worldpanel Take Home data. The empirical application confirms that our approach improves computational efficiency of policy iteration in practice, which opens the door to the estimation of DDC models by Bayesian MCMC estimator. To investigate the model fit, we simulate market shares and hazard rates for different household sizes. The model achieves

a reasonable model fitting. We also simulate the long-run elasticities. The results show the heterogeneous substitution patterns across different household sizes.

This research opens several avenues for future investigation. First, our method can be applied to random coefficient DDC models. For example, Wang [2015] estimates a random coefficient DDC model for consumer demand for soft drinks by the importance sampling estimator Ackerberg [2009]. For each draw from the distribution of random coefficients, Wang [2015] solves the dynamic programming problem by sieve approximation method proposed by Arcidiacono et al. [2013]. We may solve the DP problem by policy iteration with our model-adaptive approach instead. Second, a comparative study of the performance of our method in the Bayesian MCMC estimation against that of Imai et al. [2009] would be interesting. They propose to iterate the Bellman operator once at each MCMC step. Imai et al. [2009] Theorem 2 shows that their Markov Chain will converge to the stationary distribution of the Markov Chain obtained by the conventional Bayesian MCMC estimator. However, we conjecture that for large discount factors, their Markov Chain may take much more iterations to converge to the stationary distribution, resulting in more computational time than the Bayesian MCMC estimator with our model-adaptive approach. Third, our method can be applied to dynamic games which usually have a large state space.

References

Daniel A Ackerberg. A new use of importance sampling to reduce computational burden in simulation estimation. *QME*, 7:343–376, 2009.

Karun Adusumilli and Dita Eckardt. Temporal-difference estimation of dynamic discrete choice models. *arXiv preprint arXiv:1912.09509*, 2024.

Victor Aguirregabiria and Arvind Magesan. Solution and estimation of dynamic discrete choice structural models using euler equations. *Available at SSRN 2860973*, 2023.

Victor Aguirregabiria and Pedro Mira. Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica*, 70(4):1519–1543, 2002.

Victor Aguirregabiria and Pedro Mira. Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53, 2007.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18:343–373, 2008.

Peter Arcidiacono and Robert A Miller. Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79(6):1823–1867, 2011.

Peter Arcidiacono, Patrick Bayer, Federico A Bugni, and Jonathan James. Approximating high-dimensional dynamic models: Sieve value function iteration. In *Structural Econometric Models*, volume 31, pages 45–95. Emerald Group Publishing Limited, 2013.

Kendall Atkinson. Convergence rates for approximate eigenvalues of compact integral oper-

ators. *SIAM Journal on Numerical Analysis*, 12(2):213–222, 1975.

Kendall E Atkinson. *The numerical solution of integral equations of the second kind*, volume 4. Cambridge university press, 1997.

Bee Yan Aw, Mark J Roberts, and Daniel Yi Xu. R&d investment, exporting, and productivity dynamics. *American Economic Review*, 101(4):1312–1344, 2011.

Dimitri P Bertsekas. Dynamic programming and optimal control 4th edition, volume ii. *Athena Scientific*, 2015.

Martin Blümlinger and Robert F Tichy. Topological algebras of functions of bounded variation i. *manuscripta mathematica*, 65(2):245–255, 1989.

Pierre Bodéré. Dynamic spatial competition in early education: An equilibrium analysis of the preschool market in pennsylvania. *Job Market Paper*, 2023.

Marine Carrasco, Jean-Pierre Florens, and Eric Renault. Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization. *Handbook of econometrics*, 6:5633–5751, 2007.

Victor Chernozhukov and Han Hong. An mcmc approach to classical estimation. *Journal of econometrics*, 115(2):293–346, 2003.

John B Conway. *A course in functional analysis*, volume 96. Springer, 2019.

Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

Jean-Pierre Dubé, Jeremy T Fox, and Che-Lin Su. Improving the numerical performance of static and dynamic aggregate discrete choice random coefficients demand estimation. *Econometrica*, 80(5):2231–2267, 2012.

Pierre Dubois, Rachel Griffith, and Aviv Nevo. Do prices and attributes explain international differences in food purchases? *American Economic Review*, 104(3):832–867, 2014.

Pierre Dubois, Rachel Griffith, and Martin O'Connell. How well targeted are soda taxes? *American Economic Review*, 110(11):3661–3704, 2020.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Tülin Erdem, Susumu Imai, and Michael P Keane. Brand and quantity choice dynamics under price uncertainty. *Quantitative Marketing and economics*, 1:5–64, 2003.

José D Flores. The conjugate gradient method for solving fredholm integral equations of the second kind. *International Journal of Computer Mathematics*, 48(1-2):77–94, 1993.

Todd D Gerarden. Demanding innovation: The impact of consumer subsidies on solar panel production costs. *Management Science*, 69(12):7799–7820, 2023.

Gautam Gowrisankaran and Marc Rysman. Dynamics of consumer demand for new durable goods. *Journal of political Economy*, 120(6):1173–1219, 2012.

Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

Paul LE Grieco, Shengyu Li, and Hongsong Zhang. Input prices, productivity, and trade dynamics: long-run effects of liberalization on chinese paint manufacturers. *The RAND Journal of Economics*, 53(3):516–560, 2022.

Wolfgang Hackbusch and Zenon Paul Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.

J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Springer Netherlands, Dordrecht, 1964. ISBN 978-94-009-5821-0 978-94-009-5819-7. doi: 10.1007/978-94-009-5819-7.

Weimin Han and Kendall E Atkinson. *Theoretical numerical analysis: A functional analysis framework*. Springer, 2009.

Igal Hendel and Aviv Nevo. Measuring the implications of sales and consumer inventory behavior. *Econometrica*, 74(6):1637–1673, 2006a.

Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Edmund Hlawka. Funktionen von beschränkter variatiou in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata*, 54(1):325–333, 1961.

V Joseph Hotz and Robert A Miller. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529, 1993.

Ronald A Howard. Dynamic programming and markov processes. *MIT Press*, 2:39–47, 1960.

Ling Huang and Martin D Smith. The dynamic efficiency costs of common-pool resource exploitation. *American Economic Review*, 104(12):4071–4103, 2014.

Yan Huang, Param Vir Singh, and Anindya Ghose. A structural model of employee behavioral dynamics in enterprise social media. *Management Science*, 61(12):2825–2844, 2015.

Susumu Imai, Neelam Jain, and Andrew Ching. Bayesian estimation of dynamic discrete choice models. *Econometrica*, 77(6):1865–1899, 2009.

Kenneth L Judd. *Numerical methods in economics*. MIT press, 1998.

Myrto Kalouptsidi. Time to build and fluctuations in bulk shipping. *American Economic Review*, 104(2):564–608, 2014.

Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

Rainer Kress. *Linear Integral Equations*, volume 82 of *Applied Mathematical Sciences*. Springer, New York, NY, 2014. ISBN 978-1-4614-9592-5 978-1-4614-9593-2. doi: 10.1007/978-1-4614-9593-2.

Lawrence Narici and Edward Beckenstein. *Topological vector spaces*. CRC Press, 2010.

Andriy Norets. Estimation of dynamic discrete choice models using artificial neural network approximations. *Econometric Reviews*, 31(1):84–106, 2012.

Erich Novak. *Deterministic and stochastic error bounds in numerical analysis*, volume 1349. Springer, 2006.

James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Matthew Osborne. Approximating the cost-of-living index for a storable good. *American Economic Journal: Microeconomics*, 10(2):286–314, 2018.

Art B Owen. Multidimensional variation for quasi-monte carlo. In *Contemporary Multivariate Analysis And Design Of Experiments: In Celebration of Professor Kai-Tai Fang's 65th Birthday*, pages 49–74. World Scientific, 2005.

Martin Pesendorfer and Philipp Schmidt-Dengler. Asymptotic least squares estimators for dynamic games. *The Review of Economic Studies*, 75(3):901–928, 2008.

Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics: Functional Analysis; Rev. ed*. Academic press, 1980.

Vladimir Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of computational physics*, 60(2):187–207, 1985.

W. Rudin. *Real and Complex Analysis*. Higher Mathematics Series. McGraw-Hill Education, 1987. ISBN 9780070542341.

John Rust. Structural estimation of markov decision processes. *Handbook of econometrics*, 4:3081–3143, 1994.

John Rust. Using randomization to break the curse of dimensionality. *Econometrica: Journal of the Econometric Society*, pages 487–516, 1997a.

John P Rust. A comparison of policy iteration methods for solving continuous-state, infinite-horizon markovian decision problems using random, quasi-random, and deterministic discretizations. *Infinite-Horizon Markovian Decision Problems Using Random, Quasi-random, and Deterministic Discretizations (April 1997)*, 1997b.

Andrew Sweeting. Dynamic product positioning in differentiated product markets: The effect of fees for musical performance rights on the commercial radio industry. *Econometrica*, 81 (5):1763–1803, 2013.

Terence Tao. *An introduction to measure theory*, volume 126. American Mathematical Society Providence, 2011.

Terence Tao. *An Epsilon of Room, I: Real Analysis: pages from year three of a mathematical blog*, volume 117. American Mathematical Society, 2022.

George Tauchen. Finite state markov-chain approximations to univariate and vector autoregressions. *Economics letters*, 20(2):177–181, 1986.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

Emily Yucai Wang. The impact of soda taxes on consumer welfare: implications of storability and taste heterogeneity. *The RAND Journal of Economics*, 46(2):409–441, 2015.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural networks*, 94:103–114, 2017.

Appendix

A Proofs

Define $\mathcal{K} := \mathcal{T} + \mathcal{T}^* - \mathcal{T}\mathcal{T}^*$. The constants in this section can vary from line to line.

Proof of Theorem 1. (i) By Lemma 4, it suffices to show that $(\mathcal{I} - \mathcal{T}^*)$ is invertible on $L^2(\mathbb{X})$. By Assumption 1(v), we have $(\mathcal{I} - \mathcal{T})$ is a bounded operator on $L^2(\mathbb{X})$, since $\|\mathcal{I} - \mathcal{T}\|_{op} \leq 1 + \|\mathcal{T}\|_{op} \leq 1 + \|\mathcal{T}\|_{HS} < +\infty$ where we used $\mathcal{T}$ is a Hilbert-Schmidt operator as showed in Lemma 6. Thus, it is continuous by Kress [2014] Theorem 2.3. By Lemma 3, $L^2(\mathbb{X})$ is a separable Hilbert space. Moreover, $(\mathcal{I} - \mathcal{T})$ is bijective as it is invertible on $L^2(\mathbb{X})$ as showed in Lemma 4. Thus, by the bounded inverse theorem

(e.g., Narici and Beckenstein [2010]), $(\mathcal{I}-\mathcal{T})^{-1}$ is also bounded. By Reed and Simon [1980] Theorem VI.3 (d), the adjoint operator $(\mathcal{I}-\mathcal{T}^*)$ is invertible and has a bounded inverse on $L^2(\mathbb{X})$.

(ii) Let V^* be the unique solution to $(\mathcal{I}-\mathcal{T})V = u$ on $L^2(\mathbb{X},\mu)$. Note that $(\mathcal{I}-\mathcal{T}^*)y^*$ is the solution to $(\mathcal{I}-\mathcal{T})V = u$ on $L^2(\mathbb{X})$. Since the norm $\|\cdot\|_\mu$ and $\|\cdot\|$ are equivalent, $(\mathcal{I}-\mathcal{T}^*)y^* \in L^2(\mathbb{X},\mu)$. Therefore, $(\mathcal{I}-\mathcal{T}^*)y^*$ is also the solution on $L^2(\mathbb{X},\mu)$. As $(\mathcal{I}-\mathcal{T})V = u$ has a unique solution on $L^2(\mathbb{X},\mu)$, we have $(\mathcal{I}-\mathcal{T}^*)y^* = V^*$, μ-a.s.

□

Proof of Theorem 2. (i) Lemma 8(i) proves the orthogonality.

(ii) Let $\|h\|^2_{(\mathcal{I}-\mathcal{K})} := \langle(\mathcal{I}-\mathcal{K})h, h\rangle$. By Han and Atkinson [2009] page 251, we have:

$$y_k = \underset{y\in\text{span}\{r_0,r_1,\cdots,r_{k-1}\}}{\text{argmin}} \|y - y^*\|_{(\mathcal{I}-\mathcal{K})}$$

Note that:

$$\begin{aligned}
\|y_k - y^*\|^2_{(\mathcal{I}-\mathcal{K})} &= \langle(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^*)(y_k-y^*), (y_k-y^*)\rangle \\
&= \langle(\mathcal{I}-\mathcal{T}^*)(y_k-y^*), (\mathcal{I}-\mathcal{T}^*)(y_k-y^*)\rangle \\
&= \|(\mathcal{I}-\mathcal{T}^*)(y_k-y^*)\|^2 \\
&= \|(\mathcal{I}-\mathcal{T}^*)y_k - V^*\|^2
\end{aligned}$$

Therefore, we have $y_k = \text{argmin}_{y\in\text{span}\{r_0,r_1,\cdots,r_{k-1}\}} \|(\mathcal{I}-\mathcal{T}^*)y_k - V^*\|$.

□

Proof of Theorem 3. As $\mathcal{K}$ is a compact self-adjoint operator showed in Lemma 7, its eigenvalues are real and countable. Without loss of generality, we assume the eigenvalues are ordered as follows: $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq 0$ where $\lim_{j\to\infty}|\lambda_j| = 0$. The eigenvalues of $(\mathcal{I}-\mathcal{K})$ are $\{1-\lambda_j\}_{j\geq 1}$. By Kress [2014] Theorem 15.11 and Definition 3.8, we have $\Delta := \|\mathcal{I}-\mathcal{K}\|_{op} = \sup_j(1-\lambda_j)$. Since $(\mathcal{I}-\mathcal{K})$ is positive definite, we have $\Delta > 0$. By Kress [2014] Theorem 3.9, zero is the only possible accumulation point of the eigenvalues $\{\lambda_j\}_{j\geq 1}$. Thus, $\delta := \inf_j(1-\lambda_j) > 0$. The eigenvalues of $(\mathcal{I}-\mathcal{K})^{-1}$ are $\{\frac{1}{1-\lambda_j}\}_{j\geq 1}$, implying $\|(\mathcal{I}-\mathcal{K})^{-1}\|_{op} = \frac{1}{\inf_j(1-\lambda_j)} = \frac{1}{\delta}$. By Lemmas 3 and 7, conditions in Han and Atkinson [2009] Theorem 5.6.2 are satisfied for both discrete and continuous state spaces. Thus, the superlinear convergence and the approximation error upper bound hold for $\{y_k\}_{k\geq 1}$.

Specifically, we have:

$$\|y_k - y^*\| = O((\tau_k)^k) = O((c_k)^k)$$

where $c_k := \frac{2}{k}\sum_{j=1}^{k}\frac{|\lambda_j|}{1-\lambda_j}$ and τ_k is defined in Lemma 9. As $(\mathcal{I} - \mathcal{T}^*)$ is a bounded linear operator, we have:

$$\|V_k^{ma} - V^*\| = \|(\mathcal{I} - \mathcal{T}^*)y_k - (\mathcal{I} - \mathcal{T}^*)y^*\| \leq \|\mathcal{I} - \mathcal{T}^*\|_{op}\|y_k - y^*\| = O((c_k)^k)$$

Then, Lemma 9 applies. The monotonic decreasing directly follows from Theorem 2.

The third part follows from:

$$\|r_k\| = \|u - (\mathcal{I} - \mathcal{K})y_k\| = \|(\mathcal{I} - \mathcal{K})(y^* - y_k)\| \leq \|\mathcal{I} - \mathcal{K}\|_{op}\|y^* - y_k\| = O((c_k)^k)$$

where we used $\|\mathcal{I} - \mathcal{K}\|_{op} \leq 1 + \|\mathcal{K}\|_{op} \leq 1 + \|\mathcal{K}\|_{HS} < +\infty$ and $\mathcal{K}$ is a Hilbert-Schmidt operator as showed in Lemma 7.

For the last part, see Atkinson [1997] page 299. □

Proof of Theorem 4.

1. By Lemma 10, we have: $\|V_k^{td} - V^*\|_\mu \to 0$ as $k \to +\infty$. Thus, $\|V_k^{td} - V^*\| \to 0$ as $k \to +\infty$ by Assumption 1. Moreover, we have: for $\forall\ i$,

$$\frac{\|V_{i+1}^{td} - V^*\|_\mu}{\|V_i^{td} - V^*\|_\mu} \geq \frac{\frac{C_{td,1}}{1+\beta}}{\frac{C_{td,2}}{1-\beta}}(\frac{i+1}{i})^{-\frac{\alpha}{d}} \geq \frac{\frac{C_{td,1}}{1+\beta}}{\frac{C_{td,2}}{1-\beta}}(\frac{1}{2})^{\frac{\alpha}{d}} := C_{TD} > 0$$

   where $C_{TD} < 1$. Then, we have:

$$\|V_k^{td} - V^*\|_\mu = \|V_1^{td} - V^*\|_\mu \times \Pi_{i=1}^{k-1}\frac{\|V_{i+1}^{td} - V^*\|_\mu}{\|V_i^{td} - V^*\|_\mu} \geq \|V_1^{td} - V^*\|_\mu (C_{TD})^{k-1}$$

   Therefore, we have: $\|V_k^{td} - V^*\|_\mu^{\frac{1}{k}} \geq C_{TD}(\frac{\|V_1^{td} - V^*\|_\mu}{C_{TD}})^{\frac{1}{k}}$. Under Assumption 1, we have:

$$\limsup_{k\to+\infty}\|V_k^{td} - V^*\|^{\frac{1}{k}} > \limsup_{k\to+\infty}(\frac{1}{\sqrt{C_{\mu,2}}}\|V_k^{td} - V^*\|_\mu)^{\frac{1}{k}} > \limsup_{k\to+\infty} C_{TD}(\frac{\|V_1^{td} - V^*\|}{\sqrt{C_{\mu,2}}C_{TD}})^{\frac{1}{k}} = C_{TD} > 0$$

2. Note that: $\|V_k^{sa} - V^*\|_\mu \leq \beta^k\|V_0^{sa} - V^*\|_\mu$. Therefore, we have:

$$\limsup_{k\to+\infty}\|V_k^{sa} - V^*\|^{\frac{1}{k}} \leq \limsup_{k\to+\infty}(\frac{1}{\sqrt{C_{\mu,1}}}\|V_k^{sa} - V^*\|_\mu)^{\frac{1}{k}} \leq \limsup_{k\to+\infty}(\frac{1}{\sqrt{C_{\mu,1}}}\|V_0^{sa} - V^*\|_\mu)^{\frac{1}{k}}\beta = \beta$$

□

Proof of Theorem 5. Combining Lemma 12, 16 and 20 gives:

- If a low-discrepancy grid is used, then:

$$\|V_M - V_M^*\|_{\mathbb{M}} = O(\frac{(\log M)^{d-1}}{M})$$

- If a regular grid is used, then:

$$\|V_M - V_M^*\|_{\mathbb{M}} = O(M^{-\frac{\alpha}{d}})$$

where V_M, V_M^*, $\|\cdot\|_{\mathbb{M}}$ are defined in Lemma 12.

We only prove the bound for the low-discrepancy grid. The proof for the regular grid is similar. For the approximate solution $\widehat{V}_k^{ma}$ to V_M, by a slight modification of notations of Theorem 3, we have:

$$\|\widehat{V}_k^{ma} - V_M\|_{\tilde{\mathbb{M}}} = O((c_{1,k,M})^k)$$

where $\|\cdot\|_{\tilde{\mathbb{M}}}$ is the Euclidean norm of a $\mathbb{M}$-dimensional vector. Since $\|\cdot\|_{\tilde{\mathbb{M}}}$ is equivalent to $\|\cdot\|_{\mathbb{M}}$, we have:

$$\|\widehat{V}_k^{ma} - V_M\|_{\mathbb{M}} = O((c_{1,k,M})^k)$$

Thus, for $x \in \mathbb{M}$, we have:

$$\|\widehat{V}_k^{ma} - V_M^*\|_{\mathbb{M}} = O((c_{1,k,M})^k + \frac{(\log M)^{d-1}}{M})$$

Then, for $x \in \mathbb{X}\backslash\mathbb{M}$, by Lemma 15, and similar arguments as in the proof of Lemma 16, we have:

$$\begin{aligned}
|\widehat{V}_k^{ma}(x) - V^*(x)| &\leq |\beta \sum_i \frac{f(x_i|x)}{\sum_j f(x_j|x)} (\widehat{V}_k^{ma}(x_i) - V^*(x_i))| \\
&\qquad + |\beta \sum_i \frac{f(x_i|x)}{\sum_j f(x_j|x)} V^*(x_i) - \beta \int f(x'|x) V^*(x') dx'| \\
&\leq \beta \|\widehat{V}_k^{ma} - V_M^*\|_{\mathbb{M}} + O(\frac{(\log M)^{d-1}}{M}) \\
&= O((c_{1,k,M})^k + \frac{(\log M)^{d-1}}{M} + \frac{(\log M)^{d-1}}{M})
\end{aligned}$$

Since the bound is uniformly over x, we have:

$$\|\widehat{V}_k^{ma} - V^*\| = O((c_{1,k,M})^k + \frac{(\log M)^{d-1}}{M})$$

Furthermore, note that the discretized transition density has continuous partial derivative and the kernel for $(\hat{\mathcal{T}}_M + \hat{\mathcal{T}}_M^* - \hat{\mathcal{T}}_M\hat{\mathcal{T}}_M^*)$ is symmetric. Therefore, Atkinson [1997] page 299 applies. As the numerical operator depends on M, the constant also depends on M.

For regular grids, the proof is the similar with Lemma 15 replaced by Lemma 18. □

Proof of Theorem 6 and Corollary 1. In the following, we use the notation $c_{k,M}$ and $\lambda_{k,M}$ instead of $c_{1,k,M}$ ($c_{2,k,M}$) and $\lambda_{1,k,M}$ ($\lambda_{2,k,M}$) as the proof is the same.

Under Assumption 4 and by Lemma 25, conditions A1-A3 in Atkinson [1975] are satisfied. Recall $\mathcal{K}$ is a compact self-adjoint operator. By Han and Atkinson [2009] Theorem 2.8.15, the index of a self-adjoint compact operator is 1. By Atkinson [1975], we have for $k \leq p$ and sufficiently large M:

$$|\lambda_{k,M} - \lambda_k| \leq C_k \max_{i \leq J_k} \|\mathcal{K}\phi_{i,k} - \hat{\mathcal{K}}_M\phi_{i,k}\| \leq C_* \|\mathcal{K}\phi_{i,k} - \hat{\mathcal{K}}_M\phi_{i,k}\|$$

where C_k is a finite constant that depends on λ_k, and $C_* := \max_{k \leq p} C_k$. Under Assumption 5(i), and by Lemmas 14 and 22, we have:

$$\max_{i \leq J_k} \|\mathcal{K}\phi_{i,k} - \hat{\mathcal{K}}\phi_{i,k}\| = O(\frac{(\log M)^{d-1}}{M})$$

Therefore, we have $\lambda_{k,M}, \lambda_k$ have the same sign for sufficiently large M for all $k \leq p$.

It then follows that $\delta_{M,p} := \min_{k \leq p}\{1 - \lambda_{k,M}\}$ is bounded away from zero for all sufficiently large M, say $\delta_{M,p} \geq C_\delta > 0$. Moreover, $\Delta_{M,p} := \max_{k \leq p}\{1 - \lambda_{k,M}\}$ is bounded above for all sufficiently large M, say $\Delta_{M,p} \leq C_\Delta$. Then, we have for sufficiently large M: $\frac{\Delta_{M,p}}{\delta_{M,p}} \leq \frac{C_\Delta}{C_\delta}$. Moreover, we can compare c_k with $c_{k,M}$:

$$\begin{aligned} |c_k - c_{k,M}| &\leq \frac{2}{k}\sum_{i=1}^{k} |\frac{\lambda_i - \lambda_{i,M}}{(1-\lambda_i)(1-\lambda_{i,M})}| \leq \frac{1}{\delta\delta_{M,p}}\frac{2}{k}\sum_{i=1}^{k}|\lambda_i - \lambda_{i,M}| \\ &\leq \frac{C_*}{\delta C_\delta}\frac{2}{k}\sum_{j=1}^{k}\max_{i \leq J_j}\|\mathcal{K}\phi_{i,j} - \hat{\mathcal{K}}\phi_{i,j}\| = O(\frac{(\log M)^{d-1}}{M}) \end{aligned}$$

where the first inequality used that $\lambda_i, \lambda_{i,M}$ have the same sign for sufficiently large M.

Note that:

$$|(c_k)^k - (c_{k,M})^k| = |c_k - c_{k,M}||\sum_{i=1}^{k}(c_k)^{k-i}(c_{k,M})^{i-1}|$$

By Lemma 9, there exists k_* such that $c_k < 0.9$ for $k \geq k_*$. Moreover, there exists $C_{c_{p^*}} < +\infty$ such that $c_k < C_{c_{p^*}}$ for $k < k_*$. We may assume $p \geq k_*$. Otherwise, we can only consider $k \leq k_*$ in the following proof. Since $|c_k - c_{k,M}| = O(\frac{(\log M)^{d-1}}{M})$ for $k \leq p$ for all sufficiently large M, we may choose M sufficiently large such that $c_{k,M} < 0.95$ for $p \geq k \geq k_*$ and $c_{k,M} < C_{c_{p^*}}$ for $k < k_*$. Then, for $p \geq k \geq k_*$ and sufficiently large M, we have:

$$|\sum_{i=1}^{k}(c_k)^{k-i}(c_{k,M})^{i-1}| \leq k \times 0.95^{k-1} < 8$$

For $k < k_*$ and sufficiently large M, we have:

$$|\sum_{i=1}^{k}(c_k)^{k-i}(c_{k,M})^{i-1}| \leq k_* \times C_{c_{p^*}}^{k_*-1} = O(1)$$

Therefore, we have for $k \leq p$ and sufficiently large M:

$$(c_{k,M})^k \leq (c_k)^k + |(c_k)^k - (c_{k,M})^k| = (c_k)^k + |c_k - c_{k,M}||\sum_{i=1}^{k}(c_k)^{k-i}(c_{k,M})^{i-1}| = O((c_k)^k + \frac{(\log M)^{d-1}}{M})$$

For Corollary 1, for sufficiently large M, we have for $k \leq p$:

$$(\tau_{k,M})^k = (\frac{\Delta_{M,p}}{\delta_{M,p}})^{\frac{3}{2}}(c_{k,M})^k \leq (\frac{C_\Delta}{C_\delta})^{\frac{3}{2}}(c_{k,M})^k = O((c_k)^k + \frac{(\log M)^{d-1}}{M})$$

Therefore, we have for $k \leq p$ and sufficiently large M:

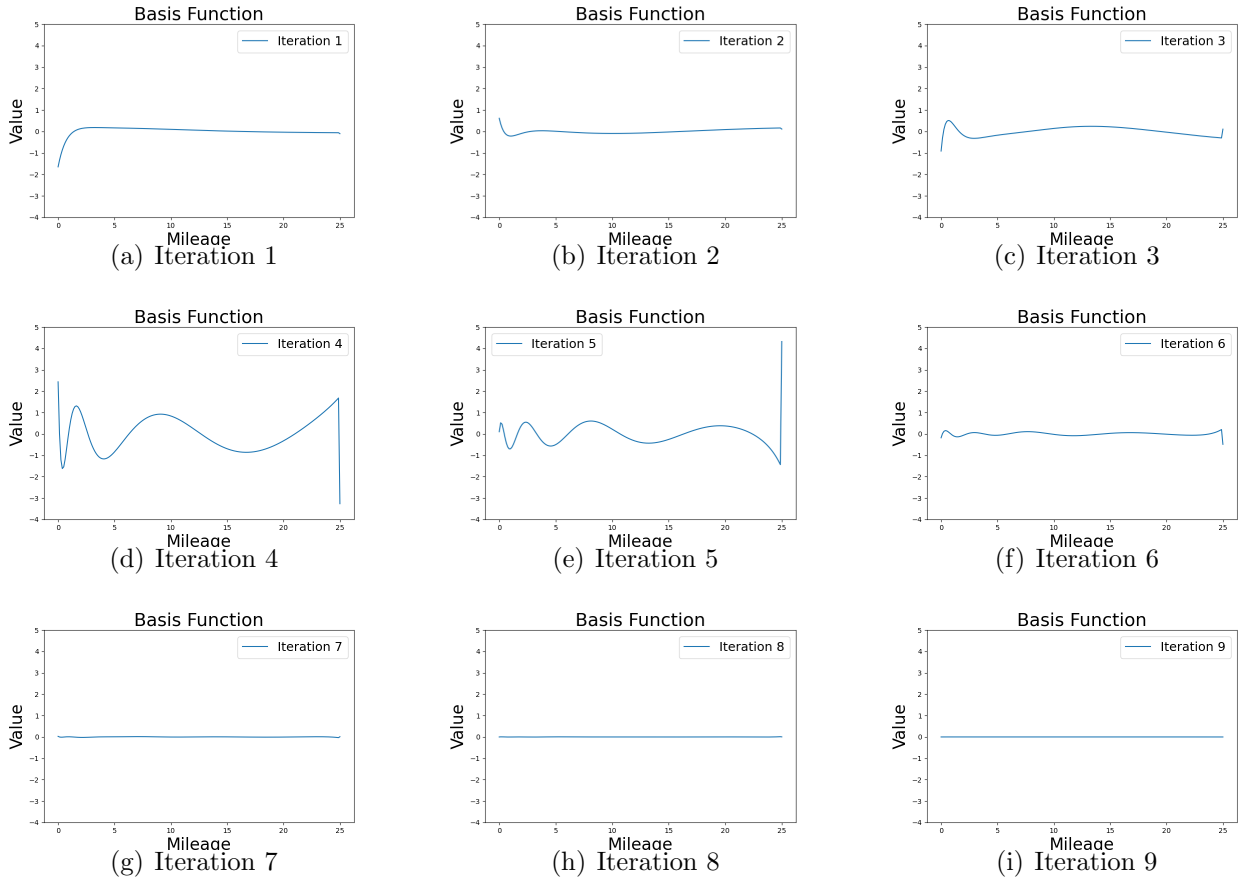
$$\|V_M - V_M^*\|_{\mathbb{M}} = O(\frac{(\log M)^{d-1}}{M} + (\tau_{k,M})^k) = O((c_k)^k + \frac{(\log M)^{d-1}}{M})$$

the proof is then similar to that of Theorem 5.

For regular grids, the proof is the similar with Assumption 5(i), Lemmas 14 and 22 replaced by Assumption 5(ii) and Lemma 18. Furthermore, the product of two functions in $\mathcal{W}_C^\alpha([0,1]^d)$ is in $\mathcal{W}_{C'}^\alpha([0,1]^d)$ for some C' by Lemma 24. □

B Online Appendix

B.1 Plots of Basis Functions in Section 5.1



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

(d) Iteration 4 (e) Iteration 5 (f) Iteration 6

(g) Iteration 7 (h) Iteration 8 (i) Iteration 9

Figure 9: Basis Functions

B.2 Details of Algorithms in Section 5.2

Algorithm 3 (Policy Iteration with Model-Adaptive Approach)**.**

1. At iteration i, given $C^i(x, j)$, update $p^{i+1}(j|x)$ by policy iteration:

   - At iteration i', given $p_{i'}(j|x)$, solve for $V_{i'}(x)$ by our **model-adaptive** method:

$$V_{i'}(x) = U_{C^i, p_{i'}}(x) + \beta \mathbb{E}_{C^i, p_{i'}}[V_{i'}(x')|x]$$

     where:

(i) $U_{C^i,p_{i'}}(x) = \sum_j p_{i'}(j|x)\,[U(C^i(x,j),I,j) + \omega_j - \log p_{i'}(j|x)]$.

(ii) $\mathbb{E}_{C^i,p_{i'}}[V_{i'}(x')|x] := \sum_j p_{i'}(j|x) \int f(x'|x,j,C^i(x,j))V_{i'}(x')dx'$.

- Then, the policy improvement is given by:

$$p_{i'+1}(j|x) = \frac{\exp(v_{i'}(x,j))}{\sum_j \exp(v_{i'}(x,j))}$$

where $v_{i'}(x,j) = U(C^i(x,j),I,j) + \beta \mathbb{E}_{C^i,p_{i'}}[V_{i'}(x')|x]$.

- Iterate until $p_{i'}(j|x)$ converges, and set $p^{i+1}(j|x) = p_{i'}(j|x)$ and $V^i(x) = V_{i'}(x)$.

2. Given $V^i(x)$, update $C^{i+1}(x,j)$ by:

$$C^{i+1}(x,j) := \underset{c_{min} \le c \le c_{max}}{\text{argmax}} \Big[U(c,I,j;\theta) + \omega_j + \beta \mathbb{E}[V^i(x')|x,c,j] \Big]$$

3. Iterate until $C^{i+1}(x,j)$ converges.[20]

Algorithm 4 (Value Iteration)**.**

1. At iteration i, given $C^i(x,j)$, update $V^{i+1}(x)$ by **value iteration**:

- At iteration i', given $V^{i'}(x)$, update $V^{i'+1}(x)$ by:

$$V_{i'+1}(x) = \log \sum_{j'} \exp \Big[U(C^i(x,j'),I,j') + \omega_{j'} + \beta \mathbb{E} \left[V_{i'}(x')|x, C^i(x,j'), j' \right] \Big]$$

- Iterate until $V_{i'+1}(x)$ converges, i.e., $\sup_x |V_{i'+1}(x) - V_{i'}(x)| \le 10^{-8}$.
- Set $V^{i+1}(x) = V_{i'+1}(x)$.

2. Given $V^{i+1}(x)$, update $C^{i+1}(x,j)$ by:

$$C^{i+1}(x,j) := \underset{c_{min} \le c \le c_{max}}{\text{argmax}} \Big[U(c,I,j;\theta) + \omega_j + \beta \mathbb{E}[V^{i+1}(x')|x,c,j] \Big]$$

3. Iterate until $C^{i+1}(x,j)$ converges, i.e., $\sup_{x,j} |C^{i+1}(x,j) - C^i(x,j)| = 0$.

Algorithm 5 (One-step Value Iteration)**.**

[20] The stopping rule for policy update, policy valuation, and consumption function are set to be $\|p^{i+1} - p^i\|_\infty \le 10^{-4}$, $\|r_k\|_\infty \le 10^{-8}$, and $\|C^{i+1} - C^i\|_\infty = 0$. We also normalize the utility function by $\frac{c}{I_{max}}$, $\frac{c^2}{I^2_{max}}$, and $\frac{I^2}{I^2_{max}}$.

1. At iteration i, given $C^i(x,j), V^i(x)$, update $V^{i+1}(j|x)$ by **one-step value iteration**:

$$V^{i+1}(x) = \log \sum_{j'} \exp\Big[U(C^i(x,j'),I,j') + \omega_{j'} + \beta \mathbb{E}\left[V^i(x')|x, C^i(x,j'), j'\right]\Big]$$

2. Given $V^{i+1}(x)$, update $C^{i+1}(x,j)$ by:

$$C^{i+1}(x,j) := \underset{c_{min}\le c\le c_{max}}{\text{argmax}} \Big[U(c,I,j;\theta) + \omega_j + \beta \mathbb{E}[V^{i+1}(x')|x,c,j]\Big]$$

3. Iterate until $C^{i+1}(x,j)$ and $V^{i+1}(x)$ converge, i.e., $\sup_{x,j} |C^{i+1}(x,j) - C^i(x,j)| = 0$ and $\sup_x |V^{i+1}(x) - V^i(x)| \le 10^{-8}$.

B.3 Adaptive MCMC Algorithm

This section presents the details of adaptive MCMC with vanishing adaptation used to estimate the dynamic parameters. The vanishing adaptation ensures that the current parameter draw depends less and less on recently sampled parameter values as the MCMC chain progresses Andrieu and Thoms [2008]. Algorithm 1 describes the algorithm (see Andrieu and Thoms [2008]). The priors are chosen to be a normal distribution with mean 0 and variance 100. As the inventory is unobserved, we simulate the inventory with $I_0 = 0$. We treat inventory as if it was observed and drop the first 30% periods to evaluate the (simulated) likelihood. We keep the acceptance rate around 0.3. The algorithm runs for 10,000 iterations, and the initial 8,000 are discarded as burn-in. Figure 10 suggests that the MCMC draws seem to stabilize after around 2,000 iterations. Therefore, the burn-in cutoff of 8,000 iterations is conservative.

Algorithm 1: Adaptive MCMC algorithm with vanishing adaptation

Input: θ_0, $\mu_0 = \theta_0$, $\Sigma_0 = \mathbf{I}_4$, $\lambda_0 = 2.38^2/4$, $\alpha^* = 0.3$, $\delta = 0.5$, $T = 10000$, and $t = 0$.
for $t = 0$ **to** T **do**
  Draw a candidate $\theta \sim \mathcal{N}(\theta_t, \lambda_t \Sigma_t)$;
  Solve the model with θ and compute the likelihood $Pr(data|\theta)$;
  Set $\theta_{t+1} = \theta$ with prob. $\alpha(\theta,\theta_t) := \min\left\{1, \frac{Pr(data|\theta)\pi(\theta)}{Pr(data|\theta_t)\pi(\theta_t)}\right\}$, else $\theta_{t+1} = \theta_t$;
  Compute $\gamma_t = \frac{1}{(1+t)^\delta}$ and update $\lambda_{t+1} = \exp\{\gamma_t(\alpha(\theta,\theta_t) - \alpha^*)\}\lambda_t$;
  Update $\mu_{t+1} = \mu_t + \gamma_t(\theta_{t+1} - \mu_t)$;
  Update $\Sigma_{t+1} = \Sigma_t + \gamma_t\{(\theta_{t+1} - \mu_t)(\theta_{t+1} - \mu_t)^T - \Sigma_t\}$;
end
Output: The sequence $\{\theta_t\}_{t=1}^T$

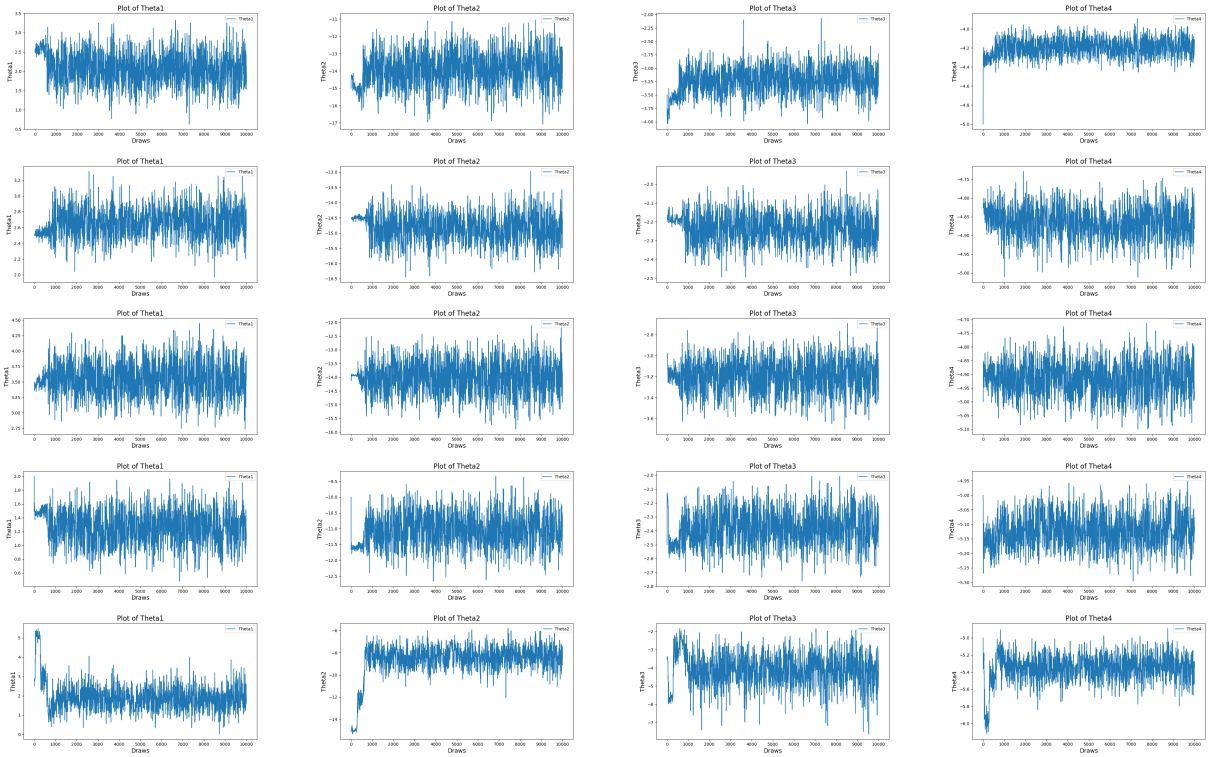


Figure 10: Plot of MCMC draws for Parameters of Different Household Sizes (Rows 1-5 correspond to Household Sizes 1-5)

B.4 Supporting Lemmas

Lemma 3. *$L^2(\mathbb{X}, \nu_{Leb})$ is a separable Hilbert space.*

Proof. First, suppose $\mathbb{X} = [0,1]^d$. By Durrett [2019] exercise 1.1.3, the Borel subsets of $\mathbb{R}^d$, $\mathcal{R}^d$ is countably generated. By Tao [2011] exercise 1.4.12, the restriction of the σ-algebra $\mathcal{R}^d \restriction_{[0,1]^d}$ is a σ-algebra on the subspace $[0,1]^d \subset \mathbb{R}^d$. Then, the σ-algebra of $[0,1]^d$, $\mathcal{R}^d \restriction_{[0,1]^d}$, can also be countably generated. The measure space $(\mathbb{X}, \mathcal{R}^d \restriction_{[0,1]^d}, \nu_{Leb})$ is a separable measure space. By Tao [2022] exercise 1.3.9, $L^2(\mathbb{X}, \nu_{Leb})$ is a separable space. By Rudin [1987] Example 4.5(b), $L^2(\mathbb{X}, \nu_{Leb})$ is a Hilbert space with the inner product $\langle h, g \rangle$. Thus, $L^2(\mathbb{X}, \nu_{Leb})$ is a separable Hilbert space. If $\mathbb{X}$ has discrete support, then $L^2(\mathbb{X}, \nu_{Leb})$ is also a separable Hilbert space as the σ-algebra can be countably generated. □

Lemma 4. *Under Assumption 1, $(\mathcal{I} - \mathcal{T})V = u$ has a unique solution on $L^2(\mathbb{X})$.*

Proof. Under Assumption 1, the norms $\|\cdot\|_\mu$ and $\|\cdot\|$ are equivalent, i.e., there exist C_1, $C_2 > 0$ such that $C_1\|\cdot\| \leq \|\cdot\|_\mu \leq C_2\|\cdot\|$. Therefore, the equation has a solution V^*. We prove the uniqueness by contradiction. Suppose it has two solutions on $L^2(\mathbb{X})$. That

is, there exist V_1, $V_2 \in L^2(\mathbb{X})$ such that $\|V_1 - V_2\| > 0$, $V_1 = u + \mathcal{T}V_1$, and $V_2 = u + \mathcal{T}V_2$. Then, we have $0 < C_1\|V_1 - V_2\| \leq \|V_1 - V_2\|_\mu \leq C_2\|V_1 - V_2\| < +\infty$. Moreover, we have V_1, $V_2 \in L^2(\mathbb{X}, \mu)$ since $\|\cdot\|_\mu$ and $\|\cdot\|$ are equivalent. This means the equation also has two solutions on $L^2(\mathbb{X}, \mu)$, contradiction. □

Lemma 5. *The adjoint operator $\mathcal{T}^*$ with respect to the inner product $\langle \cdot, \cdot \rangle$ is given by:*

$$\mathcal{T}^* V(x) = \beta \int V(x') f(x|x') dx'$$

Proof. For $\phi, \psi \in L^2(\mathbb{X})$, we have:

$$\langle \phi, \mathcal{T}^* \psi \rangle = \int \left[\int \beta f(x|x') \psi(x') dx' \right] \phi(x) dx = \int \beta f(x|x') \phi(x) \psi(x') dx dx' = \langle \mathcal{T}\phi, \psi \rangle$$

□

Lemma 6. *Under Assumption 1, we have:*

(i) *$\mathcal{T}$ maps $L^2(\mathbb{X})$ to $L^2(\mathbb{X})$.*

(ii) *$\mathcal{T}$ is a Hilbert-Schmidt operator and thus compact.*

Proof. 1. Let $h \in L^2(\mathbb{X})$. Then, $\|\mathcal{T}h\|^2 = \beta^2 \int (\int f(x'|x) h(x') dx')^2 dx \leq \beta^2 C_f^2 \|h\|^2 < +\infty$

2. Note that: $\|\mathcal{T}\|_{HS}^2 := \beta^2 \int f^2(x'|x) dx' dx \leq \beta^2 C_f^2 < +\infty$. Therefore, $\mathcal{T}$ is a Hilbert-Schmidt operator. Moreover, Hilbert-Schmidt operators are compact by Carrasco et al. [2007] Theorem 2.32.

□

Lemma 7. *Under Assumption 1, we have:*

(i) *$\mathcal{K}$ maps $L^2(\mathbb{X})$ to $L^2(\mathbb{X})$.*

(ii) *$\mathcal{K}$ is a self-adjoint operator.*

(iii) *$\mathcal{K}$ is a Hilbert-Schmidt operator and thus compact.*

(iv) *$(\mathcal{I} - \mathcal{K})$ is a positive definite operator.*

Proof. (i) Since $\mathcal{T}$ maps $L^2(\mathbb{X})$ to itself, $\mathcal{T}^*$ also does. Therefore, $\mathcal{K}$ maps $L^2(\mathbb{X})$ to $L^2(\mathbb{X})$.

(ii) Note that $\mathcal{K}^* = (\mathcal{T} + \mathcal{T}^* - \mathcal{T}\mathcal{T}^*)^* = \mathcal{T}^* + \mathcal{T} - \mathcal{T}\mathcal{T}^*$.

(iii) Note that: for any $a, b, c \in \mathbb{R}$, we have $(a+b+c)^2 \leq 2(a^2+b^2+c^2)$, which implies $\|\mathcal{K}\|_{HS}^2 \leq 2(\|\mathcal{T}\|_{HS}^2 + \|\mathcal{T}^*\|_{HS}^2 + \|\mathcal{T}\mathcal{T}^*\|_{HS}^2)$. By Conway [2019] page 267, we have: $\|\mathcal{T}\|_{HS} = \|\mathcal{T}^*\|_{HS} < +\infty$ and $\|\mathcal{T}\mathcal{T}^*\|_{HS} \leq \|\mathcal{T}\|_{op}\|\mathcal{T}^*\|_{HS} < +\infty$. Therefore, $\|\mathcal{K}\|_{HS} < +\infty$, i.e., $\mathcal{K}$ is a Hilbert-Schmidt operator. Hilbert-Schmidt operators are compact by Carrasco et al. [2007] Theorem 2.32.

(iv) Note that: $\langle(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^*)y, y\rangle = \langle(\mathcal{I}-\mathcal{T}^*)y, (\mathcal{I}-\mathcal{T}^*)y\rangle = \|(\mathcal{I}-\mathcal{T}^*)y\|^2 \geq 0$. If $\|(\mathcal{I}-\mathcal{T}^*)y\| = 0$, then $(\mathcal{I}-\mathcal{T}^*)y = 0$. As showed in the proof of Theorem 1, $(\mathcal{I}-\mathcal{T}^*)$ is invertible on $L^2(\mathbb{X})$, we have $y = 0$. Therefore, $(\mathcal{I}-\mathcal{T})(\mathcal{I}-\mathcal{T}^*)$ is positive definite.

□

B.4.1 Supporting Lemmas for Theorem 2

Lemma 8. *The sequence* $\{r_i\}_{i\geq 1}, \{s_i\}_{i\geq 1}$ *generated by* (3.2) *satisfies:*

(i) $\langle r_i, r_j\rangle = 0$, *for* $i < j$.

(ii) $\langle r_i, (\mathcal{I}-\mathcal{K})s_j\rangle = 0$, *for* $i \neq j$, $i \neq j+1$.

(iii) $\langle s_i, (\mathcal{I}-\mathcal{K})s_j\rangle = 0$ *for* $i < j$.

(iv) $\langle r_i, (\mathcal{I}-\mathcal{K})s_i\rangle = \langle s_i, (\mathcal{I}-\mathcal{K})s_i\rangle$.

Proof. Without loss of generality we assume the algorithm does not stop, i.e., $\|r_i\| > 0$ for all $i \leq k$. Therefore, $\alpha_i > 0$ for $i \leq k-1$.

By induction, it is easy to show:

$$s_k = r_k + \frac{\|r_k\|^2}{\|r_{k-1}\|^2}s_{k-1} = \cdots = \|r_k\|^2 \sum_{i=0}^{k} \frac{r_i}{\|r_i\|^2}$$

Note that:

$$\begin{aligned} r_k &= u - (\mathcal{I}-\mathcal{K})y_k \\ &= u - (\mathcal{I}-\mathcal{K})(y_{k-1} + \alpha_{k-1}s_{k-1}) \\ &= u - (\mathcal{I}-\mathcal{K})y_{k-1} - \alpha_{k-1}(\mathcal{I}-\mathcal{K})s_{k-1} \\ &= r_{k-1} - \alpha_{k-1}(\mathcal{I}-\mathcal{K})s_{k-1} \end{aligned}$$

Prove by induction, it holds for for r_0, r_1, s_0, s_1. Now, suppose it holds for $r_0, \cdots, r_k, s_0, \cdots, s_k$. For $k+1$, it suffices to show:

$$\langle r_i, r_{k+1}\rangle = 0, \text{ for } i < k+1 \tag{B.1}$$
$$\langle r_{k+1}, (\mathcal{I}-\mathcal{K})s_j\rangle = 0, \text{ for } j < k \tag{B.2}$$
$$\langle r_i, (\mathcal{I}-\mathcal{K})s_{k+1}\rangle = 0, \text{ for } i < k+1 \tag{B.3}$$
$$\langle s_i, (\mathcal{I}-\mathcal{K})s_{k+1}\rangle = 0, \text{ for } i < k+1 \tag{B.4}$$
$$\langle r_{k+1}, (\mathcal{I}-\mathcal{K})s_{k+1}\rangle = \langle s_{k+1}, (\mathcal{I}-\mathcal{K})s_{k+1}\rangle \tag{B.5}$$

1. For (B.1), if $i < k$,

$$\langle r_i, r_{k+1}\rangle = \langle r_i, r_k - \alpha_k(\mathcal{I}-\mathcal{K})s_k\rangle = \underbrace{\langle r_i, r_k\rangle}_{=0 \text{ by 8(i) with } j=k} - \alpha_k \underbrace{\langle r_i, (\mathcal{I}-\mathcal{K})s_k\rangle}_{=0 \text{ by 8(ii) with } j=k} = 0$$

   If $i = k$,

$$\langle r_i, r_{k+1}\rangle = \langle r_k, r_k - \alpha_k(\mathcal{I}-\mathcal{K})s_k\rangle = \|r_k\|^2 - \frac{\|r_k\|^2}{\langle(\mathcal{I}-\mathcal{K})s_k, s_k\rangle} \underbrace{\langle r_k, (\mathcal{I}-\mathcal{K})s_k\rangle}_{=\langle s_k, (\mathcal{I}-\mathcal{K})s_k\rangle \text{ by 8(iv) with } i=k} = 0$$

2. For (B.2), and $j < k$, by (B.1)

$$\begin{aligned} 0 &= \langle r_{k+1}, r_{j+1}\rangle \\ &= \langle r_{k+1}, r_j - \alpha_j(\mathcal{I}-\mathcal{K})s_j\rangle \\ &= \langle r_{k+1}, r_j\rangle - \alpha_j\langle r_{k+1}, (\mathcal{I}-\mathcal{K})s_j\rangle \\ &= -\alpha_j\langle r_{k+1}, (\mathcal{I}-\mathcal{K})s_j\rangle \end{aligned}$$

   Since $\alpha_j > 0$, we have $\langle r_{k+1}, (\mathcal{I}-\mathcal{K})s_j\rangle = 0$.

3. For (B.4) and $i < k+1$, note that by 8(i) with $j = i+1$

$$\langle r_{i+1}, s_{k+1}\rangle = \langle r_{i+1}, \|r_{k+1}\|^2 \sum_{q=0}^{k+1} \frac{r_q}{\|r_q\|^2}\rangle = \|r_{k+1}\|^2$$

   Similarly, $\langle r_i, s_{k+1}\rangle = \|r_{k+1}\|^2$. Also,

$$\begin{aligned} &\langle r_{i+1}, s_{k+1}\rangle \\ &= \langle r_i - \alpha_i(\mathcal{I}-\mathcal{K})s_i, s_{k+1}\rangle \end{aligned}$$

$$
\begin{aligned}
&= \langle r_i, s_{k+1}\rangle - \alpha_i \langle (\mathcal{I}-\mathcal{K}) s_i, s_{k+1}\rangle \\
&= \langle r_i, s_{k+1}\rangle - \alpha_i \langle s_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle \\
&= \|r_{k+1}\|^2 - \alpha_i \langle s_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle
\end{aligned}
$$

Since $\alpha_i > 0$, we have $\langle s_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle = 0$ for $i < k+1$.

4. For (B.5),

$$
\begin{aligned}
\langle s_{k+1}, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle &= \langle r_{k+1} + \beta_k s_k, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle \\
&= \langle r_{k+1}, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle + \beta_k \underbrace{\langle s_k, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle}_{=0 \text{ by (B.4) with } i=k} \\
&= \langle r_{k+1}, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle
\end{aligned}
$$

5. For (B.3) and $i < k+1$,

$$
\begin{aligned}
0 &= \langle s_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle \\
&= \langle r_i + \beta_{i-1} s_{i-1}, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle \\
&= \langle r_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle + \beta_{i-1} \underbrace{\langle s_{i-1}, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle}_{=0 \text{ by (B.4) with } i=i-1} \\
&= \langle r_i, (\mathcal{I}-\mathcal{K}) s_{k+1}\rangle
\end{aligned}
$$

□

B.4.2 Supporting Lemmas for Theorem 3

Lemma 9. *Under Assumption 1, let* $\tau_k := (\frac{\Delta}{\delta})^{\frac{3}{2k}} (\frac{2}{k} \sum_{j=1}^{k} \frac{|\lambda_j|}{1-\lambda_j}) \to 0$ *as* $k \to +\infty$. *Note that, we have* $(\tau_k)^k = (\frac{\Delta}{\delta})^{\frac{3}{2}} (c_k)^k$. *Moreover, we have:*

(i) c_k *goes to zero no faster than* $\frac{1}{k}$ *and no slower than* $\frac{1}{\sqrt{k}}$*:*

$$
\frac{2}{k} \frac{|\lambda_1|}{1-\lambda_1} \leq c_k \leq \frac{2\|\mathcal{K}\|_{HS}}{\delta} \frac{1}{\sqrt{k}}
$$

where $\|\mathcal{K}\|_{HS} < +\infty$ *is the Hilbert-Schmidt norm of* $\mathcal{K}$.

(ii) If $f(x'|x)$ *has continuous partial derivatives of order up to* l*, then*

$$
c_k \leq \frac{2C(l)}{k}
$$

where $C(l)$ is a constant depending on l.

Proof. 1. The lower bound on c_k is given by: $\frac{2}{k}\frac{|\lambda_1|}{1-\lambda_1} \leq c_k$.

2. By the Cauchy–Schwarz inequality, we have:

$$\frac{2}{k}\sum_{j=1}^{k}\frac{|\lambda_j|}{1-\lambda_j} \leq \frac{2}{k}\sqrt{(\sum_{j=1}^{k}|\lambda_j|^2)(\sum_{j=1}^{k}\frac{1}{(1-\lambda_j)^2})} \leq \frac{2}{k}\sqrt{(\sum_{j=1}^{k}|\lambda_j|^2)(\sum_{j=1}^{k}\frac{1}{\delta^2})} \leq \frac{2\|\mathcal{K}\|_{HS}}{\delta\sqrt{k}}$$

3. The kernel of $\mathcal{K}$ is $\beta f(x'|x) + \beta f(x|x') - \beta^2 \int f(x|y)f(x'|y)dy$, which is symmetric in x and x'. Then, Flores [1993] Theorem 3 applies.

□

B.4.3 Supporting Lemmas for Theorem 4

Lemma 10. *Under Assumption 2, we have:*

$$\frac{C_{td,1}}{1+\beta}k^{-\frac{\alpha}{d}} \leq \|V_k^{td} - V^*\|_\mu \leq \frac{C_{td,2}}{1-\beta}k^{-\frac{\alpha}{d}}$$

Proof. Note that $u + \mathcal{T}V_k^{td} = \Pi_{\mathbb{S}_k}(u + \mathcal{T}V_k^{td})$ and $\Pi_{\mathbb{S}_k}$ is non-expansive, i.e., $\|\Pi_{\mathbb{S}_k} f\|_\mu \leq \|f\|_\mu$, we have:

$$\begin{aligned}
\|V_k^{td} - V^*\|_\mu &= \|V_k^{td} - \Pi_{\mathbb{S}_k}V^* + \Pi_{\mathbb{S}_k}V^* - V^*\|_\mu \\
&\leq \|V_k^{td} - \Pi_{\mathbb{S}_k}V^*\|_\mu + \|\Pi_{\mathbb{S}_k}V^* - V^*\|_\mu \\
&= \|\Pi_{\mathbb{S}_k}(u + \mathcal{T}V_k^{td}) - \Pi_{\mathbb{S}_k}V^*\|_\mu + \|\Pi_{\mathbb{S}_k}V^* - V^*\|_\mu \\
&\leq \|u + \mathcal{T}V_k^{td} - V^*\|_\mu + C_{td,2}k^{-\frac{\alpha}{d}} \\
&= \|u + \mathcal{T}V_k^{td} - (u + \mathcal{T}V^*)\|_\mu + C_{td,2}k^{-\frac{\alpha}{d}} \\
&\leq \beta\|V_k^{td} - V^*\|_\mu + C_{td,2}k^{-\frac{\alpha}{d}}
\end{aligned}$$

Thus, $\|V_k^{td} - V^*\| \leq \frac{C_{td,2}}{1-\beta}k^{-\frac{\alpha}{d}}$. Similarly:

$$\begin{aligned}
\|V_k^{td} - V^*\|_\mu &= \|V_k^{td} - \Pi_{\mathbb{S}_k}V^* + \Pi_{\mathbb{S}_k}V^* - V^*\|_\mu \\
&\geq \|\Pi_{\mathbb{S}_k}V^* - V^*\|_\mu - \|V_k^{td} - \Pi_{\mathbb{S}_k}V^*\|_\mu \\
&= \|\Pi_{\mathbb{S}_k}V^* - V^*\|_\mu - \|\Pi_{\mathbb{S}_k}(u + \mathcal{T}V_k^{td}) - \Pi_{\mathbb{S}_k}V^*\|_\mu
\end{aligned}$$

Therefore:

$$\begin{aligned}\|\Pi_{\mathbb{S}_k} V^* - V^*\|_\mu &\leq \|V_k^{td} - V^*\|_\mu + \|\Pi_{\mathbb{S}_k}(u + \mathcal{T}V_k^{td}) - \Pi_{\mathbb{S}_k} V^*\|_\mu \\ &\leq \|V_k^{td} - V^*\|_\mu + \|(u + \mathcal{T}V_k^{td}) - V^*\|_\mu \\ &= \|V_k^{td} - V^*\|_\mu + \|(u + \mathcal{T}V_k^{td}) - (u + \mathcal{T}V^*)\|_\mu \\ &\leq \|V_k^{td} - V^*\|_\mu + \beta\|V_k^{td} - V^*\|_\mu\end{aligned}$$

Thus, $\|V_k^{td} - V^*\|_\mu \geq \frac{C_{td,1}}{1+\beta} k^{-\frac{\alpha}{d}}$. □

B.4.4 Supporting Lemmas for Theorem 5

Lemma 11. *Under Assumption 1,* $\Gamma V(x) := u(x) + \mathcal{T}V(x)$ *maps* $\mathcal{V} := \{V | \sup_x |V(x)| \leq \frac{C_u}{1-\beta}\}$ *to itself. Moreover,* $V^* \in \mathcal{V}$.

Proof. Note that $\sup_x |V^*(x)| \leq \sum_{i=0}^{+\infty} \beta^i C_u = \frac{C_u}{1-\beta}$, i.e., $V^* \in \mathcal{V}$. For any $V \in \mathcal{V}$ and x, we have:

$$|u(x) + \mathcal{T}V(x)| \leq C_u + \frac{\beta}{1-\beta} C_u = \frac{C_u}{1-\beta}$$

Therefore, Γ maps $\mathcal{V}$ to itself. □

Lemma 12. *Let* V_M *be the solution to* $V_M = u_M + \hat{\mathcal{T}}_M V_M$ *and* V_M^* *the* $|\mathbb{M}|$*-vector where the* i*-th element is* $V_{i,M}^* := V^*(x_i)$. *Denote* $(\mathcal{T}V^*)_M$ *the* $|\mathbb{M}|$*-vector where* $(\mathcal{T}V^*)_{i,M} := \mathcal{T}V^*(x_i)$ *and* $\|\cdot\|_\mathbb{M}$ *the sup-norm. Then, we have:*

$$\|V_M - V_M^*\|_\mathbb{M} \leq \frac{1}{1-\beta} \|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_\mathbb{M}$$

Proof. Note that $V_M^* = u_M + (\mathcal{T}V^*)_M$. By construction, $\hat{\mathcal{T}}_M$ is a β-contraction with respect to the sup-norm $\|\cdot\|_\mathbb{M}$. Thus,

$$\begin{aligned}\|V_M - V_M^*\|_\mathbb{M} &= \|\hat{\mathcal{T}}_M V_M + u_M - (\mathcal{T}V^*)_M - u_M\|_\mathbb{M} \\ &= \|\hat{\mathcal{T}}_M V_M - \hat{\mathcal{T}}_M V_M^* + \hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_\mathbb{M} \\ &\leq \beta\|V_M - V_M^*\|_\mathbb{M} + \|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_\mathbb{M}\end{aligned}$$

Thus, we have:

$$\|V_M - V_M^*\|_\mathbb{M} \leq \frac{1}{1-\beta} \|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_\mathbb{M}$$

□

Next, we prove the following lemmas for low-discrepancy grids part. The Quasi-Monte Carlo requires the higher order smoothness. We first prove the following lemmas about the smoothness of functions:

Lemma 13. *Under Assumption 3(i), $V^*(x) \in \mathcal{HK}_{C_{V^*}}$ for some C_{V^*}.*

Proof. First, we have the mixed partial derivatives of $D^m \Gamma V(x_m : 1_{-m})$ exists and is continuous for $V \in \mathcal{V}$ and all $m \in 1 : d$. By Lemma 11, $V^* \in \mathcal{V}$, it suffices to show that $\Gamma V(x)$ maps $\mathcal{V}$ to $\mathcal{HK}_{C_{V^*}}$ for some C_{V^*}. We have:

$$
\begin{aligned}
V_{HK}(\Gamma V) &= \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m \Gamma V(x_m : 1_{-m})| dx_m \\
&= \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m (u(x_m : 1_{-m}) + \beta \int_{x'} f(x'|x_m : 1_{-m}) V(x') dx')| dx_m \\
&\leq \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m u(x_m : 1_{-m})| dx_m + \beta \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} \int_{x'} |D^m f(x'|x_m : 1_{-m})||V(x')| dx' dx_m \\
&\leq \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m u(x_m : 1_{-m})| dx_m + \frac{\beta C_u}{1-\beta} \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} \int_{x'} |D^m f(x'|x_m : 1_{-m})| dx' dx_m \\
&= \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m u(x_m : 1_{-m})| dx_m + \frac{\beta C_u}{1-\beta} \int_{x'} V_{HK}(f(x'|\cdot)) dx' \\
&\leq V_{HK}(u) + \frac{\beta C_u}{1-\beta} \sup_{x'} V_{HK}(f(x'|\cdot)) \\
&\leq C + \frac{\beta C_u}{1-\beta} C := C_{V^*} < +\infty
\end{aligned}
$$

□

Lemma 14. *Under Assumption 3(i) and 3(ii), $K(\cdot|x) \in \mathcal{HK}_{C_K}$ for any x and some C_K.*

Proof. Recall that: $K(x'|x) = \beta f(x'|x) + \beta f(x|x') - \beta^2 \int f(x|y) f(x'|y) dy$. Therefore,

$$
\begin{aligned}
V_{HK}(K(\cdot|x)) &= \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m K(x_m : 1_{-m}|x)| dx_m \\
&\leq \beta \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m f(x_m : 1_{-m}|x)| dx_m + \beta \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m f(x|x_m : 1_{-m})| dx_m \\
&\qquad + \beta^2 \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} \int |f(x|y) D^m f(x_m : 1_{-m}|y)| dy dx_m \\
&\leq \beta V_{HK}(f(\cdot|x)) + \beta V_{HK}(f(x|\cdot)) + \beta^2 C_f \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} \int |D^m f(x_m : 1_{-m}|y)| dy dx_m
\end{aligned}
$$

$$
\begin{aligned}
&\leq \beta C + \beta C + \beta^2 C_f \int \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m f(x_m : 1_{-m}|y)| dx_m dy \\
&\leq 2\beta C + \beta^2 C_f \int V_{HK}(f(\cdot|y)) dy \\
&\leq 2\beta C + \beta^2 C_f \sup_y V_{HK}(f(\cdot|y)) \\
&\leq 2\beta C + \beta^2 C_f C := C_{\mathcal{K}} < +\infty
\end{aligned}
$$

As the bound holds for any x, we have $K(\cdot|x) \in \mathcal{HK}_{C_{\mathcal{K}}}$ for any x. □

Lemma 15. *Under Assumption 3(i) and 3(ii), $f(\cdot|x)V^*(\cdot)$ and $K^2(\cdot|x) \in \mathcal{HK}_C$ for any x.*

Proof. Combining Lemma 22 and Lemma 14 proves the result. □

Lemma 16. *Let Assumption 3 holds. If a low-discrepancy grid is used, then:*

$$
\|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_{\mathbb{M}} = O(\frac{(\log M)^{d-1}}{M})
$$

Proof. Let $C_V := \frac{C_u}{1-\beta}$ and $\tilde{\hat{\mathcal{T}}}_M V^*$ be the $|\mathbb{M}|$-vector where the i-th element is:

$$
(\tilde{\hat{\mathcal{T}}}_M V^*)_{i,M} := \frac{\beta}{M} \sum_j f(x_j|x_i) V^*(x_j)
$$

By the definition of $\hat{\mathcal{T}}_M$, we have $(\hat{\mathcal{T}}_M V^*)_{i,M} = \beta \sum_j \frac{f(x_j|x_i)V^*(x_j)}{\sum_j f(x_j|x_i)}$. Then, we have:

$$
|(\hat{\mathcal{T}}_M V_M^*)_i - (\mathcal{T}V^*)_{i,M}| \leq |(\hat{\mathcal{T}}_M V_M^*)_i - (\tilde{\hat{\mathcal{T}}}_M V^*)_{i,M}| + |(\tilde{\hat{\mathcal{T}}}_M V^*)_{i,M} - (\mathcal{T}V^*)_{i,M}|
$$

For the first part, under Assumption 3(i), we have:

$$
\begin{aligned}
|(\hat{\mathcal{T}}_M V_M^*)_i - (\tilde{\hat{\mathcal{T}}}_M V^*)_{i,M}| &= |\beta \sum_j \frac{f(x_j|x)V^*(x_j)}{\sum_j f(x_j|x_i)} - \frac{\beta}{M} \sum_j f(x_j|x_i)V^*(x_j)| \\
&\leq \beta \sum_j \frac{1}{M} f(x_j|x_i)|V^*(x_j)||1 - \frac{M}{\sum_j f(x_j|x_i)}| \\
&\leq \beta C_V \sum_j \frac{1}{M} f(x_j|x_i)|1 - \frac{M}{\sum_j f(x_j|x_i)}| \\
&= \beta C_V \sum_j |1 - \frac{1}{M} f(x_j|x_i)| = O(\frac{(\log M)^{d-1}}{M})
\end{aligned}
$$

where the last equality follows from $f(\cdot|x) \in \mathcal{HK}_C$ for any x and Lemma 2.

For the second part, by Lemma 15 and Lemma 2, we have:

$$\beta|(\tilde{\hat{\mathcal{T}}}_M V^*)_{i,M} - (\mathcal{T}V^*)_{i,M}| = \beta\left|\frac{1}{M}\sum_j f(x_j|x_i)V^*(x_j) - \int f(x'|x_i)V^*(x')dx'\right| = O(\frac{(\log M)^{d-1}}{M})$$

Combining both gives: $\|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_{\mathbb{M}} = O(\frac{(\log M)^{d-1}}{M})$. □

Then, we prove the following lemmas similar to that of the low-discrepancy grid part.

Lemma 17. *Under Assumption 3(iii), $V^* \in \mathcal{W}^{\alpha}_{C'}([0,1]^d)$ for some C'.*

Proof. By Lemma 11, $V^* \in \mathcal{V}$, it suffices to show that $\Gamma V(x)$ maps $\mathcal{V}$ to $\mathcal{W}^{\alpha}_{C'}([0,1]^d)$. Note that for $|\boldsymbol{k}| < \alpha$, we have:

$$\begin{aligned}|D^{\boldsymbol{k}}\Gamma V(x)| &= |D^{\boldsymbol{k}}(u(x) + \beta\int f(x'|x)V(x')dx')| \\ &\leq \sup_x |D^{\boldsymbol{k}}(u(x))| + \beta\sup_x \int |D^{\boldsymbol{k}}f(x'|x)||V(x')|dx' \leq C + \beta C_V C\end{aligned}$$

Moreover, for $|\boldsymbol{k}| = \lfloor\alpha\rfloor$, we have:

$$\frac{|D^{\boldsymbol{k}}\Gamma V(x) - D^{\boldsymbol{k}}\Gamma V(y)|}{\|x-x'\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \leq \frac{|D^{\boldsymbol{k}}(u(x)-u(y))|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} + \beta\frac{\int |D^{\boldsymbol{k}}f(x'|x) - D^{\boldsymbol{k}}f(x'|y)||V(x')|dx'}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \leq C + \beta C_V C$$

Therefore, $V^* \in \mathcal{W}^{\alpha}_{C'}([0,1]^d)$ where $C' := 2\max\{C + \beta C_V C, C_V\}$. □

Lemma 18. *Under Assumption 3(iii) and 3(iv), $f(\cdot|x)V^*(\cdot) \in \mathcal{W}^{\alpha}_{C'}([0,1]^d)$ for any x and some C'.*

Proof. Since $f(\cdot|x), V^*(\cdot) \in \mathcal{W}^{\alpha}_{C}([0,1]^d)$ for any x, Lemma 24 applies. □

Lemma 19. *Under Assumption 1(v), 3(iii) and 3(iv), $K(\cdot|x), K^2(\cdot|x) \in \mathcal{W}^{\alpha}_{C'}([0,1]^d)$ for any x and some C'.*

Proof. Recall that $K(x'|x) = \beta f(x'|x) + \beta f(x|x') - \beta^2\int f(x|y)f(x'|y)dy$. Therefore, for any x and $|\boldsymbol{k}| < \alpha$, we have:

$$\begin{aligned}|D^{\boldsymbol{k}}K(x'|x)| &= |\beta D^{\boldsymbol{k}}f(x'|x) + \beta D^{\boldsymbol{k}}f(x|x') - \beta^2\int f(x|y)D^{\boldsymbol{k}}f(x'|y)dy| \\ &\leq \beta C + \beta C + \beta^2 C\int f(x|y)dy \\ &\leq 2\beta C + \beta^2 C_f\end{aligned}$$

Moreover, for any x and $|\boldsymbol{k}| = \lfloor \alpha \rfloor$, we have:

$$\begin{aligned}\frac{|D^{\boldsymbol{k}}K(x'|x) - D^{\boldsymbol{k}}K(x''|x)|}{\|x' - y''\|_\infty^{\alpha - \lfloor\alpha\rfloor}} &\leq \beta \frac{|D^{\boldsymbol{k}}f(x'|x) - D^{\boldsymbol{k}}f(x''|x)|}{\|x' - x''\|_\infty^{\alpha-\lfloor\alpha\rfloor}} + \beta \frac{|D^{\boldsymbol{k}}f(x|x') - D^{\boldsymbol{k}}f(x|x'')|}{\|x' - x''\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \\ &\quad + \beta^2 \frac{\int f(x|y)|D^{\boldsymbol{k}}f(x'|y) - D^{\boldsymbol{k}}f(x''|y)|dy}{\|x' - x''\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \\ &\leq 2\beta C + \beta^2 C_f C\end{aligned}$$

Therefore, $K(\cdot|x) \in \mathcal{W}^\alpha_{C'}([0,1]^d)$ for any x and some C'. Moreover, $K^2(\cdot|x) \in \mathcal{W}^\alpha_{C'}([0,1]^d)$ for any x by Lemma 24. □

Lemma 20. *Suppose Assumption 3 holds. If a regular grid is used, then:*

$$\|\hat{\mathcal{T}}_M V_M^* - (\mathcal{T}V^*)_M\|_{\mathbb{M}} = O(M^{-\frac{\alpha}{d}})$$

Proof. The overall structure remains the same as the proof presented in Lemma 16, except that the term $O(\frac{(\log M)^{d-1}}{M})$ is substituted with $O(M^{-\frac{\alpha}{d}})$. Furthermore, the conditions for low-discrepancy grids are replaced by those for regular grids. □

Lemma 21. *Assume the mixed partial derivative $D^m h(x_m : 1_{-m})$ exists and is continuous on $[0,1]^{|m|}$ for all $m \subset 1:d$, then the Hardy-Krause variation of h is given by:*

$$V_{HK}(h) = \sum_{m \neq \emptyset} \int_{[0,1]^{|m|}} |D^m h(x_m : 1_{-m})| \, dx_u \tag{B.6}$$

where $|m|$ is the cardinality of m.

Proof. By Owen [2005] Proposition 14, the total variation of $h(x_m : 1_{-m})$ in Vitali's sense satisfies:

$$V_{[0,1]^{|m|}} h(\cdot : 1_{-m}) = \int_{[0,1]^{|m|}} |D^m h(x_m : 1_{-m})| \, dx_m$$

when $D^m h(x_m : 1_{-m})$ exists and is continuous. By Owen [2005] definition 2, the Hardy-Krause variation of h is given by:

$$V_{HK}(h) = \sum_{m \neq \emptyset} V_{[0,1]^{|m|}} h(\cdot : 1_{-m})$$

The result follows. □

Lemma 22. *If $h_1, h_2 \in \mathcal{HK}_C$, then $h_1 h_2 \in \mathcal{HK}_{C'}$ for some C'.*

Proof. It suffices to show that $V_{HK}(h_1 h_1) < C'$ for some C'. By Blümlinger and Tichy [1989] Proposition 2, h_1, h_2 are bounded. Then, Blümlinger and Tichy [1989] "From Proposition 1 we get xxx" proves $V_{HK}(h_1 h_2) \leq C'$ for some C'. □

Lemma 23. *If $h \in \mathcal{W}_C^\alpha([0,1]^d)$, then there exists some constant C' such that for all multi-index $\boldsymbol{\beta}$ with $|\boldsymbol{\beta}| \leq \lfloor \alpha \rfloor$, we have:*

$$\sup_{x \neq y} \frac{|D^{\boldsymbol{\beta}} h_2(x) - D^{\boldsymbol{\beta}} h_2(y)|}{\|x - y\|_\infty^{\alpha - \lfloor \alpha \rfloor}} \leq C'$$

Proof. For $|\boldsymbol{\beta}| = \lfloor \alpha \rfloor$, the result follows from the definition of the Hölder space. For $|\boldsymbol{\beta}| < \lfloor \alpha \rfloor$, by Taylor's theorem for multivariate functions, we have for some z:

$$D^{\boldsymbol{\beta}} h(x) = D^{\boldsymbol{\beta}} h(y) + \sum_{1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}| - 1} \frac{D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(y)}{\boldsymbol{\gamma}!} (x - y)^{\boldsymbol{\gamma}} + \sum_{\alpha - |\boldsymbol{\beta}| - 1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}|} \frac{D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(z)}{\boldsymbol{\gamma}!} (x - y)^{\boldsymbol{\gamma}}$$

where $(x - y)^{\boldsymbol{\gamma}} := \Pi_{i=1}^d (x_i - y_i)^{\gamma_i}$. Therefore, we have:

$$|D^{\boldsymbol{\beta}} h(x) - D^{\boldsymbol{\beta}} h(y)| \leq \sum_{1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}| - 1} \frac{|D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(y)|}{\boldsymbol{\gamma}!} \|x - y\|_\infty^{|\boldsymbol{\gamma}|} + \sum_{\alpha - |\boldsymbol{\beta}| - 1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}|} \frac{|D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(z)|}{\boldsymbol{\gamma}!} \|x - y\|_\infty^{|\boldsymbol{\gamma}|}$$

where we used $(x - y)^{\boldsymbol{\gamma}} = \Pi_{i=1}^d (x_i - y_i)^{\gamma_i} \leq \|x - y\|_\infty^{|\boldsymbol{\gamma}|}$. Then, we have:

$$\begin{aligned} \sup_{x \neq y} \frac{|D^{\boldsymbol{\beta}} h_2(x) - D^{\boldsymbol{\beta}} h_2(y)|}{\|x - y\|_\infty^{\alpha - \lfloor \alpha \rfloor}} \leq & \sum_{1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}| - 1} \frac{|D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(y)|}{\boldsymbol{\gamma}!} \|x - y\|_\infty^{|\boldsymbol{\gamma}| - \alpha + \lfloor \alpha \rfloor} \\ & + \sum_{\alpha - |\boldsymbol{\beta}| - 1 \leq |\boldsymbol{\gamma}| < \alpha - |\boldsymbol{\beta}|} \frac{|D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(z)|}{\boldsymbol{\gamma}!} \|x - y\|_\infty^{|\boldsymbol{\gamma}| - \alpha + \lfloor \alpha \rfloor} \end{aligned}$$

Since $|\boldsymbol{\gamma}| + |\boldsymbol{\beta}| < \alpha$ and $h \in \mathcal{W}_C^\alpha([0,1]^d)$, we have $\sup_y |D^{\boldsymbol{\gamma} + \boldsymbol{\beta}} h(y)| \leq C$. Moreover, as $\alpha - \lfloor \alpha \rfloor < 1$, $|\boldsymbol{\gamma}| \geq 1$ and $\|x - y\|_\infty \leq 1$, we have $\|x - y\|_\infty^{|\boldsymbol{\gamma}| - \alpha + \lfloor \alpha \rfloor} \leq 1$. Therefore, we have:

$$\sup_{x \neq y} \frac{|D^{\boldsymbol{\beta}} h_2(x) - D^{\boldsymbol{\beta}} h_2(y)|}{\|x - y\|_\infty^{\alpha - \lfloor \alpha \rfloor}} \leq C_{\boldsymbol{\beta}}$$

for some constant $C_{\boldsymbol{\beta}}$ that depends on $\boldsymbol{\beta}$. Since we have a finite number of multi-indices $\boldsymbol{\beta}$, we can take $C' := \max_{\boldsymbol{\beta}} C_{\boldsymbol{\beta}}$. □

Lemma 24. *If $h_1, h_2 \in \mathcal{W}_C^\alpha([0,1]^d)$, then $h_1 h_2 \in \mathcal{W}_{C'}^\alpha([0,1]^d)$ for some C'.*

Proof. For the multi-index $\boldsymbol{\beta}$ with $|\boldsymbol{\beta}| \leq \lfloor \alpha \rfloor$ by the Leibnitz formula, we have:

$$D^{\boldsymbol{\beta}} h_1 h_2 = \sum_{|\boldsymbol{\beta}_1|+|\boldsymbol{\beta}_2|=|\boldsymbol{\beta}|} \binom{|\boldsymbol{\beta}|}{|\boldsymbol{\beta}_1|} D^{\boldsymbol{\beta}_1} h_1 D^{\boldsymbol{\beta}_2} h_2$$

Then, we have:

$$\sup |D^{\boldsymbol{\beta}} h_1 h_2| \leq \sum_{|\boldsymbol{\beta}_1|+|\boldsymbol{\beta}_2|=|\boldsymbol{\beta}|} \binom{|\boldsymbol{\beta}|}{|\boldsymbol{\beta}_1|} \sup |D^{\boldsymbol{\beta}_1} h_1| \sup |D^{\boldsymbol{\beta}_2} h_2| \leq C_{\boldsymbol{\beta}} C^2$$

where we used $\sup |D^{\boldsymbol{\beta}} h_1 h_2| := \sup_x |D^{\boldsymbol{\beta}} h_1(x) h_2(x)|$ for notational simplicity and $C_{\boldsymbol{\beta}} := \sum_{|\boldsymbol{\beta}_1|+|\boldsymbol{\beta}_2|=|\boldsymbol{\beta}|} \binom{|\boldsymbol{\beta}|}{|\boldsymbol{\beta}_1|}$. Therefore, we have:

$$\sup_{|\boldsymbol{\beta}|<\alpha} \sup_x \left| D^{\boldsymbol{\beta}} f(x) \right| \leq \sup_{|\boldsymbol{\beta}|<\alpha} C_{\boldsymbol{\beta}} C^2 \leq C'' \tag{B.7}$$

Furthermore, for $|\boldsymbol{\beta_1}|, |\boldsymbol{\beta_2}| \leq \lfloor \alpha \rfloor$, we have:

$$\begin{aligned}
&\frac{|D^{\boldsymbol{\beta}_1} h_1(x) D^{\boldsymbol{\beta}_2} h_2(x) - D^{\boldsymbol{\beta}_1} h_1(y) D^{\boldsymbol{\beta}_2} h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \\
&\leq \frac{|D^{\boldsymbol{\beta}_1}(x) h_1 D^{\boldsymbol{\beta}_2} h_2(x) - D^{\boldsymbol{\beta}_1} h_1(x) D^{\boldsymbol{\beta}_2} h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} + \frac{|D^{\boldsymbol{\beta}_1} h_1(x) D^{\boldsymbol{\beta}_2} h_2(y) - D^{\boldsymbol{\beta}_1} h_1(y) D^{\boldsymbol{\beta}_2} h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \\
&\leq \sup |D^{\boldsymbol{\beta}_1} h_1| \frac{|D^{\boldsymbol{\beta}_2} h_2(x) - D^{\boldsymbol{\beta}_2} h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} + \sup |D^{\boldsymbol{\beta}_2} h_2| \frac{|D^{\boldsymbol{\beta}_1} h_1(x) - D^{\boldsymbol{\beta}_1} h_1(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \\
&\leq C \frac{|D^{\boldsymbol{\beta}_2} h_2(x) - D^{\boldsymbol{\beta}_2} h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} + C \frac{|D^{\boldsymbol{\beta}_1} h_1(x) - D^{\boldsymbol{\beta}_1} h_1(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \leq 2CC'
\end{aligned}$$

where C' is the constant in Lemma 23. Therefore, taking $|\boldsymbol{\beta}| = \lfloor \alpha \rfloor$ gives:

$$\sup_{|\boldsymbol{\beta}|=\lfloor\alpha\rfloor} \sup_{x \neq y} \frac{|D^{\boldsymbol{\beta}} h_1 h_2(x) - D^{\boldsymbol{\beta}} h_1 h_2(y)|}{\|x-y\|_\infty^{\alpha-\lfloor\alpha\rfloor}} \leq \sup_{|\boldsymbol{\beta}|=\lfloor\alpha\rfloor} 2CC'C_{\boldsymbol{\beta}} \leq C''' \tag{B.8}$$

Combining (B.7) and (B.8) proves the lemma. □

B.4.5 Supporting Lemmas for Theorem 6

Lemma 25. *Under Assumption 3, $\|\hat{\mathcal{K}}_M y - \mathcal{K} y\| \to 0$ as $M \to +\infty$ for any $y \in L^2(\mathbb{X})$.*

Proof. Since the operator $(\hat{\mathcal{K}}_M - \mathcal{K})$ is a bounded linear operator, the operator norm is upper bounded by the Hilbert-Schmidt norm, i.e.,

$$\|\hat{\mathcal{K}}_M - \mathcal{K}\|_{op} \leq \|\hat{\mathcal{K}}_M - \mathcal{K}\|_{HS}$$

Let $\tilde{K}(x'|x)$ and $K(x'|x)$ be the kernel of $\mathcal{K}_M$ and $\mathcal{K}$. Note that $\tilde{K}(x'|x) = 0$ for $x' \in \mathbb{X}\backslash\mathbb{M}$. Then, we have:

$$\begin{aligned}\|\hat{\mathcal{K}} - \mathcal{K}\|_{HS}^2 &= \int (\tilde{K}(x'|x) - K(x'|x))^2 d(x, x') \\ &= \int \left(\sum_i \tilde{K}^2(x_i|x) + \int K^2(x'|x)dx' - 2\sum_i \tilde{K}(x_i|x)K(x_i|x) \right) dx\end{aligned}$$

By Lemma 15, $K^2(\cdot|x) \in \mathcal{HK}_C$ for any x. Therefore, by the same arguments as in the proof of Lemma 16, we have:

$$\sum_i \tilde{K}^2(x_i|x) = \int K^2(x'|x)dx' + O(\frac{(\log M)^{d-1}}{M})$$

and

$$\sum_i \tilde{K}(x_i|x)K(x_i|x) = \int K^2(x'|x)dx' + O(\frac{(\log M)^{d-1}}{M})$$

where the constant in $O(\cdot)$ is uniformly over x. Then, we have:

$$\|\hat{\mathcal{K}}_M - \mathcal{K}\|_{HS}^2 = O(\frac{(\log M)^{d-1}}{M})$$

Therefore:

$$\|\hat{\mathcal{K}}_M y - \mathcal{K}y\| \leq \|\hat{\mathcal{K}}_M - \mathcal{K}\|_{op}\|y\| \leq \|\hat{\mathcal{K}}_M - \mathcal{K}\|_{HS}\|y\| \to 0 \text{ as } M \to +\infty$$

For the regular grids, the proof is the similar with Lemma 15 replaced by Lemma 18. □