ARTICLE IN PRESS

Journal of Econometrics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Econometrics

journal homepage: www.elsevier.com/locate/jeconom

Solving dynamic discrete choice models using smoothing and sieve methods☆

Dennis Kristensen [a,*], Patrick K. Mogensen [b], Jong Myun Moon [c], Bertel Schjerning [b]

[a] Department of Economics, University College London, Gower Street, London, United Kingdom
[b] Department of Economics, University of Copenhagen, Øster Farimagsgade 5, Building 35, DK-1353 Copenhagen K, Denmark
[c] PIMCO, United States of America

ARTICLE INFO

Article history:
Received 5 April 2018
Received in revised form 29 January 2020
Accepted 27 February 2020
Available online xxxx

Keywords:
Dynamic discrete choice
Numerical solution
Monte Carlo
Sieves

ABSTRACT

We propose to combine smoothing, simulations and sieve approximations to solve for either the integrated or expected value function in a general class of dynamic discrete choice (DDC) models. We use importance sampling to approximate the Bellman operators defining the two functions. The random Bellman operators, and therefore also the corresponding solutions, are generally non-smooth which is undesirable. To circumvent this issue, we introduce smoothed versions of the random Bellman operators and solve for the corresponding smoothed value functions using sieve methods. We also show that one can avoid using sieves by generalizing and adapting the "self-approximating" method of Rust (1997b) to our setting. We provide an asymptotic theory for both approximate solution methods and show that they converge with $\sqrt{N}$-rate, where N is number of Monte Carlo draws, towards Gaussian processes. We examine their performance in practice through a set of numerical experiments and find that both methods perform well with the sieve method being particularly attractive in terms of computational speed and accuracy.

1. Introduction

Discrete Decision Processes (DDPs) are widely used in economics to model forward-looking discrete decisions. For their implementation, researchers are required to solve the model which generally cannot be done in closed form. Instead, a number of methods have been proposed for solving the model numerically; see, e.g., Rust (2008) for an overview. We propose two novel methods for approximating the solutions to a general class of Markovian DDP models in terms of either the so-called integrated or expected value function. These two functions are relevant for estimation of DDP's and

☆ We would like to thank Mike Keane, John Rust, Victor Aguirregabiria, Lars Nesheim, Aureo de Paula and many other people for helpful comments and suggestions. Kristensen gratefully acknowledges financial support from the ERC (through starting Grant No. 312474 and advanced Grant No. GEM 740369). Schjerning gratefully acknowledges the financial support from the Independent Research Fund Denmark (Grant No. DFF 4182-00052) and the URBAN research project financed by the Innovation Fund Denmark.

* Corresponding author.
E-mail addresses: d.kristensen@ucl.ac.uk (D. Kristensen), Patrick.Kofod.Mogensen@econ.ku.dk (P.K. Mogensen), Bertel.Schjerning@econ.ku.dk (B. Schjerning).
URLs: https://sites.google.com/site/econkristensen (D. Kristensen), http://www.economics.ku.dk/staff/phd_kopi/?pure=en/persons/374766 (P.K. Mogensen), http://bschjerning.com/ (B. Schjerning).

https://doi.org/10.1016/j.jeconom.2020.02.007
0304-4076/

for welfare analysis of policy experiments. Our framework allows for both continuous and discrete state variables, non-separable utility functions and unrestricted dynamics. As such, we cover most relevant models used in empirical work. The proposed implementation of model and estimators are found to be computationally very efficient, and at the same time providing precise results with small approximation errors due to the use of simulations and sieve methods.

Our first proposal proceeds in three steps: First, we develop smoothed simulated versions of the Bellman operators that returns the integrated and expected value functions as fixed points. Next, we approximate the unknown value function by a sieve, that is, a parametric function class, thereby turning the problem into a finite-dimensional one. Finally, we solve for the parameters entering the chosen sieve using projection-based methods. When the chosen sieve is linear in the parameters, the approximate solution can be computed using an iterative procedure where each step is on closed form.

As an alternative to the above sieve-based method, we also adapt and generalize the so-called "self-approximating" method proposed in Rust (1997b) to our setting: We design the importance sampler used in the simulated Bellman operators so that the corresponding expected and integrated value functions can be solved for directly without the use of sieves. In comparison with the sieve approach, the self-approximating solution method has the advantage that it will not suffer from any biases due to function approximations. But at the same time, the importance sampler used in its implementation will generally have a larger variance compared to the class of samplers that can be used for the sieve method. This larger variance also translates into a larger simulation bias of the self-approximating solution due to the non-linear nature of the problem. Thus, neither method strictly dominates the other.

Our two procedures, the sieve-based and self-approximating one, differ from existing proposals in three important aspects: First, we solve for either the integrated or expected value function instead of the value function itself. This reduces the dimensionality of the problem since we integrate out any i.i.d. shocks appearing in the model before solving it. Moreover, while the value function is non-differentiable, the integrated and expected value functions are generally smooth which means that our sieve method performs better compared to existing ones that aim at approximating the value function. Second, we allow for a general class of importance samplers in the simulation of the Bellman operator; these can be designed to reduce variances and biases due to simulations. Third, we smooth the simulated Bellman operator by replacing the max-function appearing in its expression by a smoothed version where the degree of smoothing is controlled by a parameter akin to the bandwidth in kernel smoothing methods. This is similar to the logit-smoothed accept–reject simulator of probit models as proposed by McFadden (1989); see also Fermanian and Salanie (2004), Kristensen and Shin (2012) and Iskhakov et al. (2017). The smoothing turns the problem of solving for the integrated and expected value functions into differentiable ones. In particular, the exact solutions to the smoothed simulated Bellman equations become smooth as functions of state variables and any underlying structural parameters. This in turn means that standard sieves, such as polynomials, will approximate the exact solutions well and that we can control the error rate due to function approximation. Moreover, if used in estimation, standard numerical solvers can be employed in computing estimators of the structural parameters. The smoothing entails an additional bias but this can be controlled for by suitable choice of aforementioned smoothing parameter.

The smoothing device also facilitates the theoretical analysis of the approximate value functions since it allows us to use a functional Taylor expansion of it. This expansion is then used to analyze the leading numerical error terms of the approximate value functions due to simulations, smoothing and function approximations. In particular, under regularity conditions, we show that the approximate value function will converge weakly towards a Gaussian process which is the first result of its kind to our knowledge. These results allow researchers to, for example, build confidence intervals around the approximate value function and should be useful when analyzing the impact of value function approximation when used in welfare analysis and estimation of structural parameters. They may also be potentially helpful in designing selection rules for number of basis functions and the smoothing parameter.

A numerical study investigates the performance of the solution methods in practice. We implement the proposed methods for the engine replacement model of Rust (1987) and investigate how smoothing, number of basis functions and number of simulations affect the approximation errors. We also investigate how the procedures are affected by the dimensionality of the problem and how derivative-based solvers affect computation times. We find that the sieve method generally performs best of the two methods: It is computationally faster and in most situations provides a better approximation in terms of bias and variance. Moreover, the sieve method is found to also work well in higher dimensions with its bias and variance being fairly stable as we increase the number of state variables of the model. In contrast, variances of the self-approximating method increase dramatically as the number of state variables increases and so appears to be less robust. Finally, the errors due to simulations and function approximation behave according to theory and are found to vanish at the expected rates.

Our proposed methods share similarities with the ones developed in, amongst others, Arcidiacono et al. (2013), Keane and Wolpin (1994), Munos and Szepesvari (2008), Norets (2012), Pal and Stachurski (2013) and Rust (1997b) who also use simulations and/or sieve methods to solve DDP's. However, except for Keane and Wolpin (1994), the methods proposed in these papers approximate the value function while ours target the integrated or expected value function which are more well-behaved (smooth) objects and therefore easier to approximate. Moreover, in contrast to the cited papers, we employ importance sampling and smoothing in our implementation which comes with the aforementioned computational advantages. From a theory perspective, we provide a more complete asymptotic analysis of the approximate integrated and expected value functions. On the other hand, Munos and Szepesvari (2008) and Rust (1997b) provide an analysis

of the computational complexity of solving for the value function and so the theories of this paper and these studies complement each other.

The remainder of the paper are organized as follows: Section 2 introduces a general class of DDP's and their corresponding value functions. In Section 3, we develop our smoothed simulated versions of the Bellman operators that the integrated and expected value functions are fixed points to. We then show how to (approximately) solve these simulated Bellman equations in Section 4. An asymptotic theory of the approximate value function is presented in Section 5, while the results of the numerical experiments are found in Section 6. Appendix A contains some general results for approximate solutions to fixed point problems, while proofs of the main results can be found in Appendix B.

2. Model

We consider the following DDP where a single agent at time $t \geq 1$ solves

$$d_t = \arg\max_{d \in \mathcal{D}} \{u(S_t, d) + \beta E\left[v(S_{t+1})|S_t, d_t = d\right]\}, \tag{2.1}$$

where $\mathcal{D} = \{1, \ldots, D\}$ is the set of alternatives, $u(S_t, d)$ is the per-period utility, $0 < \beta < 1$ is the discount factor, S_t is a set of state variables that follows a controlled Markov process with transition kernel $F_S(S_t|S_{t-1}, d_{t-1})$ and the so-called value function v solves the following fixed-point problem,

$$v(S_t) = \max_{d \in \mathcal{D}} \{u(S_t, d) + \beta E\left[v(S_{t+1})|S_t, d_t = d\right]\}. \tag{2.2}$$

Following Rust (1987) and many subsequent empirical specifications, we assume that $S_t = (Z_t, \varepsilon_t) \in \mathcal{Z} \times \mathcal{E} \subseteq \mathbb{R}^{d_Z} \times \mathbb{R}^{d_\varepsilon}$ where Z_t and ε_t satisfy the following conditional independence condition,

$$F_S(Z_t, \varepsilon_t|Z_{t-1}, \varepsilon_{t-1}, d_{t-1}) = F_\varepsilon(\varepsilon_t|Z_t) F_Z(Z_t|Z_{t-1}, d_{t-1}).$$

In many cases $F_\varepsilon(\varepsilon_t|Z_t) = F_\varepsilon(\varepsilon_t)$ in which case ε_t is an i.i.d. sequence and so can be thought of as idiosyncratic shocks to utility. If no shocks are present in the model, we can always choose $\varepsilon_t = \emptyset$ to be an empty variable so that $S_t = Z_t$. Throughout, we will assume that the support $\mathcal{Z}$ is a compact set. This is done to simplify the theoretical analysis since it, for example, implies that value functions defined below will lie in the space of bounded functions on $\mathcal{Z}$, $\mathbb{B}(\mathcal{Z})$, equipped with the sup-norm, $\|v\|_\infty = \sup_{z \in \mathcal{Z}} |v(z)|$. At the same time, we allow the support of the error term, $\mathcal{E}$, be unbounded and for both countable and continuously distributed state variables.

In the above formulation, the model is characterized by the value function $v(s)$. However, it is possible to rewrite the models in terms of either the so-called *integrated value function* or the *expected value function* and solve for these instead. These are defined as

$$v(Z_t) = E\left[v(Z_t, \varepsilon_t)|Z_t\right] = \int_{\mathcal{E}} v(Z_t, e) \, dF_\varepsilon(e|Z_t),$$

and

$$V(Z_t, d_t) = E\left[v(Z_{t+1}, \varepsilon_{t+1})|Z_t, \varepsilon_t, d_t\right] = E[v(Z_{t+1})|Z_t, d_t] = \int_{\mathcal{E}} v(z') dF_Z\left(z'|Z_t, d_t\right),$$

respectively, where we have used the conditional independence assumption. Observe that given $v(z)$, we can recover $V(z, d) = E[v(Z_{t+1})|Z_t, d_t]$ which in turn can be used to compute $v(S_t) = \max_{d \in \mathcal{D}} \{u(Z_t, \varepsilon_t, d) + \beta V(Z_t, d)\}$. Thus, there is no loss in focusing on the integrated and expected value functions, except that in the former case we need to compute $E[v(Z_{t+1})|Z_t, d_t]$ numerically to obtain the value function. Furthermore, in many cases the expected value function itself is of interest. For example, the conditional choice probabilities, which are needed for counterfactuals and for estimation, take as input the relative expected value function,

$$P(d_t = d|Z_t = z) = M_{u,d}(\beta \Delta V(z)|z), \; M_{u,d}(r|z) = \frac{\partial M_u(r|z)}{\partial r(d)},$$

where $\Delta V(z, d) = V(z, d) - V(z, D)$, $d \in \mathcal{D}$, and $M_u(r|z)$ is a generalized version of the so-called social surplus function defined as, for any $r = (r(1), \ldots, r(D))$,

$$M_u(r|z) = \int_{\mathcal{E}} \max_{d \in \mathcal{D}} \{u(z, e, d) + r(d)\} \, dF_\varepsilon(e|z). \tag{2.3}$$

It is also useful in welfare analysis of policy experiments where we wish to see how a policy change will affect the expected present value of lifetime utility (i.e., the expected value function).

Except for a few special cases, analytical expressions of v and V are not available and so numerical approximations have to be employed. We will here develop numerical methods for solving for either v or V instead of v for the following reasons: First, v is a function of $s = (z, \varepsilon)$ while V and v are functions of z alone and therefore their approximations are lower-dimensional problems. Second, v is non-differentiable due to the max-function in (2.2); in contrast, $v(z)$ and $V(z, d)$ are both smooth functions of z if $F_\varepsilon(e|z)$ and $F_Z\left(z'|z, d\right)$ are. If there is no i.i.d. component in the model, $\varepsilon_t = \emptyset$,

then $v(s) = v(z) = v(z)$ and so the integrated value function becomes non-smooth. In contrast, $V(z, d)$ remains smooth even in this case. The functions v and V each solves their own fixed-point problem: Taking conditional expectations on both sides of Eq. (2.2), V can be expressed as the solution to

$$V(z, d) = \Gamma(V)(z, d), \tag{2.4}$$

where, with M_u defined in Eq. (2.3),

$$\begin{aligned}\Gamma(V)(z, d) &= E\left[\max_{d' \in \mathcal{D}} \left\{u(Z_{t+1}, \varepsilon_{t+1}, d') + \beta V(Z_{t+1}, d')\right\} | Z_t = z, d_t = d\right] \\ &= \int_{\mathcal{Z}} \int_{\mathcal{E}} \max_{d' \in \mathcal{D}} \left\{u(z', e, d') + \beta V(z', d')\right\} dF_\varepsilon(e|z') dF_Z(z'|z, d) \\ &= \int_{\mathcal{Z}} M_u(\beta V(z')|z') dF_Z(z'|z, d).\end{aligned}$$

Here and in the following, we let $V(z) = (V(z, 1), \ldots, V(z, D))'$ denote the $D \times 1$-vector of expected value function and similar for other objects. With this notation, we can represent the fixed-point problem in vector form, $V(z) = \Gamma(V)(z)$, where

$$\Gamma(V)(z) = \int_{\mathcal{Z}} M_u(\beta V(z')|z') dF_Z(dz'|z). \tag{2.5}$$

Next, to derive the fixed-point problem that v solves, again take conditional expectations on both sides of Eq. (2.2) but now only condition on Z_t to obtain

$$v(z) = M_u(\beta V(z)|z). \tag{2.6}$$

Combining this with Eq. (2.4),

$$v(z) = M_u\left(\beta \int_{\mathcal{Z}} M_u(\beta V(z')|z') dF_Z(dz'|z) \middle| z\right) = \bar{\Gamma}(v)(z). \tag{2.7}$$

where

$$\bar{\Gamma}(v)(z) = M_u\left(\beta \int_{\mathcal{Z}} v\left(z'\right) dF_Z(dz'|z) \middle| z\right).$$

Under regularity conditions provided below, Γ and $\bar{\Gamma}$ are contraction mappings and so V and v are well-defined and unique. The above two transformations of the original problem into the ones for either the integrated or expected value function are particular cases of the general class of transformations analyzed in Ma and Stachurski (2020).

Example 1. Consider the special case where $u(Z_{t+1}, \varepsilon_{t+1}, d) = \bar{u}(Z_{t+1}, d) + \lambda \varepsilon_{t+1}(d)$ for some scale parameter $\lambda > 0$ and $F_\varepsilon(e|z) = F_\varepsilon(e)$ in which case

$$M_u(r|z) = \int_{\mathcal{E}} \max_{d \in \mathcal{D}} \{\bar{u}(z, d) + \lambda e(d) + r(d)\} dF_\varepsilon(e) = G_\lambda(\bar{u}(z) + r),$$

where $G_\lambda(r) := \int_{\mathcal{E}} \max_{d \in \mathcal{D}} \{\lambda e(d) + r(d)\} dF_\varepsilon(e)$. Thus,

$$\bar{\Gamma}(v)(z) = G_\lambda\left(\bar{u}(z) + \beta \int_{\mathcal{Z}} v\left(z'\right) dF_Z(z'|z)\right).$$

If $\varepsilon_t(1), \ldots, \varepsilon_t(D)$ are mutually independent and each component follows a suitably normalized extreme value distribution then (see, e.g., Rust et al., 2002),

$$G_\lambda(r) = \lambda \log\left[\sum_{d \in \mathcal{D}} \exp\left(\frac{r(d)}{\lambda}\right)\right]. \tag{2.8}$$

3. Simulated Bellman operators

As a first step towards a computationally feasible method for solving for either v or V, we here develop simulated versions of their two Bellman operators and then introduce the smoothing device. To allow for added flexibility and precision in the implementation and to cover as special case a modified version of Rust's self-approximating solution method, we employ importance sampling: Let $\Phi_Z\left(z'|z, d\right)$ and $\Phi_\varepsilon(e|z)$ be conditional importance sampling distribution functions as chosen by the researcher. These have to be chosen such that $F_Z(\cdot|z, d)$ and $F_\varepsilon(\cdot|z)$ are absolutely continuous w.r.t. $\Phi_Z(\cdot|z, d)$ and $\Phi_\varepsilon(\cdot|z)$, respectively, with Radon–Nikodym derivatives $w_Z(\cdot|z, d) \geq 0$ and $w_\varepsilon(\cdot|z) \geq 0$ so that

$$\frac{dF_Z(z'|z, d)}{d\Phi_Z(z'|z, d)} = w_Z\left(z'|z, d\right), \quad \frac{dF_\varepsilon(e|z)}{d\Phi_\varepsilon(e|z)} = w_\varepsilon(e|z). \tag{3.1}$$

We will throughout assume that Eq. (3.1) is satisfied. In the leading case $dF_Z = f_Z d\mu_Z$ and $d\Phi_Z = \phi_Z d\mu_Z$ for some measure μ_Z in which case $w_Z = f_Z/\phi_Z$ and similar for the sampling of ε_t. The above covers the case where $F_Z(z'|z,d)$ is a continuous distribution (in which case μ_Z is the Lesbesgue measure), a discrete distribution (in which case μ_Z is the counting measure) and the mixed case. With discrete finite support, we could in principle compute the exact Bellman equation and its corresponding solution and so would not need to resort to numerical methods. But if the discrete support is large this may still be computationally very demanding and so even in this case the numerical methods developed below may be computationally attractive, cf. Arcidiacono et al. (2013).

Given the chosen importance sampler, we can rewrite $\Gamma(V)(z,d)$ as

$$\Gamma(V)(z,d) = \int_{\mathcal{Z}} \int_{\mathcal{E}} \max_{d' \in \mathcal{D}} \left\{ u(s',d') + \beta V(z',d') \right\} w\left(s'|z,d\right) d\Phi(s'|z,d)$$

where $s' = \left(z', e'\right)$ and

$$w\left(s'|z,d\right) = w_\varepsilon\left(e'|z'\right) w_Z\left(z'|z,d\right), \ \Phi\left(s'|z,d\right) = \Phi_\varepsilon(e'|z')\Phi_Z\left(z'|z,d\right).$$

For any given candidate V, we can then approximate this integral by Monte Carlo methods: First generate $N \geq 1$ i.i.d. draws, $Z_i(z,d) \sim \Phi_Z(\cdot|z,d)$ and $\varepsilon_i(z,d) \sim \Phi_\varepsilon(\cdot|Z_i(z,d))$, $i = 1,\ldots,N$, and then compute

$$\Gamma_N(V)(z,d) = \sum_{i=1}^{N} \max_{d' \in \mathcal{D}} \left\{ u\left(S_i(z,d), d'\right) + \beta V\left(Z_i(z,d), d'\right) \right\} w_{N,i}(z,d), \tag{3.2}$$

where $S_i(z,d) = (Z_i(z,d), \varepsilon_i(z,d))$ and

$$w_{N,i}(z,d) = \frac{w\left(S_i(z,d)|z,d\right)}{\sum_{i=1}^{N} w\left(S_i(z,d)|z,d\right)}. \tag{3.3}$$

Note here that we normalize the importance weights so that $\sum_{i=1}^{N} w_{N,i}(z,d) = 1$. This is done to ensure that Γ_N is a contraction mapping on $\mathbb{B}(\mathcal{Z})^D$. Similarly, we approximate $\bar{\Gamma}(v)$ by

$$\bar{\Gamma}_N(v)(z) = \sum_{j=1}^{N} \max_{d' \in \mathcal{D}} \left\{ u(z, \varepsilon_j\left(z,d'\right), d') + \beta \sum_{i=1}^{N} v\left(Z_i\left(z,d'\right)\right) w_{Z,N,i}\left(z,d'\right) \right\} w_{\varepsilon,N,j}\left(z,d'\right), \tag{3.4}$$

where again we normalize the weights to ensure $\bar{\Gamma}_N$ is a contraction on $\mathbb{B}(\mathcal{Z})$,

$$w_{Z,N,i}(z,d) = \frac{w_Z\left(Z_i(z,d)|z,d\right)}{\sum_{i=1}^{N} w_Z\left(Z_i(z,d)|z,d\right)}, \ w_{\varepsilon,N,i}(z,d) = \frac{w_\varepsilon\left(\varepsilon_i(z,d)|z\right)}{\sum_{i=1}^{N} w_\varepsilon\left(\varepsilon_i(z)|z\right)}.$$

When $\varepsilon_t = \emptyset$, the simulated Bellman operator $\bar{\Gamma}_N$ includes as special cases the ones considered in Rust (1997b) (who chooses Φ_Z as the uniform distribution on $\mathcal{Z}$) and Pal and Stachurski (2013) (who choose $\Phi_Z = F_Z$).

Example 1 (*Continued*)**.** Suppose we can compute the integral w.r.t. ε_t analytically. In this case, the following simplified version of the simulated Bellman operator can be employed,

$$\bar{\Gamma}_N(v)(z) = G_\lambda\left(\bar{u}(z) + \beta \sum_{i=1}^{N} v\left(Z_i(z)\right) w_{Z,N,i}(z)\right), \tag{3.5}$$

where G_λ was defined in Eq. (2.8). Importantly, the max-function has been replaced by its smoothed version $G_\lambda(\cdot)$.

If $F_\varepsilon(e|z)$ and $F_Z(z'|z,d)$ are smooth functions w.r.t. z then $\Gamma(V)(z)$ and $\bar{\Gamma}(v)(z)$ will be smooth functions of z as well. In contrast, the general versions of $\Gamma_N(V)(z,d)$ and $\bar{\Gamma}_N(v)(z)$ are non-smooth due to the presence of the max-function in their definitions which does not get smoothed for finite N. This in turn implies that their corresponding fixed points, $V_N(z) = \Gamma_N(V_N)(z)$ and $v_N(z) = \bar{\Gamma}_N(v_N)(z)$, will be non-differentiable w.r.t. the state variables, z, and w.r.t. any underlying structural parameters in the model. This is an unattractive feature for two reasons: First, estimation and counterfactuals will be non-smooth problems. Second, the theoretical analysis of V_N and v_N becomes more complicated.

To resolve this issue, we take inspiration from the additive model in and propose to smooth the simulated Bellman operators by replacing the "hard" max-function appearing in Eqs. (3.2) and (3.4) by its smoothed version $G_\lambda(r)$ defined in Eq. (2.8). This yields the following smoothed simulated operators,

$$\Gamma_N(V)(z;\lambda) = \sum_{i=1}^{N} G_\lambda\left(u\left(S_i(z,d)\right) + \beta V\left(Z_i(z,d)\right)\right) w_{N,i}(z), \tag{3.6}$$

$$\bar{\Gamma}_N(v)(z;\lambda) = \sum_{j=1}^{N} G_\lambda\left(u(z,\varepsilon_j(z)) + \beta \sum_{i=1}^{N} v\left(Z_i(z)\right) w_{Z,N,i}(z)\right) w_{\varepsilon,N,j}(z), \tag{3.7}$$

where $u(z, \varepsilon_j(z)) = \left(u(z, \varepsilon_j(z, 1), 1), \ldots, u(z, \varepsilon_j(z, D), D)\right)$ and other vector functions are defined similarly. Setting $\lambda = 0$ in Eqs. (3.6)–(3.7), we recover the original non-smooth versions defined in Eqs. (3.2)–(3.4). Thus, the smoothed versions are generalized versions of the original ones. The use of $G_\lambda(r)$ in place of $\max_{d\in\mathcal{D}} r(d)$ generates an additional bias in the approximate solutions, but this can be controlled for by suitable choice of λ. We now interpret $\lambda > 0$ as a smoothing parameter that plays a role similar to that of the bandwidth in kernel regression estimation. Elementary calculations show

$$0 \leq G_\lambda(r) - \max_{d\in\mathcal{D}} r(d) \leq \lambda \log D, \tag{3.8}$$

so that $G_\lambda(r) \to \max_{d\in D} r(d)$, as $\lambda \to 0$, uniformly in $r \in \mathbb{R}^D$. Thus, the smoothing entails a bias of order $O_P(\lambda)$. We discuss the choice of λ in practice in the next section.

In some situations, $G_\lambda(r)$ appears in the Bellman operators as an inherent feature of the model specification in which case no smoothing bias will be present. We saw this in and it extends to the following class of models: Suppose that $\varepsilon_t = \left(\varepsilon_t^{(1)}, \varepsilon_t^{(2)}\right)$ with $\varepsilon_t^{(1)} = \left(\varepsilon_t^{(1)}(1), \ldots, \varepsilon_t^{(1)}(D)\right)$ are mutually independent extreme value shocks that enter the per-period utility additively,

$$d_t = \arg\max_{d\in\mathcal{D}} \left\{\bar{u}(Z_t, \varepsilon_t^{(2)}, d) + \lambda\varepsilon_t^{(1)}(d) + \beta V(Z_t, d)\right\}, \tag{3.9}$$

where as in $\lambda > 0$ is scale parameter that determines the impact of $\varepsilon_t^{(1)}(d)$ on the per-period utility. By the same arguments as in , we find that the expected value function in this case solves $\Gamma_\lambda(V) = V$ where

$$\Gamma_\lambda(V)(z) = \int_{\mathcal{Z}}\int_{\mathcal{E}} G_\lambda\left(u(z', e') + \beta V(z')\right) dF_{\varepsilon^{(2)}}(e'|z) dF_Z(z'|z, d),$$

and $\Gamma_{N,\lambda}$ in Eq. (3.6) is clearly an unbiased simulated version of Γ_λ. Similarly, $\bar{\Gamma}_{N,\lambda}$ is an unbiased estimator of $\bar{\Gamma}_\lambda$. To summarize, if the original model of interest contains an additive extreme value term, which is the case in many empirical papers, G_λ appears as part of the model and so no smoothing bias will be present in our proposed simulated Bellman operators.

The above shows that the smoothing device corresponds to adding structural shocks to the DDP of interest. In earlier work on solving DDPs, researchers have in some cases done the opposite and removed structural errors in order to facilitate the numerical solution of the model; see Lumsdaine et al. (1992) for one example of this. This was, however, done in the context of discrete state variables with a small number of support points in which case removing continuous structural errors meant that the Bellman operators could be evaluated analytically. Our method is aimed at models where the state variables are either continuous or have a very large discrete support in which case simulations are required in the first place to evaluate the Bellman operator. Once simulations are introduced, there is little computational gain from removing shocks from the model and instead the introduction of smoothing facilitates solving and analyzing the corresponding solution.

4. Approximate value functions

The smoothed simulated Bellman operators $\Gamma_N(V)(z, \lambda)$ and $\bar{\Gamma}_N(v)(z, \lambda)$ in Eqs. (3.6)–(3.7) are functionals that, for given function V and v, depend on (z, λ). We will here and in the following treat them as functionals that take given function $V(z, \lambda)$ and $v(z, \lambda)$, respectively, and map them into functions of $(z, \lambda) \in \mathcal{Z} \times \left[0, \bar{\lambda}\right]$ for some $\bar{\lambda} > 0$. This simplifies the analysis of the impact of smoothing. In particular, under suitable regularity conditions, Γ_N and $\bar{\Gamma}_N$ are contraction mappings on $\mathbb{B}\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)^D$ and $\mathbb{B}\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)$, respectively, where $\mathbb{B}\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)$ denotes the space of bounded functions with domain $\mathcal{Z} \times \left[0, \bar{\lambda}\right]$. Thus, they have unique fixed points $V_N(z, \lambda)$ and $v_N(z, \lambda)$ solving

$$V_N(z, \lambda) = \Gamma_N(V_N)(z, \lambda), \quad v_N = \bar{\Gamma}_{N,\lambda}(v_N)(z, \lambda). \tag{4.1}$$

In practice, we will only solve for the particular value of λ as chosen by us, but for the theory it proves helpful to treat the solutions as mappings defined on $(z, \lambda) \in \mathcal{Z} \times \left(0, \bar{\lambda}\right)$. However, solving these two simulated Bellman equations are not generally feasible since these are infinite-dimensional problems. We here present two ways to reduce the problems to finite-dimensional ones. The first method is a generalized version of the so-called self-approximating method proposed in Rust (1997b) while the second one uses projection-based methods as advocated by Pal and Stachurski (2013).

4.1. Self-approximating method

Rust (1997b) proposed to turn the infinite-dimensional problems in Eq. (4.1) into a finite-dimensional ones by choosing the importance sampling to be based on marginal, instead of conditional distributions. In our generalized version this corresponds to restricting $\Phi_Z\left(z'|z, d\right) = \Phi_Z\left(z'\right)$ for some marginal distribution $\Phi_Z(\cdot)$ so that the draws $Z_i \sim \Phi_z(\cdot)$ and $\varepsilon_i \sim \Phi_\varepsilon(\cdot|Z_i)$, $i = 1, \ldots, N$ no longer depend on (z, d). In this case, for a given value of $\lambda \in \left[0, \bar{\lambda}\right]$, the fixed-point problems in Eq. (4.1) reduce to the following two sets of N nonlinear equations,

$$V_{N,k} = \sum_{i=1}^{N} G_\lambda\left(u(S_i) + \beta V_{N,i}\right) w_{N,i}(Z_k), \tag{4.2}$$

$$v_{N,k} = \sum_{j=1}^{N} G_\lambda \left(u(Z_k, \varepsilon_j) + \beta \sum_{i=1}^{N} v_{N,i} w_{z,i}(Z_k) \right) w_{\varepsilon,N,j}(Z_k), \tag{4.3}$$

for $k = 1, \ldots, N$, that can be solved for w.r.t. $\{V_{N,\lambda,k} : k = 1, \ldots, N\}$ and $\{v_{N,\lambda,k} : k = 1, \ldots, N\}$, respectively. Here, $V_{N,k} = V_N(Z_k, \lambda)$ and $v_{N,k} = v_{N,\lambda}(Z_k, \lambda)$, $k = 1, \ldots, N$. Each of the two sets of equations have a unique solution due to the contracting property of $\Gamma_{N,\lambda}$ and $\bar{\Gamma}_{N,\lambda}$. Once, for example, Eq. (4.2) has been solved, the approximate expected value functions can be evaluated at any other value z by

$$V_N(z, \lambda) = \sum_{i=1}^{N} G_\lambda \left(u(S_i) + \beta V_{N,i} \right) w_{N,i}(z).$$

Note that $V_N(z, \lambda)$ is a smooth function even if $\lambda = 0$ as long as $w_{N,i}(z)$ is smooth and so smoothing is not needed for this property to hold when marginal samplers are employed. However, without smoothing, the set of equations (4.2) becomes non-smooth w.r.t. the variables $\{V_{N,k} : k = 1, \ldots, N\}$ and so cannot be solved using derivative-based methods. Thus, the numerical implementation of the self-approximating method still benefits from smoothing.

In addition to smoothing, the above self-approximating method differs from Rust's original proposal in two other ways: First, while Rust (1997b) solved for the value function $v(z, \varepsilon)$, we here solve for either $V(z, \lambda)$ or $v(z, \lambda)$ for a fixed value of λ. As explained in Section 2, V and v convey the same information as v and at the same time they are of lower dimension in terms of variables and are more smooth, features which facilitate their numerical approximation. Moreover, our formulation allows for the following generalized version of the simulated Bellman equations for v_N,

$$v_{N,k} = \sum_{j=1}^{\tilde{N}} G_\lambda \left(u(Z_k, \varepsilon_j) + \beta \sum_{i=1}^{N} v_{N,i} w_{z,N,i}(Z_k) \right) w_{\varepsilon,N,j}(Z_k), \tag{4.4}$$

where we allow for different number of draws from Φ_ε and Φ_Z. In particular, we can choose $\tilde{N}$ as large as we wish (thereby decreasing the variance of the problem) without increasing the number of variables that need to be solved for (N). A similar generalization of the simulated Bellman equations for V_N is possible. Second, we here only require that the state dynamics together with the chosen importance sampler satisfy (3.1); in contrast, Rust (1997b) assumed that S_t was continuously distributed with compact support and chose as importance sampler the uniform distribution with same support. Thus, our version allows for a broader class of models and samplers.

The self-approximating method may not always work well: First, finding a marginal distribution $\Phi_Z(\cdot)$ so that (3.1) holds can be difficult in some models. For example, in many specifications with continuous dynamics, the transition density $f_Z(z'|z, d)$ of Z_t will have singularities, e.g., $\lim_{z' \to z} f_Z(z'|z, d) = +\infty$, in which case $w_Z(z'|z, d) = f_Z(z'|z, d) / \phi_Z(z')$ is not well-defined no matter how we choose $\phi_Z(z')$. And even if (3.1) does hold, the use of marginal samplers instead of conditional ones will generally lead to a larger variance of the solutions since the "marginal" draws $Z_1, \ldots, Z_N$ do not adapt to the changing shape of $F_Z(\cdot|z, d)$ as a function of z. In particular, many of the draws may fall outside of the support of $F_Z(\cdot|z, d)$ and so are "wasted" in which case a large N is required to achieve a reasonable approximation; see Section 6 for an example of this. This issue tends to become more severe in higher dimensions (d_Z is "large") since the volume of the support shrinks, and so the self-approximating method will generally suffer from a built-in curse-of-dimensionality. This curse-of-dimensionality does not appear in the subclass of models that Rust (1997b) focused on where it was assumed that $S_t|S_{t-1}, d_{t-1}$ has support $[0, 1]^{\dim(S_t)}$ for all values of S_{t-1}, d_{t-1}.

Finally, given that $V_{N,\lambda}$ and $v_{N,\lambda}$ are solutions to non-linear equations, a large variance in the simulated Bellman operator translates into a large bias as is well-known from non-linear GMM estimators. This can be controlled for by choosing N large. But large N means that numerically solving either (4.2) or (4.3) becomes computationally very costly. These issues motivate us to pursue a sieve-based solution strategy.

4.2. Sieve-based method

We now return to the general versions of the simulated Bellman operators and so again allow for conditional importance samplers. Let $\mathcal{V} \subseteq \mathbb{B}(\mathcal{Z})$ be a suitable function space that $z \mapsto v_N(z; \lambda)$ defined in (4.1) is known to lie in; see below for more details on this. We then choose a finite-dimensional function space (commonly called a sieve in the econometrics literature) $\bar{\mathcal{V}}_K = \{v_K(\cdot; \alpha) : \mathcal{Z} \mapsto \mathbb{R} | \alpha \in \mathcal{A}_K\} \subseteq \bar{\mathcal{V}}$, where $\mathcal{A}_K \subseteq \mathbb{R}^K$ is a parameter set with $K < \infty$, that provides a good approximation to functions in $\bar{\mathcal{V}}$. Similarly, we let $\mathcal{V} \subseteq \mathbb{B}(\mathcal{Z})^D$ be a space of D-dimensional vector functions that the solution $z \mapsto V_N(z; \lambda)$ to (4.1) lie in and $\mathcal{V}_K = \{V_K(\cdot; \alpha) : \mathcal{Z} \mapsto \mathbb{R}^D | \alpha \in \mathcal{A}_K\} \subseteq \mathcal{V}$ be our sieve for this space. Let

$$\bar{\Pi}_K(v) = \arg\min_{v' \in \bar{\mathcal{V}}_K} \|v - v'\|_{\bar{\mathcal{V}}}, \quad \Pi_K(V) = \arg\min_{V' \in \mathcal{V}_K} \|V - V'\|_{\mathcal{V}}, \tag{4.5}$$

be the corresponding projections for given (pseudo-) norms $\|\cdot\|_{\bar{\mathcal{V}}}$ and $\|\cdot\|_{\mathcal{V}}$ as chosen by us as well. We then approximate $V_{N,\lambda}$ and $v_{N,\lambda}$ by the solutions to the projected Bellman equations,

$$\hat{v}_{N,\lambda} = \arg\min_{v \in \bar{\mathcal{V}}_K} \left\| v - \bar{\Pi}_K \bar{\Gamma}_{N,\lambda}(v) \right\|_{\bar{\mathcal{V}}}, \quad \hat{V}_{N,\lambda} = \arg\min_{V \in \mathcal{V}_K} \left\| V - \Pi_K \Gamma_{N,\lambda}(V) \right\|_{\mathcal{V}}. \tag{4.6}$$

These are finite-dimensional problems of size K. When K is small relative to N, which will generally be the case, the above problems are computationally much more tractable compared to the corresponding self-approximating ones. Note here that these projection-based approximations are different from the least-squares approximations that would solve $\min_{V\in\mathcal{V}_K}\left\|V-\Gamma_{N,\lambda}(V)\right\|_{\mathcal{V}}$ and $\min_{v\in\bar{\mathcal{V}}_K}\left\|v-\bar{\Gamma}_{N,\lambda}(v)\right\|_{\bar{\mathcal{V}}}$, respectively. In particular, by suitable choice of the projection operators, $\Pi_K\Gamma_{N,\lambda}$ and $\bar{\Pi}_K\bar{\Gamma}_{N,\lambda}$ will be contraction mappings w.r.t. $\|\cdot\|_\infty$ guaranteeing that $\hat{V}_{N,\lambda}$ and $\hat{v}_{N,\lambda}$ exist and are unique. The following discussion focuses on the integrated value function approximation since it carries over with only minor modifications to the one of the expected value function. We discuss their numerical implementation in further detail in the subsection below.

The projection operator $\bar{\Pi}_K$ can be thought of as a function approximator with the approximation error being $v-\bar{\Pi}_K(v)$ for a given function v. Roughly speaking, the projection-based method approximates v_N in (4.1) by $\hat{v}_N=\bar{\Pi}_K(v_N)$ which incurs an additional sieve approximation error, $v_N-\bar{\Pi}_K(v_N)$. The smoothness of v_N here proves helpful since many well-known sieves are able to provide good approximations of smooth functions using a low-dimensional space ("small" K). Due to these features, our proposed projection-based solutions will generally suffer from quite small additional biases relative to the exact simulated solution. This is in contrast to existing projection-based solution methods, such as the one in Pal and Stachurski (2013), that aim at approximating the value function $v(s)$ which is non-differentiable.

The smoothness of v_N here help guiding us in choosing the sieve: It allows us to restrict $\bar{\mathcal{V}}$ to a suitable smoothness class and then import existing approximation methods for smooth functions as developed in the literature on numerical methods and nonparametric econometrics. A leading example is the class of linear function approximations where the finite-dimensional function space takes the form of $\bar{\mathcal{V}}_K=\left\{\alpha' B_K(z):\alpha\in\mathbb{R}^K\right\}$ for a set of basis functions $B_K(z)=(b_1(z),\ldots,b_K(z))'$. The basis functions can be chosen as, for example, Chebyshev polynomials or B-splines that are able to approximate smooth functions well. However, other non-linear function spaces are possible such as wavelets, artificial neural networks and shrinkage-type function approximators such as LASSO, where the additional constraints are imposed on α; we refer to Chen (2007) for a general overview of different function approximators and constrained sieve estimators. We also allow for flexibility in terms of the chosen norms $\|\cdot\|_{\bar{\mathcal{V}}}$ with a leading example being $\|v\|_{\bar{\mathcal{V}}}=\sum_{i=1}^M v^2(z_i)$ for a set of design points $z_1,\ldots,z_M\in\mathcal{Z}$. Very often the $M\geq 1$ design points will be chosen in conjunction with the sieve.

The above procedure does not suffer from any of the above mentioned issues of the self-approximating method: We can use conditional importance samplers freely which can be designed to control the variance of the simulated Bellman operators; and the dimension of the problem remains K irrespectively of the number of draws N. The main drawback is that unique solutions to Eqs. (4.6) do not necessarily exist for a given choice of N and K. A sufficient condition for this to hold is that $\bar{\Pi}_K$ is a non-expansive operator w.r.t. $\|\cdot\|_\infty$, that is, $\left\|\bar{\Pi}_K(v_1)-\bar{\Pi}_K(v_2)\right\|_\infty\leq\|v_1-v_2\|_\infty$ for any two functions $v_1,v_2\in\bar{\mathcal{V}}$, since this translates into $\bar{\Pi}_K\Gamma_N$ being a contraction mapping. However, while $\bar{\Pi}_K$ is non-expansive w.r.t. $\|\cdot\|_{\bar{\mathcal{V}}}$ by definition, it is not necessarily non-expansiveness w.r.t. $\|\cdot\|_\infty$. Pal and Stachurski (2013) provide some examples of projections that are non-expansive w.r.t. $\|\cdot\|_\infty$, but these are unfortunately computationally expensive to use in general. But $\bar{\Pi}_K$ will generally be close to non-expansive w.r.t. $\|\cdot\|_\infty$ asymptotically as $K\to\infty$ for a wide range of sieves and pseudo-norms in the sense that

$$\left\|\bar{\Pi}_K\right\|_{op,\infty}:=\sup_{v\in\bar{\mathcal{V}},\|v\|=1}\left\|\bar{\Pi}_K(v)\right\|_\infty\leq\sup_{v\in\bar{\mathcal{V}},\|v\|=1}\left\|\bar{\Pi}_K(v)-v\right\|_\infty+1,$$

where the first term in the last expression will go to zero in great generality as $K\to\infty$ for many popular sieves (see next subsection for details). Given that $\bar{\Gamma}_N$ is a contraction with Lipschitz coefficient $\beta<1$, this in turn implies that $\bar{\Pi}_K\bar{\Gamma}_N$ will be a contraction mapping for all K large enough. This will be used in our asymptotic analysis of the algorithm. Unfortunately, it is generally not known how large K should be chosen to ensure $\bar{\Pi}_K\bar{\Gamma}_N$ is a contraction. But in our numerical experiments we did not experience any convergence problems.

4.3. Numerical implementation of the two methods

We here discuss in more detail the numerical implementation of the self-approximating and projection-based methods. First, the researcher has to choose the importance sampling distributions, the smoothing parameter λ and, in the case of the projection-based method, the function approximation method. Second, given these choices, either Eq. (4.3) or (4.6) has to be solved for. As before, the discussion here focuses on solving for the integrated value function since most results and arguments for this case carries over with very minor modifications to the expected value function.

4.3.1. Importance sampler

The choices of Φ_Z and Φ_ε determine the variance of $\bar{\Gamma}_N$ and should ideally be tailored to minimize it. In the case of projection-based methods, where we can choose Φ_Z and Φ_ε as conditional distributions, we can rely on the already existing theory for efficient importance sampling for how to do so; see Chapter 3 in Robert and Casella (2013) for an introduction. In our numerical experiments, we did not experiment with different choices and throughout set $\Phi_Z=F_Z$ and $\Phi_\varepsilon=F_\varepsilon$.

In the case of the self-approximating method, the choice of Φ_Z is restricted to the class of marginal distributions. Generally, this entails a large variance of the corresponding simulated Bellman operators. It will hold in great generality that (Z_t,d_t) has a stationary distribution, say, $F_S^*(z,d)$. In this case, a suitable choice would be the marginal of this,

$\Phi_Z(z) = F_S^*(z, D)$. However, the stationary distribution depends on the value function and so is rarely available on closed form; so this strategy requires an initial exploration of the model and its solution. Alternatively, one can try to construct a good approximation of the stationary approximation through a mixture Markov model on the form $\Phi_Z(z') = \sum_{d\in\mathcal{D}} \int \omega_d(z) F_Z(z'|z, d) d\mu_Z(z)$ for a set of pre-specified mixture weights $\omega_d(z) \geq 0$. In the numerical experiments, we follow Rust (1997b) and choose $\Phi_Z(z)$ as the uniform distribution on $\mathcal{Z}$ which we conjecture is far from optimal in many cases, and so more research in this direction is needed.

4.3.2. Smoothing

The use of $G_\lambda(r)$ in place of $\max_{d\in\mathcal{D}} r(d)$ generally generates an additional bias in the corresponding integrated value function of order $O(\lambda)$. At the same time, the variance of v_N is an increasing function of λ. Thus, ideally we would like to choose λ to balance these two effects. A natural criterion would be to minimize the so-called integrated mean-square-error, $\lambda^* = \arg\min_{\lambda\geq 0} E\left[\int_{\mathcal{Z}} \left\| v_{N,\lambda}(z) - v(z)\right\|^2 dF_Z(z)\right]$, where $F_Z(z)$ is a suitably chosen distribution such as the stationary one of Z_t. Since $v(z)$ is unknown and we cannot evaluate the expectations, λ^* cannot be solved for but cross-validation methods can be used instead. This could in principle be done along the same lines as bandwidth selection for smoothed empirical cdfs, see Bowman et al. (1998). However, this is computationally somewhat burdensome. Moreover, in our numerical experiments we found that the quality of the approximate value function was quite insensitive to the choice of λ and so in practice we recommend using very little smoothing such as $\lambda = 0.01$.

4.3.3. Function approximation

As mentioned earlier, many approximation architectures are available in the literature. In our numerical experiments we focus on the class of linear function approximators where $\bar{\mathcal{V}}_K = \left\{\alpha' B_K(z) : \alpha \in \mathbb{R}^K\right\}$ for a set of pre-specified basis functions $B_K(z) \in \mathbb{R}^K$. For a given set of $M \geq 1$ design points in $\mathcal{Z}$, $z_1, \ldots, z_M$, Eq. (4.5) then becomes

$$\bar{\Pi}_K(v)(z) = B_K(z)' \left[\sum_{i=1}^M B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^M B_K(z_i) v(z_i). \tag{4.7}$$

The design points may either be random or deterministic and can be chosen relative to the basis functions to ensure that $\bar{\Pi}_K$ is easy to compute and provides a good approximation for a broad class of functions. The performance of most function approximations will depend on the smoothness of the function of interest.

A standard smooth function class often considered in approximation theory is the following: For any vector $a = \left(a_1, \ldots, a_{d_Z}\right) \in \mathbb{N}_0^{d_X}$, let $D^a f(x) = \partial^{|a|} f(x) / \left(\partial x_1^{\alpha_1} \cdots \partial x_{d_Z}^{a_{d_Z}}\right)$, where $|a| = a_1 + \cdots + a_{d_z}$, be the corresponding partial derivative. For $\alpha > 0$, let $\underline{\alpha} \geq 0$ be the greatest integer smaller than α. For any $\underline{\alpha}$ times differentiable function $f(x)$, we then define

$$\|f\|_{\alpha,\infty} = \max_{|a|\leq\underline{\alpha}} \|D^a f\|_\infty + \max_{|a|=\underline{\alpha}} \sup_{x_1\neq x_2} \frac{|D^a f(x_1) - D^a f(x_2)|}{\|x_1 - x_2\|^{\alpha-\underline{\alpha}}}, \tag{4.8}$$

and let $\mathbb{C}_r^\alpha(\mathcal{X})$ be the space of all $\underline{\alpha} \geq 0$ times continuously differentiable functions $f : \mathcal{X} \mapsto \mathbb{R}$ with $\|f\|_{\alpha,\infty} < r$. Due to smoothing, $v_N \in \mathbb{C}_r^\alpha(\mathcal{Z})$, for some $r < \infty$, if u and F_Z are sufficiently smooth (see Theorem 1). We can therefore import existing results for approximation methods for functions in $\mathbb{C}_r^\alpha(\mathcal{Z})$:

Example 2. Polynomial interpolation using tensor products. Suppose we use Jth order Chebyshev interpolation with $M \geq J$ nodes in each of the d_z dimensions, or a Jth order B-spline interpolation with $M \geq J$ number of nodes in each of the d_z dimensions (see Appendix C for their precise expressions). Let $p_1, \ldots, p_J$ denote the J polynomials; we then have

$$B_K(z) = \left\{p_{j_1}(z_1) \cdots p_{j_{d_z}}\left(z_{d_z}\right) : j_1, \ldots, j_{d_z} = 1, \ldots, J\right\},$$

which is of dimension $K = J^{d_Z}$. Choosing $J \geq \underline{\alpha}$, where $\underline{\alpha} \geq 1$ denotes the number of derivatives of $v(z)$, both interpolation schemes satisfy, for any radius $r < \infty$,

$$\sup_{v\in\mathbb{C}_r^\alpha(\mathcal{Z})} \left\|\bar{\Pi}_K(v) - v\right\|_\infty = O\left(\frac{\log(J)}{J^\alpha}\right) = O\left(\frac{\log(K)}{K^{\alpha/d}}\right);$$

see p. 14 in Rivlin (1990) for Chebyshev interpolation and Schumaker (2007) for B-splines. If $v(z)$ is analytic ($\underline{\alpha} = \infty$), the above result holds for any (large) $J, \alpha < \infty$.

As can be seen from the above example, standard polynomial tensor product approximations suffer from the well-known computational curse of dimensionality: To reach a given level of error tolerance, the total number of basis functions K has to grow exponentially as d_Z increases. This issue can be partially resolved by using more advanced function approximation methods:

Example 3. Interpolation with sparse grids. Instead of using tensor-product basis functions to approximate a given function, where the total number of basis function and interpolation points will have to grow exponentially with d_Z to

control the approximation error, one can instead use so-called Smolyak sparse grids; see, e.g., Judd et al. (2014) and Brumm and Scheidegger (2017). Using these, the number of grid points needed to obtain a given error tolerance is reduced from $O\left(M^{d_Z}\right)$ to $O\left(M\left(\log M\right)^{d_Z}\right)$ with only slightly deteriorated accuracy.

Example 4. Variable selection, shape constraints, shrinkage estimators, and machine learning. An alternative way of breaking the curse of dimensionality appearing in Example 2 is to select the basis functions judiciously. This could, for example, be done using standard variable selection methods; one example of this approach can be found in Chen (1999). Alternatively, one can in some cases show that the value functions satisfy certain shape constraints that can then be imposed on the sieve; see, for example, Cai and Judd (2013). Other automated selection methods include shrinkage methods where a penalization term is added to the least-squares criterion. Again this leads to a more sparse representation which is able to break the curse-of-dimensionality. Finally, machine learning algorithms, such as neural networks, may potentially be useful in approximating the value functions; see, for example, Chen and White (1999). On the other hand, these methods are generally computationally more expensive compared to the least-squares projection method in (4.7) and require that the value function satisfies certain sparsity. We will investigate the performance of such more advanced projection operators in future work.

As noted earlier, there is no guarantee that a given function approximator is non-expansive. But this can, in principle, be examined numerically for a given choice of $\bar{\Pi}_K$. For the least-squares projection, this amounts to solving, for a given choice of basis functions and grid points,

$$\left\|\bar{\Pi}_K\right\|_{op,\infty} = \sup_{v\in\mathbb{R}^M,\|v\|=1}\sup_{z\in\mathcal{Z}}\left|B_K(z)'\left[\sum_{i=1}^M B_K(z_i)B_K(z_i)'\right]^{-1}\sum_{i=1}^M B_K(z_i)v_i\right|. \tag{4.9}$$

When M and/or $\dim\mathcal{Z}$ is large this may be computationally demanding and instead one can obtain a lower bound by restricting z to only take values on the chosen set of grid points: With $\mathbf{B}_{K,M}\in\mathbb{R}^{K\times M}$ containing the basis functions evaluated at the grid points, we can represent $\bar{\Pi}_K$ when only evaluated at chosen grid points $z_1,\ldots,z_M$ in terms of

$$\mathbf{P}_{K,M} = \mathbf{B}_{K,M}'\left[\mathbf{B}_{K,M}\mathbf{B}_{K,M}'\right]^{-1}\mathbf{B}_{K,M}\in\mathbb{R}^{M\times M}.$$

In particular, it is easily checked that with the supremum in (4.9) being only taken over $z\in\{z_1,\ldots,z_M\}$, $\left\|\bar{\Pi}_K\right\|_{op,\infty} = \left\|\mathbf{P}_{K,M}\right\|_{op,\infty}$. Furthermore, $\left\|\mathbf{P}_{K,M}\right\|_{op,\infty}\leq 1$ if and only if

$$\max_{i=1,\ldots,M}\sum_{j=1}^M\left|p_{ij}\right|\leq 1,$$

where p_{ij} is the (i,j)th element of $\mathbf{P}$, cf. Lizotte (2011).

4.3.4. Solving for the approximate value functions

Computing the simulated self-approximating solution or the projection-based one can be done using three different numerical algorithms: Successive approximation (SA), Newton–Kantorovich (NK), or a combination of the two. The latter corresponds to the hybrid solution method proposed in Rust (1988). We here discuss the implementation of these algorithms with focus on the sieve-based approximation of v; the implementations of the sieve approximation of V and the self-approximating solutions of either of the two follow along the same lines. The main difference between solving for V or v is that the latter involves smaller computational burden since it is a scalar function while the former is a D-dimensional vector function.

SA utilizes that (for K chosen large enough), $\bar{\Pi}_K\bar{\Gamma}_{N,\lambda}$, is a contraction mapping which guarantees that the following algorithm will converge towards the solution to (4.2),

$$\hat{v}_N^{(k)} = \bar{\Pi}_K\bar{\Gamma}_N(\hat{v}_N^{(k-1)}), \tag{4.10}$$

for $k=1,2,\ldots$, given some initial guess $\hat{v}_N^{(0)}$. In the leading case of (4.7), this can be expressed as a sequence of least-squares problems that are easily computed: $\hat{v}_N^{(k)}(z)=\hat{\alpha}_k'B_K(z)$ where

$$\hat{\alpha}_k = \left[\sum_{i=1}^M B_K(z_i)B_K(z_i)'\right]^{-1}\sum_{i=1}^M B_K(z_i)\bar{\Gamma}_{N,\lambda}(\hat{\alpha}_{k-1}'B_K)(z_i)'\in\mathbb{R}^K,$$

for $k=1,2,\ldots$, given some initial guess $\hat{\alpha}_0$. In the case where $\varepsilon_t=\emptyset$ or when the model is of the form of Eq. (3.9) with $\varepsilon_t^{(1}$ being extreme-valued distributed and $\varepsilon_t^{(-1)}=\emptyset$,

$$\bar{\Gamma}_N(\alpha'B_K)(z_i;\lambda) = G_\lambda\left(u(z)+\beta\alpha'\sum_{j=1}^N B_K\left(Z_j(z_i)\right)w_{Z,N,j}(z_i)\right),$$

and so $\sum_{j=1}^{N} B_K\left(Z_j(z_i)\right) w_{z,N,j}(z_i)$, $i = 1, \ldots, M$, only need to be computed once and then recycled in each iteration; in contrast, the simulated averages appearing in $\Gamma_N(\hat{\alpha}_{k-1}' B_K)(z_i)$, $i = 1, \ldots, M$, have to be recomputed in each step of the SA algorithm. Thus, in this special case, it is faster to (approximately) solve for $v_{N,\lambda}$ instead of $V_{N,\lambda}$. While SA is guaranteed to converge globally when $\bar{\Pi}_K \bar{\Gamma}_N$ is a contraction, the rate of convergence will be slow with the error vanishing at rate β^k,

$$\left\| \hat{v}_N^{(k)} - v_N \right\|_\infty \leq \frac{\beta^k (1+\beta)}{1-\beta} \left\| \hat{v}_N^{(0)} - v_N \right\|_\infty . \tag{4.11}$$

To speed up convergence, we therefore follow Rust (1988) and combine SA with NK iterations since NK converges with a quadratic rate once a given guess of the value function is close enough to the fixed point. Moreover, in situations where $\bar{\Pi}_K \bar{\Gamma}_N$ is expansive, NK is still guaranteed to converge locally. Since both the self-approximating and sieve-based methods solve finite-dimensional problems, the NK algorithm for these are equivalent to Newton's method. First consider the sieve-based method where we focus on the least-squares projection as given in (4.7). We are then seeking $\hat{\alpha}$ solving the following K equations,

$$\bar{S}_{N,K}(\alpha; \lambda) = 0,$$

with

$$\bar{S}_{N,K}(\alpha; \lambda) = \alpha - \left[\sum_{i=1}^{M} B_K(z_i) B_K(z_i)' \right]^{-1} \sum_{i=1}^{M} B_K(z_i) \bar{\Gamma}_N\left(\alpha' B_K\right)(z_i; \lambda) .$$

The corresponding derivatives of the left-hand side as a function w.r.t. α can be expressed in terms of the Hadamard differential of $\bar{\Gamma}_N$ w.r.t. v,

$$\begin{aligned} \nabla \bar{\Gamma}_N(v)[dv](z; \lambda) = \beta \sum_{d \in \mathcal{D}} \sum_{j=1}^{N} \dot{G}_{d,\lambda} \left(u\left(z, \varepsilon_j(z)\right) + \beta \sum_{i=1}^{N} v\left(Z_i(z); \lambda\right) w_{Z,N,i}(z) \right) \\ \times \left(\sum_{k=1}^{N} dv\left(Z_k(z, d); \lambda\right) w_{Z,N,k}(z, d) \right) w_{\varepsilon,N,j}(z) , \end{aligned}$$

where $dv : \mathcal{Z} \times \left[0, \bar{\lambda}\right] \mapsto \mathbb{R}$ is the direction and

$$\dot{G}_{\lambda,d}^{(r)}(r) = \frac{\partial G_\lambda(r)}{\partial r(d)} = \frac{\exp\left(\frac{r(d)}{\lambda}\right)}{\sum_{d' \in \mathcal{D}} \exp\left(\frac{r(d')}{\lambda}\right)}. \tag{4.12}$$

The partial derivatives of $\bar{S}_{N,K}(\alpha; \lambda)$ then become

$$\bar{H}_{N,K}(\alpha; \lambda) = I_K - \left[\sum_{i=1}^{M} B_K(z_i) B_K(z_i)' \right]^{-1} \sum_{i=1}^{M} B_K(z_i) \nabla \bar{\Gamma}_N\left(\alpha' B_K\right)[B_K](z_i; \lambda)' \in \mathbb{R}^{K \times K}.$$

With these definitions, the NK algorithm takes the form

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} - \bar{H}_{N,K}^{-1}\left(\hat{\alpha}_{k-1}; \lambda\right) \bar{S}_{N,K}\left(\hat{\alpha}_{k-1}; \lambda\right).$$

The NK algorithm for the self-approximating method is on the same form, except that we now solve directly for the value function at the N draws. With slight abuse of notation, let $v_N = \left\{ v_{N,\lambda}(Z_k; \lambda) : k = 1, \ldots, N \right\}$ be the vector of integrated values across the set of draws solving $\bar{S}_N(v_N; \lambda) = 0$ where

$$\bar{S}_{N,k}(v_N; \lambda) = v_{N,k} - \sum_{j=1}^{N} G_\lambda \left(u(Z_k, \varepsilon_j) + \beta \sum_{i=1}^{N} v_{N,i} w_{Z,N,i}(Z_k) \right) w_{\varepsilon,N,j}(Z_k) , \tag{4.13}$$

for $k = 1, \ldots, N$. The corresponding derivatives are $\bar{H}_N(\alpha; \lambda) = \left(\bar{H}_{N,1}(\alpha; \lambda), \ldots, \bar{H}_{N,N}(\alpha; \lambda)\right)' \in \mathbb{R}^{N \times N}$ where, with $\mathbf{1}_N = (1, \ldots, 1)' \in \mathbb{R}^N$,

$$\bar{H}_{N,k}(\alpha; \lambda) = I_N - \nabla \bar{\Gamma}_N(v_N)[\mathbf{1}_N](z_k; \lambda) \in \mathbb{R}^N.$$

Finally, we note that the NK algorithm for the expected value function takes a similar form with the functional differential of Γ_N given by

$$\nabla \Gamma_N(V)[dV](z; \lambda) = \beta \sum_{d \in \mathcal{D}} \sum_{i=1}^{N} \dot{G}_{\lambda,d}^{(r)}(r) \left(u\left(S_i(z)\right) + \beta V\left(Z_i(z); \lambda\right) \right) dV\left(Z_i(z), d\right) w_{N,i}(z) , \tag{4.14}$$

where $dV(z) = \left(dV(z, 1), \ldots, dV(z, D)\right)'$.

Comparing the NK algorithm for the self-approximating and the sieve-based method, we note that the former involves inverting a $N \times N$-matrix while the latter a $K \times K$-matrix. As pointed out earlier, the self-approximating method generally needs N to be chosen quite large to achieve a precise simulated version of the Bellman operator, in particular in higher dimensions, and so the NK algorithm for this method may become numerically infeasible in some cases. While the projection-based method also suffers from a curse of dimensionality, since the number of basis functions, K, has to be quite large in higher dimensions to achieve a reasonable approximation, it is less severe and is implementable for higher-dimensional models. If more advanced function approximation methods are employed, even better performance can be achieved.

5. Theory

We here develop an asymptotic theory for the self-approximating and sieve-based methods. We first establish some important properties of the smoothed simulated Bellman operators and their exact solutions, v_N and V_N defined in (4.1). These are then used in the asymptotic analysis of the self-approximating solution method and the sieve-based one. This analysis will rely on two general results for estimated solutions to fixed point problems as stated in Theorems A.1 and A.2 in the appendix. The asymptotic analysis will mostly focus on V_N and $\hat{V}_N$ since our results for these easily translate into similar results for the approximate integrated value function. For example, $v_N(z;\lambda) = M_{N,u}(\beta V_N(z;\lambda)|z;\lambda)$, where

$$M_{N,u}(r|z;\lambda) = \sum_{j=1}^{N} G_\lambda\left(u(z,\varepsilon_j(z)) + r\right) w_{\varepsilon,N,j}(z),$$

and so the asymptotic results for V_N in conjunction with the functional Delta method can be used to obtain similar results for v_N.

Without loss of generality, we assume that the draws can be written as

$$Z_i(z) = \psi_Z(U_i;z) \in \mathcal{Z},\ \varepsilon_i(z) = \psi_\varepsilon(U_i;z) \in \mathcal{E}, \tag{5.1}$$

for some i.i.d. draws $U_i \sim P_U$, $i = 1,\ldots,N$, and some functions ψ_Z and ψ_ε. We then define, with $\psi = (\psi_Z, \psi_\varepsilon)$ and for any given function $V(z,\lambda)$,

$$u_\psi(U;z) := u(\psi(U;z)),\ w_\psi(U;z) = w(\psi(U;z)|z),\ V_\psi(U;z,\lambda) = V(\psi_Z(u;z),\lambda) \tag{5.2}$$

so that

$$\Gamma(V)(z,\lambda) = E\left[G_\lambda\left(u_\psi(U;z) + \beta V_\psi(U;z,\lambda)\right) w_\psi(U;z)\right] \tag{5.3}$$

where expectations are taking over $U \sim P_U$, and

$$\Gamma_N(V)(z,\lambda) = \sum_{i=1}^{N} G_\lambda\left(u_\psi(U_i;z) + \beta V_\psi(U_i;z,\lambda)\right) w_{\psi,N}(U_i;z), \tag{5.4}$$

where $w_{\psi,N}(U_i;z,d) = w_\psi(U_i;z,d)/\sum_{j=1}^{N} w_\psi\left(U_j;z,d\right)$. Here, and in the following, we let $V_0(z,\lambda)$ denote the exact solution to

$$V_0(z,\lambda) = \Gamma(V_0)(z,\lambda) \tag{5.5}$$

As explained earlier, we here define the two operators to take a given function $V(z,\lambda)$, $(z,\lambda) \in \mathcal{Z} \times \left[0,\bar{\lambda}\right]$, for some given $\bar{\lambda} > 0$, and map them into another function with domain $\mathcal{Z} \times \left[0,\bar{\lambda}\right]$. In particular, $V_0(z,0)$ and $V_N(z,0)$ are the non-smoothed ($\lambda = 0$) exact and simulated solutions, respectively. We then impose the following regularity conditions on the model and chosen importance sampler, where we recall the function norm defined in Eq. (4.8) and the function set $\mathbb{C}_r^\alpha(\mathcal{Z})$ defined below this equation:

Assumption 1. The support $\mathcal{Z}$ is a compact set; $\bar{u}_\psi(u) := \sup_{z\in\mathcal{Z}} \left\|u_\psi(u;z)\right\|$ and $\bar{w}_\psi(u) := \sup_{z\in\mathcal{Z}} w_\psi(u;z)$ satisfy $E\left[\bar{u}_\psi^2(U)\,\bar{w}_\psi^2(U)\right] < \infty$.

Assumption 2. For some $\alpha > 0$, $z \mapsto u_\psi(U;\cdot)$ and $z \mapsto w_\psi(U;z)$ belong to $\mathbb{C}_\infty^\alpha(\mathcal{Z})$ P_U-almost surely with $E\left[\|u_\psi(U;\cdot)\|_{\alpha,\infty}^2 \|w_\psi(U;\cdot)\|_{\alpha,\infty}^2\right] < \infty$.

Assumption 1 shares some similarities with the regularity conditions found in Rust (1988) who considered an additive version of our general model. Importantly, we only require that Z_t has bounded support while ε_t can have potentially unbounded support. This is in contrast to Pal and Stachurski (2013) and Rust (1997b) who require both components to be bounded. We conjecture that the subsequent results can be generalized to also hold in the case of $\mathcal{Z}$ unbounded but then our conditions and arguments would have to be changed. For example, the existence of unique fixed points would have to be verified in a function space equipped with a weighted sup-norm and with additional moment conditions on Z_t,

see, e.g., Norets (2010). Similarly, our empirical process results would need to be established using bracketing conditions with weighted norms and additional moment conditions, see, e.g., Section 2.10.4 in van der Vaart and Wellner (1996). Similar to the results in Rust (1988), Assumption 1 implies $\Gamma(V) \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ for all $V \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$, see below. This particular result actually holds under the weaker requirement that $E\left[\|u_\psi(U;\cdot)\|_{0,\infty}\|w_\psi(U;\cdot)\|_{,\infty}\right] < \infty$ but the existence of the second order moment is needed for the subsequent asymptotic analysis of $V_N(z;\lambda)$ and so we impose this restriction throughout.

Assumption 2 imposes smoothness conditions on the model and the chosen samplers in terms of the state variables Z_t. These conditions imply that the expected and integrated value functions will be smooth too. Note here that the degree of smoothness α is left unrestricted at this stage and so the functions are not required to be differentiable, merely Lipschitz. While the focus is on models with continuous state variables our theory also covers models with discrete state space. In this case, we can dispense of Assumption 2 and instead rely on the more general Theorem B.1 which implies that the subsequent results still go through when Z_t is discrete.

We first establish existence and uniqueness of the (generally) infeasible simulated solutions and show that they inherit the smoothness properties of $z \mapsto u_\psi(U;\cdot)$ and $z \mapsto w_\psi(U;z)$. This feature of the approximate solutions is important for two reasons: First, it allows us to show uniform convergence of certain functionals as part of our proof of weak convergence. Second, we can control the approximation error due to the use of sieves later on.

Theorem 1. *Suppose Assumption 1 holds and, for a given $N \geq 1$, $\inf_{z\in\mathcal{Z}} \sum_{i=1}^N w_\psi(U_i;z) > 0$. Then the operators Γ and Γ_N in Eqs. (5.3) and (5.4) are almost surely contraction mappings on $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ and so $V_0 : \mathcal{Z} \times \left(0, \bar{\lambda}\right) \mapsto \mathbb{R}^D$ and $V_N : \mathcal{Z} \times \left(0, \bar{\lambda}\right) \mapsto \mathbb{R}^D$ exist and are unique. If furthermore Assumption 2 holds then $V_0 \in \mathbb{C}^\alpha_{r_0}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)$ for some constant $r_0 < \infty$ while $V_N \in \mathbb{C}^\alpha_{r_N}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)$ for some $r_N < \infty$ P_U-almost surely.*

The above is a fixed N result with the bound on V_N, r_N, being random since it depends on the particular set of draws. We derive a deterministic bound on r_N as $N \to \infty$ below. The condition $\inf_{z\in\mathcal{Z}} \sum_{i=1}^N w_\psi(U_i;z) > 0$ will hold with probability approaching 1 (w.p.a.1) as $N \to \infty$ and so can be dropped in our asymptotic analysis. Next, we analyze the effect of smoothing on the exact and simulated value function:

Theorem 2. *Under the conditions of Theorem 1, the following hold: $\|V_0(\cdot;\lambda) - V_0(\cdot;0)\|_\infty = O(\lambda)$ and $\|V_N(\cdot;\lambda) - V_N(\cdot;0)\|_\infty = O_P(\lambda)$ for any given $N \geq 1$.*

This shows that the smoothing can be controlled for by suitable choice of λ both asymptotically ($N = +\infty$) and for any finite number of simulations ($N < \infty$). Also note that the above result holds independently of the smoothness properties of the unsmoothed exact and simulated solutions.

5.1. Self-approximating method

In this section, we provide an analysis of the smoothed fixed point, V_N, to Γ_N defined in Eq. (5.4) thereby allowing for general importance samplers. As a special case, we obtain an asymptotic theory for the self-approximating solution method (where $\Phi_z\left(z'|z,d\right)$ is restricted to be marginal distribution). The results for V_N will then in turn be used in the analysis of the corresponding sieve-based methods in the next section.

Theorem 3. *Suppose Assumptions 1–2 hold for some $\alpha > 0$. Then V_N solving $\Gamma_N(V_N) = V_N$ satisfies $\|V_N - V_0\|_\infty = O_P(1/\sqrt{N})$.*
If Assumption 2 holds with $\alpha \geq 1$, then $V_0, V_N \in \mathbb{C}^1_r\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)$ w.p.a.1 for some constant $r < \infty$. Moreover, $\sup_{z\in\mathcal{Z}} \left\| \partial V_N(z,\lambda)/\left(\partial z_j\right) - \partial V_0(z,\lambda)/\left(\partial z_j\right)\right\| = O_P\left(\sqrt{N}/\lambda\right)$ uniformly over $\lambda \in \left(0, \bar{\lambda}\right)$.

The first part of this theorem is similar to results found in Rust (1997b) and Pal and Stachurski (2013) who also show $\sqrt{N}$-convergence of their value function approximation. Importantly, the convergence result holds uniformly over the smoothing parameter λ and so there is no first-order effect from smoothing if λ vanishes sufficiently fast. Specifically, for any sequence λ_N satisfying $\sqrt{N}\lambda_N \to 0$, Theorems 2 and 3 yield $\sup_{z\in\mathcal{Z}} \|V_N(z,\lambda_N) - V(z,0)\| = O_p(1/\sqrt{N})$. This is similar to convergence of smoothed empirical cdf where the indicator function is replaced by a smoothed version; this also does not affect the convergence rate as long as the smoothing bias is controlled for. The second part of the theorem appears to be a new result and shows that if the problem is smooth enough, the first-order partial derivatives of $V_N(z,\lambda)$ also converge uniformly over z with rate $\sqrt{N}/\lambda$. Since we need $\lambda \to 0$ to kill the smoothing bias, this could seem to imply that the first-order derivatives converge with slower than $\sqrt{N}$-rate. However, we conjecture that the derived rate is not sharp and that $\sqrt{N}$-convergence does actually hold. The proof of this appears to require a more delicate and refined arguments, however, and so we leave this for future research.

The above result is then in turn used to derive the asymptotic distribution of $V_N(z,\lambda)$ uniformly in $(z,\lambda) \in \mathcal{Z} \times \left(0, \bar{\lambda}\right)$. Here, the smoothing proves important since it allows us to generalize the standard arguments used in the analysis of finite-dimensional extremum estimators to our setting: We first expand the "first-order condition", $V_N - \Gamma_N(V_N) = 0$, around $V_0 = \Gamma(V_0)$ to obtain, with $\nabla\Gamma_N$ defined in (4.14),

$$0 = \Gamma(V_0) - \Gamma_N(V_0) + \{I - \nabla\Gamma_N(V_0)\}[V_N - V_0] + o_P\left(1/\sqrt{N}\right),$$

where the rate of the remainder term follows from Theorem 3. Next, employing empirical process theory, we show that $\sqrt{N}\{\Gamma(V_0) - \Gamma_N(V_0)\} \rightsquigarrow \mathbb{G}$ in $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ for a Gaussian process $\mathbb{G}(z, \lambda)$ with covariance kernel

$$\Omega(z_1, \lambda_1, z_2, \lambda_2) = E_U\left[g(U; z_1, \lambda_1)\, g(U; z_2, \lambda_2)'\right], \tag{5.6}$$

$$g(U; z, \lambda) = \left\{G_\lambda\left(u_\psi(U; z) + \beta V_0(\psi_Z(U; z), \lambda)\right) - \Gamma(V_0)(z, \lambda)\right\} w_\psi(U; z). \tag{5.7}$$

Finally, we show that $\nabla\Gamma_N(V_0)[dV] \to^P \nabla\Gamma(V_0)[dV]$ uniformly over $dV \in \mathbb{C}^1_{2r}\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)^D$ with $r < \infty$ given in Theorem 3. Since $V_N - V_0 \in \mathbb{C}^1_{2r}\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)^D$, we conclude that:

Theorem 4. *Suppose Assumptions 1 and 2 hold with $\alpha \geq 1$. Then, $\sqrt{N}\{V_N - V_0\} \rightsquigarrow \mathbb{G}_V$ on $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ where $\mathbb{G}_V(z, \lambda) = \{I - \nabla\Gamma(V_0)\}^{-1}[\mathbb{G}](z, \lambda)$ is a D-dimensional Gaussian process.*

The above result implies, for example, that $\sqrt{N}\{V_N(z, \lambda) - V_0(z, \lambda)\} \to^d N(, \Omega_V(z, \lambda, z, \lambda)/N)$ as $N \to \infty$ for any given (z, λ), where

$$\Omega_V(z_1, \lambda_1, z_2, \lambda_2) = \iint r^*\left(z_1', \lambda_1' | z_1, \lambda_1\right) \Omega\left(z_1', \lambda_1', z_2, \lambda_2\right) r^*\left(z_2', \lambda_2' | z_2, \lambda_2\right)' d\left(z_1', \lambda_1'\right) d\left(z_2', \lambda_2'\right),$$

where r^* is the Riesz representer of $dV \mapsto \{I - \nabla\Gamma(V_0)\}^{-1}[dV](z, \lambda)$. Thus, it allows us to construct (pointwise or uniform) confidence bands for the expected value function. We expect that the result will also be useful in analyzing the impact of value function approximation when used in estimation. This could be done by combining the above weak convergence result with, e.g., the results for approximate estimators found in Kristensen and Salanie (2017).

The proof of Theorem 4 proceeds by verifying the two high-level conditions of the "master" Theorem B.1 where the same weak convergence result is obtained under more general conditions. Theorem B.1 allows us to replace the smoothness conditions in Assumption 2 with some other conditions implying that $V_N - V_0$ is situated in a function set with finite entropy. One example would be to impose restrictions on u and w so that the value function and its estimator are both monotone functions, cf. Pal and Stachurski (2013), in which case we could then appeal to Theorem 2.7.5 in van der Vaart and Wellner (1996) to obtain the results of Theorem 4.

We conjecture that a similar weak convergence result will hold for the non-smoothed value function approximation ($\lambda = 0$). However, the proof of such a result would require different arguments and seemingly stronger assumptions. In particular, the current proof only requires the empirical process $(z, \lambda) \mapsto \Gamma_N(V_0)(z, \lambda)$ to converge weakly. To allow for non-smooth value function approximation, we conjecture that we would now need to show that the empirical process $(V, z) \mapsto \Gamma_N(V)(z, 0)$ converges weakly over a suitable function set that the estimated non-smooth solution, V_N, would be situated in. For this to hold, the uniform entropy of the function set would need to be finite. Standard choices of function sets are smooth classes, but V_N and its limit V_0 are both non-smooth now and so the proof appears to be rather delicate.

Finally, for a complete analysis that takes into account the smoothing bias, we state the following corollary to Theorem 4: For any $\lambda_N \to 0$ such that $\lambda_N\sqrt{N} \to 0$, $\sqrt{N}\{V_N(\cdot; \lambda_N) - V_0(\cdot; 0)\} \rightsquigarrow \mathbb{G}_V(\cdot, 0)$.

5.2. Sieve-based approximation of value functions

We now proceed to analyze the asymptotic properties of the sieve-based approximate value function, $\hat{V}_N$. To this end, we use the following decomposition of the over-all error,

$$\hat{V}_N - V_0 = \left\{\hat{V}_N - V_N\right\} + \{V_N - V_0\}, \tag{5.8}$$

where the second term converges weakly towards a Gaussian process, cf. Theorem 4. What remains is to control the first term which is due to the sieve approximation; this is done by imposing the following high-level assumption on the projection operator when applied to a function set $\mathcal{V}$ which is chosen so that $V_N \in \mathcal{V}$ w.p.a.1.:

Assumption 3. The projection operator Π_K satisfies $\sup_{v \in \mathbb{C}^\alpha_{r_0}(\mathcal{Z})} \|\Pi_K(v) - v\|_\infty = O_P(\rho_K)$ for some sequence $\rho_K \to 0$, where α is given in Assumption 2 and $r_0 < \infty$ in Theorem 1.

This is a high-level condition that requires the chosen function approximation method to have a uniform error rate over the function class $\mathbb{C}^\alpha_r(\mathcal{Z})$ which we know $z \mapsto V_0(z, \lambda)$ belongs uniformly in λ under Assumption 2. As discussed earlier, one could replace Assumption 2 with other regularity conditions that ensure $z \mapsto V_0(z, \lambda)$ is sufficiently regular (e.g., monotonic) in which case $\mathbb{C}^\alpha_r(\mathcal{Z})$ in Assumption 3 should be modified accordingly. Assumption 3 is satisfied for standard polynomial approximators with $\rho_K = \log(K)/K^{(s+1)/d}$, cf. Section 4.3. Compared to results on sieve approximations of value functions found elsewhere in the literature, our rate is better since we are here seeking to approximate the expected value function that is situated in $\mathbb{C}^\alpha_r(\mathcal{Z})$. In contrast, sieve-based approximations developed in other papers, such as Munos and Szepesvari (2008) and Pal and Stachurski (2013), try to approximate the value function which is at most Lipschitz and for such functions the approximation error will be larger in general. In the case of $\mathcal{Z}$ being

finite, we have $\sup_{V\in\mathcal{V}} \|\Pi_K(V) - V\|_\infty = 0$ for $K > |\mathcal{Z}|$ under great generality and so there will be no asymptotic bias component due to sieve approximations in this case.

The second part of Theorem A.1 together with the fact that $\Gamma_{N,\lambda}(V_\lambda) - \Gamma_\lambda(V_\lambda) = O_P\left(1/\sqrt{N}\right)$, cf. Proof of Theorem 3, now yields the following result:

Theorem 5. *Suppose Assumptions 1–3 hold. Then $\hat{V}_N$, defined as the solution to $\Pi_K\Gamma_N(\hat{V}_N) = \hat{V}_N$, satisfies $\|\hat{V}_N - V_0\|_\infty = O_p(1/\sqrt{N}) + O_P(\rho_K)$. Suppose in addition that $\alpha \geq 1$ in Assumption 2. Then, if $\sqrt{N}\rho_K \to 0$, $\sqrt{N}\{\hat{V}_N - V_0\} \rightsquigarrow \mathbb{G}_V$.*

The discussions following Theorems 3 and 4 carry over to the above result. In particular, the rate result still goes through when no smoothing is employed ($\lambda = 0$) but the current proof of the asymptotic distribution result requires smoothing ($\lambda > 0$). Compared to the rate results for $V_{N,\lambda}$, the projection-based method suffers from an additional error due to the sieve approximation, $O_P(\rho_K)$. This can be interpreted as a bias term, while $O_p(1/\sqrt{N})$ is its variance component which is shared with $V_{N,\lambda}$. The requirement that $\sqrt{N}\rho_K \to 0$ is used to kill the sieve bias term so that $\hat{V}_N$ is centered around V_0.

The above result provides a refinement over existing results where a precise rate for the bias is not available; see, e.g., Lemma 5.2 in Pal and Stachurski (2013). It shows that there is an inherent computational curse-of-dimensionality built into our projection-based value function approximation when polynomial interpolation is employed: In high-dimensional models, a large number of basis functions are needed which in turn increases the computational effort. In the case of polynomial approximations, the rate condition becomes $\sqrt{N}\log(K)/K^{(s+1)/d} \to 0$ and so, as d increases, we need K to increase faster with N to kill the sieve bias component. However, also note that K has no first-order effect on the variance and so there is no bias–variance trade-off present. In particular, we can let K increase with N as fast as we wish and so our procedure should in principle also work for models with high-dimensional state space. That is, we can achieve $\sqrt{N}$-rate regardless of the dimension of the problem and so our method does not suffer from any statistical curse-of-dimensionality. However, this requires choosing K large enough in order to control the sieve approximation bias which will increase computation time as the dimension grows. Thus, there is a potential computational curse-of-dimensionality.

6. Numerical results

In this section we examine the numerical performance of the proposed solution algorithms with focus on how the theoretical results derived in the previous sections translate into practice and how different features of model and implementation affect their performances.

We focus exclusively on approximating the integrated value function, $v(z)$, and measure the performance of a given approximate solution, say, $\tilde{v}(z)$ in terms of its pointwise bias, variance and mean-square error (MSE) defined as $Bias(z) := E[\tilde{v}(z)] - v(z)$, $Var(z) := Var(\tilde{v}(z)) = E\left[(\tilde{v}(z) - E[\tilde{v}(z)])^2\right]$ and $MSE(z) = Bias^2(z) + Var(z)$, respectively. As overall measures we use uniform bias, variance and MSE, $\|Bias\|_\infty = \sup_{z\in\mathcal{Z}}|Bias(z)|$, $\|Var\|_\infty = \sup_{z\in\mathcal{Z}}|Var(z)|$ and $\|MSE\|_\infty = \sup_{z\in\mathcal{Z}}|MSE(z)|$. Given that the exact solution $v(z)$ is unknown, we replace this by a very precise approximate solution computed in the following way: First, instead of using simulations in the computation of the Bellman operator, we utilize that the state transitions follow a Beta distribution in the chosen model (see below) and so we can use nodes and weights based on Jacobi polynomials to compute it using numerical integration. We then implement the sieve method using $K = 60$ Chebyshev polynomials and $N = 60$ sets of quadrature nodes and weights. The "exact" solution was computed by successive approximation until a contraction tolerance of machine precision was reached. We approximate the point-wise bias and variance of a given method through $S \geq 1$ independent replications of it: Let $\tilde{v}_1(z), \ldots, \tilde{v}_S(z)$ be the solutions obtained across the S replications, where S generally was chosen to 2000. We then approximate the mean by $\hat{E}[\tilde{v}(z)] = \frac{1}{S}\sum_{s=1}^S \tilde{v}_s(z)$ which in turn is used to obtain the following pointwise bias and variance estimates, $\hat{Bias}(z) = \hat{E}[\tilde{v}(z)] - v_0(z)$, and $\hat{Var}(z) = \frac{1}{S}\sum_{s=1}^S(\tilde{v}_s(z) - \hat{E}[\tilde{v}(z)])^2$. Based on these, we approximate $\|Bias\|_\infty$ and $\|Var\|_\infty$ by the maximum pointwise biases and variances over a uniform grid over [0, 1000] of size 500 in the univariate case and a uniform grid over $[0, 1000]^2$ of size 250.

To implement the sieve-based method, we need to choose the sieve space used in constructing Π_K. We here focus on Chebyshev basis functions and B-Splines as discussed in Section 4.3.[1]

6.1. A model of optimal replacement

To provide a test bed for comparison of the sieve-based approximation method, we use the well-known engine replacement model by Rust (1987). Rust's model has become the basic framework for modeling dynamic discrete-choice problems and has been extensively used in other studies to evaluate the performance of alternative solution algorithms and estimators. While the model and its solution is well described in many papers, for completeness we briefly describe our variation of it below.

We consider the optimal replacement of a durable asset (such as a bus engine) whose controlled state $Z_t \in \mathbb{R}_+$ is summarized by the accumulated utilization (mileage) since last replacement. In each period, the decision maker faces the

[1] For more details on their implementation, see Appendix C.

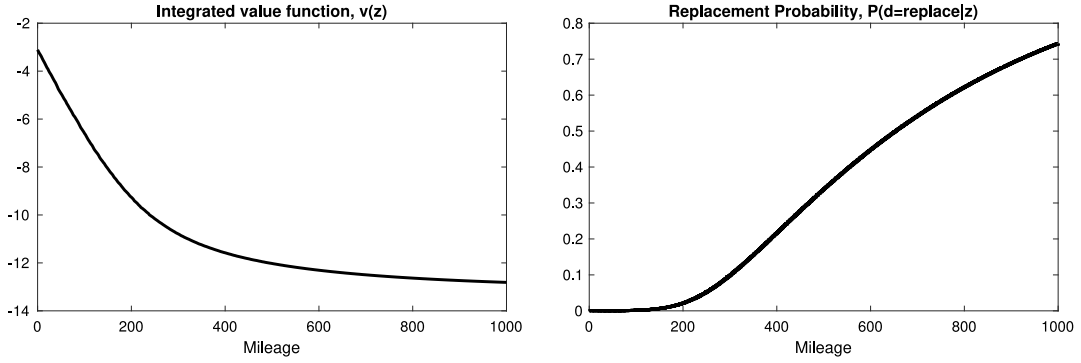

Fig. 1. Fine Approximation as "Exact" Solution. *Notes*: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 1$, $\lambda = 1$ and transition parameters are $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

binary decision $d_t \in \mathcal{D} = \{0, 1\}$ whether to keep ($d_t = 0$) or replace ($d_t = 1$) the durable asset with a fixed replacement cost $RC > 0$. If the asset is replaced, accumulated usage Z_t regenerates to zero. The maintenance/operating costs are assumed to be linear in usage Z_t, $c(Z_t) = \theta_c \cdot 0.001 \cdot Z_t$. The state and decision dependent per period utility is then given by $\bar{u}(Z_t, d_t) + \varepsilon_t(d_t)$ where $\bar{u}(Z_t, d_t) = (RC + c(0)) I\{d_t = 0\} + c(Z_t) I\{d_t = 1\}$ and the utility shocks $\varepsilon_t = (\varepsilon_t(0), \varepsilon_t(1))$ are i.i.d. extreme value and fully independent of Z_t. This specification is a special case of with $\lambda = \sigma_\varepsilon$ and so the simulated Bellman operator takes the form (3.5) where $G_{\sigma_\varepsilon}(\cdot)$ appears as part of the model. Thus, there is no smoothing bias present in the baseline model. In Section 6.5, we investigate the effect of smoothing by pretending that we are not able to integrate out ε_t analytically in the baseline model and instead we simulate both Z_t and ε_t and then include our smoothing device in the computation of the simulated Bellman operator.

We assume that Z_t (in absence of the replacement decision) follows a mixture of a discrete distribution with a probability mass $\pi > 0$ at zero and a linearly transformed Beta distribution with shape parameters a and b and scale parameter $\sigma_\varepsilon > 0$. Thus,

$$F_Z\left(z'|z, d\right) = \pi I\left\{z' = z\right\} + (1 - \pi) F_+\left(z'|z, d\right), \tag{6.1}$$

where $F_+\left(z'|z, d\right)$ has density $f_+\left(z'|z, d\right) = f_\beta\left((z' - z)/\sigma_Z; a, b\right)/\sigma_Z$, $\pi > 0$ is the probability of no usage and $f_\beta(x; a, b)$ is the probability density function of the Beta distribution with shape parameters a, b. Note here that it has bounded support $(0, \sigma_Z)$ so that $f_+\left(z'|z, d\right) = 0$ for $z' < z$ or $z' - z > \sigma_Z$. This is in line with the discretized model in the original formulation in Rust (1987) where monthly mileage were only allowed to take a few discrete values and monthly mileage is naturally bounded above and below (busses never drives backwards and there are limits how far a bus can drive within a month). We introduce probability mass π at $z' = z$ to allow for the possibility that the asset is not used in a given period and thereby can end in the same state with positive probability when $\pi > 0$. As explained below, this feature turns out to be quite important for the applicability of the self-approximating method of Rust (1997). Note that the support of Z_t is unbounded (the positive half line) and therefore the theory does not apply directly, since we throughout assumed bounded support. However, we expect that the theory extends to the unbounded case after suitable modifications, cf. discussion following Assumptions 1–2.

In the numerical illustrations below we use the following set of benchmark parameter values unless otherwise specified: We set replacement cost to $RC = 10$ and the cost function parameter to $\theta_c = 2$ so that RC is 5 times as large as $c(1000)$. This implies a large variation in the probability of replacement over Z_t compared to Rust (1987) and a more curved value function. The parameters indexing the transition density $f_+\left(z'|z, d\right)$ are set to $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 10^{-10}$ as default. This implies a quite sparse transition density, which is similar to the fitted model in Rust (1987). In Fig. 1 we plot the corresponding "exact" solution as described earlier. Importantly, since the transition density is an analytic function the value function is also analytic and so well-approximated by polynomial interpolation methods.

6.2. Numerical implementation of simulated Bellman operators

The simulated Bellman operators in (3.6) and (3.7) require the user to choose an importance sampling distribution. For the self-approximating solution method we need to choose a marginal sampler, $d\Phi_Z\left(z'|z, d\right) = \phi_Z\left(z'\right) dz'$. We follow Rust (1997b) and choose $\phi_Z\left(z'\right) = I\left\{0 < z' < z^{\max}\right\}$ as a uniform density with support $[0, z^{\max}]$ for some truncation point $0 < z^{\max} < \infty$ chosen by us. First note that this entails that the simulated Bellman operator used for the self-approximating value function is biased since we do not sample from the full support $\mathcal{Z} = \mathbb{R}_+$; however, this bias can be controlled by choosing $z^{\max}$ large enough. We will explain below why we do not choose $\phi_Z\left(z'\right)$ as a density with support $\mathbb{R}_+$. Using a uniform sampler, the corresponding Radon–Nikodym derivative takes the form $w_Z\left(z'|z, d\right) = \pi\delta\left(z' - z\right) + (1 - \pi) f_+\left(z'|z, d\right)$ where $\delta(\cdot)$ denotes Dirac's delta function. We approximate this by

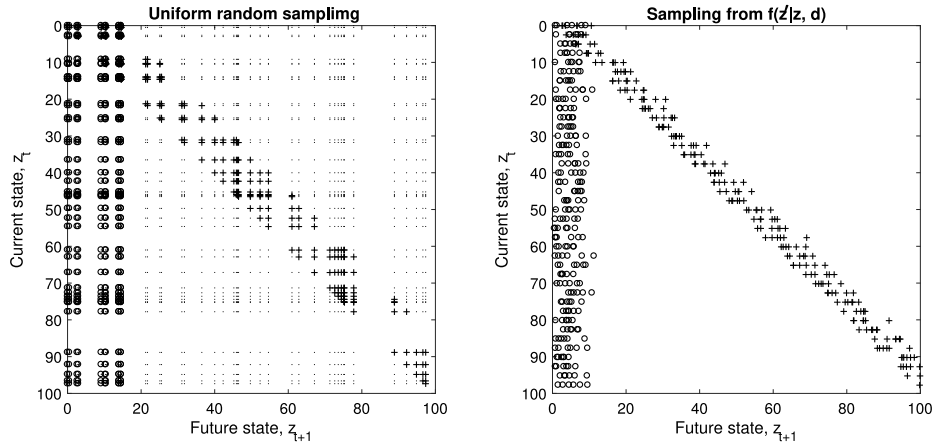

Fig. 2. Random Grids. *Notes*: In the left panel we present the grids used for the self-approximate random Bellman operator. We have uniformly sampled a random grid, $\{Z_1, \ldots, Z_N\}$ on the interval [0; 1000] with $N = 400$. Dots (.) mark sampled grid points in R^2: $Z_N \times Z_N$, plus (+) mark grid points where $f(z_j|z_i, d = 0) > 0$ and circles (o) mark points where $f(z_j|z_i, d = 1) > 0$. In the right panel, we plot the grid the projected random Bellman operator, where we have sampled directly from the conditional transition density in each of the $M = 400$ uniformly spaced evaluation points. To have equally many grid-points with non-zero transition density we only need $N = 400 * \sigma_Z / \max(Z_N) = 9$ random grids for each of the $M = 400$ evaluation points. Both figures show only a subset of the state space, $(z, z') \in [0; 100]^2$. Parameters are $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 0.0000000001$.

$\hat{w}_Z\left(z'|z,d\right) = \pi I\left\{z' = z\right\} + (1-\pi)f_+\left(z'|z,d\right)$ which entails another small approximation error. For the sieve-based version, we simply choose $\Phi_Z\left(z'|z,d\right) = F_Z\left(z'|z,d\right)$ and so $w_Z\left(z'|z,d\right) = 1$.

As explained in Section 4.3, using a marginal importance sampler creates issues since it fails to adapt to the particular shape of the support of $F_Z\left(z'|z,d\right)$. In particular, for a given choice of z, many of the draws from $\phi_Z\left(z'\right)$ will tend to fall outside the support of $f_Z\left(z'|z,d\right)$ and so will not contribute. In contrast, when $\phi_Z\left(z'|z,d\right) = f_Z\left(z'|z,d\right)$, the draws from ϕ_Z will by construction fall within the support of $f_Z\left(z'|z,d\right)$. This can be seen in Fig. 2 where we have plotted the random draws obtained from the two different importance samplers used for the sieve-based and self-approximating solutions together with the actual support of $f_Z\left(z'|z,d\right)$. On the left-hand side panel we have plotted pairs of the uniform draws, $\left(Z_i, Z_j\right)$ for $i, j = 1, \ldots, N$, used for Rust's self-approximating method with $N = 400$ and $z^{max} = 1,000$, while on the right-hand side we have plotted $\left(z_i, Z_j\left(z_i, d\right)\right)$ where z_i are uniform draws and $Z_j\left(z,d\right) \sim f_Z\left(\cdot|z,d\right)$. In both cases, we have marked the pairs for which the corresponding density, $f_Z\left(Z_j|Z_i,d\right)$ and $f_Z\left(Z_j\left(z_i,d\right)|z_i,d\right)$, respectively, is positive. Clearly, the use of a marginal importance sampling density leads to very poor coverage of the actual support of $f_Z\left(z'|z,d\right)$ as z varies while by construction $\Phi_Z\left(z'|z,d\right) = F_Z\left(z'|z,d\right)$ does an excellent job. This translates into the former simulated Bellman operator exhibiting much larger variance compared to the latter.

This issue is further amplified when we introduce the normalization given in Eq. (3.3): Suppose that we had not included a discrete component $\pi I\left\{z' = z\right\}$ in the model. Then, with $Z_i \sim U\left[0, z^{\max}\right]$, $w_{N,Z,i}(Z_j,d) = f_+\left(Z_i|Z_j,d\right) / \sum_{k=1}^N f_+\left(Z_k|Z_j,d\right)$. Since $f_+(z'|z,d)$ has bounded support, it often happens that $\sum_{k=1}^N f_+\left(Z_k|Z_j,d\right) = 0$ for even large values of N and so the simulated Bellman operator is not even well-defined. This issue will vanish as $N \to \infty$, but this on the other hand increases the computational burden since the self–approximating method require us to solve for the value function at the N draws. Introducing the discrete component in the model resolves this issue since now $w_{Z,N,i}(Z_j,d) = \hat{w}_Z\left(Z_i|Z_j,d\right) / \sum_{k=1}^N \hat{w}_Z\left(Z_k|Z_j,d\right)$, where $\sum_{k=1}^N \hat{w}_Z\left(Z_k|Z_j,d\right) > 0$ for all $j = 1, \ldots, N$ by construction. Thus, $\pi > 0$ functions as a regularization device.

Why not choose $\phi_Z\left(z\right)$ as a density with unbounded support in order to avoid the issue of truncation? In our initial experimentation, we did try out sampling from distributions with unbounded support, but the above numerical issues became even more severe in this case since the resulting draws are even more dispersed. Fig. 3 shows how the solution depends on $z^{\max}$. The effect of the truncation $z^{\max}$ will be model specific and in practice experimentation is required. If we, for example, simply set $z^{\max} = 1000{,}000$, the variance of the simulated Bellman operator becomes very large for a given N due to the issue with undefined sample weights $w_{N,i}\left(z,d\right)$ mentioned above. At the same time, choosing $z^{\max}$ too small leads to a large bias. To balance the bias and variance, we ended up using $z^{\max} = 1000$ which all subsequent numerical results for the self-approximating method is based on. Finally, we would like to stress that none of these issues appear for the sieve-based method.

6.3. Convergence properties and computation times

We first investigate the convergence properties of our solution methods for given choice of K and N. Do they converge and if so how fast?

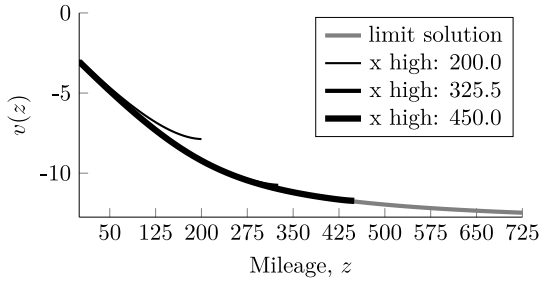

Fig. 3. Truncation bias due to z^{max} being too low.

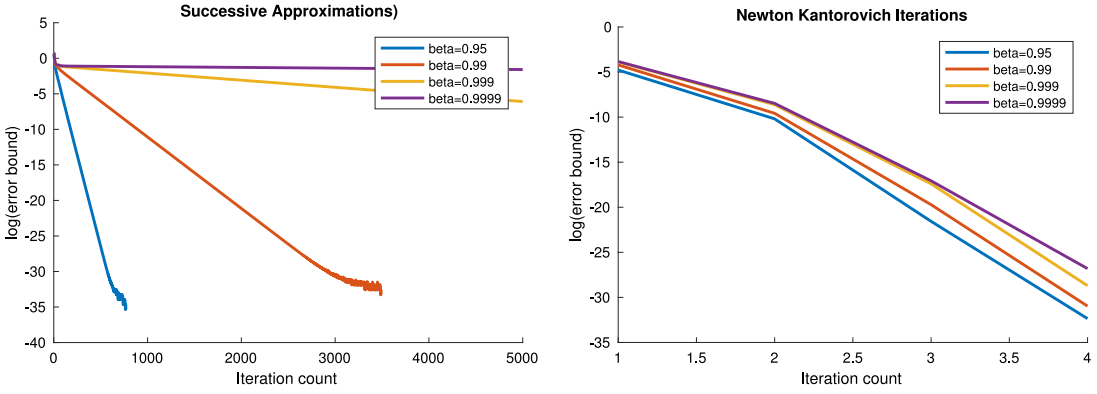

Fig. 4. Convergence and discount factor. *Notes*: Discount factor is $\beta \in \{0.95, 0.99, 0.999, 0.9999\}$, utility function parameters are $\theta_c = 2$, $RC = 1$, $\lambda = 1$ and transition parameters are $\sigma_\varepsilon = 15$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

Global convergence properties of sieve method

As demonstrated in Theorem 1, the simulated Bellman operators are always contraction mappings and so the self-approximating method is guaranteed to converge using successive approximations. In contrast, $\Pi_K \bar{\Gamma}_{N,\lambda}$ is not necessarily a contraction and so global convergence of the sieve method may fail, cf. discussion in Section 5.2. A sufficient condition for global convergence is $\|\Pi_K\|_{op,\infty} < 1/\beta$ and we saw that $\|\mathbf{P}_K\|_{op,\infty} > 1$ implies $\|\Pi_K\|_{op,\infty} > 1$. However, even if $\|\mathbf{P}_K\|_{op,\infty} > 1$, successive approximation may still converge: Across various parameter values of model, choices of sieve spaces and number of simulations, we did not encounter any failure of the sieve method to converge and the resulting approximate solution was well-behaved. This finding held across various initializations of the solution algorithms (initial choice of sieve coefficients). For example, we implemented the sieve method using $M = 64$ evaluation points and using either $K = 1$ or $K = 4$ Chebyshev basis functions. We found that $\|\mathbf{P}_1\|_{op,\infty} = 1$ while $\|\mathbf{P}_4\|_{op,\infty} > 1.78$ and so the sieve method was guaranteed to converge for $K = 1$ but not for $K = 4$. Nevertheless, the method of successive approximations did in fact converge to a tolerance of 10^{-12} for both $K = 1$ and $K = 4$.

Successive approximation versus Newton–Kantorovich

In Section 4.3 we advocated a hybrid of successive approximation (SA) and Newton–Kantorovich (NK) where we start with SA to ensure global convergence, and switch to NK iterations once the domain of attraction has been reached since NK generally converges faster. We illustrate this attractive feature of the NK algorithm in Fig. 4 where we have plotted the log residual error of the current value function approximation (relative to the "exact" solution) against the iteration count for the SA and NK algorithms, respectively, for four different values of β. As expected, the convergence of the SA algorithm requires a very large number of iterations (> 1000) with computation time increasing in β, where as NK converges after less than 10 iterations and with the value of β having little effect on its performance.

Fig. 4 is silent about the over-all computation time of SA relative to NK. Compared to SA, each NK iteration is more expensive since the former only requires computing the simulated Bellman operator evaluated at the value function obtained in the previous step while the latter, in addition, requires computing its functional derivative and inverting a $K \times K$ dimensional matrix for the integrated value function and a $KD \times KD$ dimensional matrix for the expected value function, cf. Section 4.3. With K large, one could therefore fear that NK would become computationally too expensive.

In Fig. 5 we report best of 10 run-times for various levels of K and β and tolerance levels of SA and NK where we also include set-up time (time spent on initial computations before starting the actual algorithm). As expected, we find that NK is the faster of the two algorithms when β is relatively large and K is relatively small. With $K = 5$ NK is faster across all

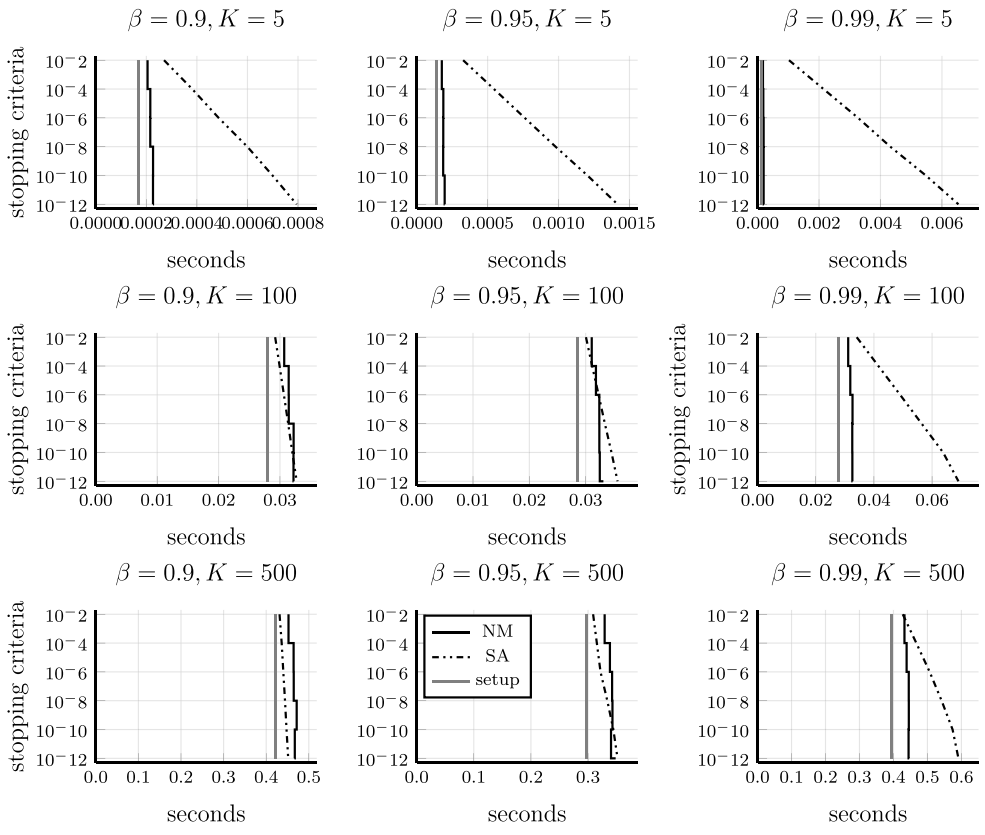

Fig. 5. Run-times (incl setup times) for SA (dotted lines) and NK (drawn lines) algorithms.

levels of β while for $K = 100$ and $K = 500$, SA is faster for moderate values of β. However, as we shall subsequently see, with $K = 5$ the sieve method carries almost no bias and so choosing K larger (such as 100 or 500) is actually unnecessary here and is only included here to illustrate potential issues with NK for models where a large number of sieve terms are needed to obtain a good approximation of the value function. Moreover, in most empirical applications, β is chosen to be larger than 0.99 in which case NK still dominates SA even with $K = 500$.

6.4. Approximation quality

We here investigate how the approximate value function is affected by the number of draws and the chosen projection basis. The goal is to demonstrate the rate results of the theoretical sections, and to compare the two types of basis functions spaces that we described above. We will take a partial approach and first fix N to study the role of K, and then afterwards fix K to study the role of N. All subsequent results are for the case of $\beta = 0.95$. This is to save space. We implemented the methods for other values of β and since the numerical results were qualitatively the same, we have left these out. The main difference in the numerical results is that higher values of β tend to shift the overall level of the value function upwards and add more curvature to it. This in turn generally leads to an increase in the absolute bias and variance numbers. However, in terms of percentage bias and variance, the performance of the methods was very similar across different values of β.

Effect of varying K for projection-based value function approximation

The theory for the projection-based value function approximation informs us that the choice of the basis functions will have a first-order effect on the bias while only a second-order effect on the variance. In particular, we expect $Bias\,(z)$, as defined in the beginning of this section, to satisfy $Bias\,(z) \cong \Pi_K\,(v)\,(z) - v\,(z)$, cf. discussion following Theorem 5, while $Var\,(z)$ should be much less affected by K. The actual size of the bias obviously depends on the curvature and smoothness of v_0 and the particular choice of basis functions. But we know that v is an analytic function and with only moderate curvature, cf. Fig. 1 and so expect it to be well-approximated by a small number of polynomial basis functions.

This is confirmed by the pointwise bias and standard deviation, $\sqrt{Var\,(z)}$, reported in Fig. 6: First, as can be seen on the left-hand side panel of Fig. 6, first-order B-splines lead to significantly larger point-wise bias compared to the other two sieve bases, namely second-order B-splines and Chebyshev polynomials. This is accordance with theory since we know

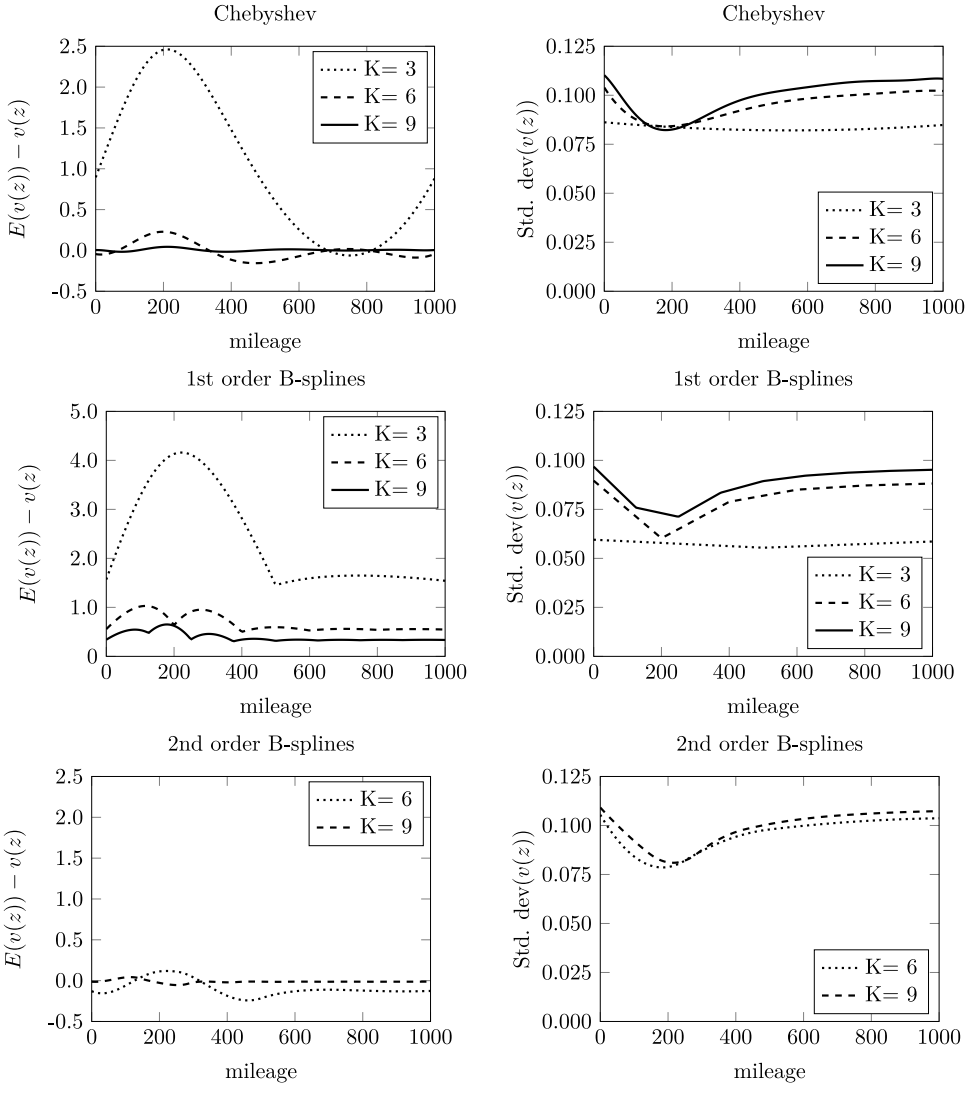

Fig. 6. Point-wise bias and standard deviation of solutions for various choices of K using different interpolation schemes, $N = 200$, $S = 200$, $\sigma_Z = 15$.

that a smooth function is better approximated by higher-order polynomials, cf. the error rates reported in as a function of s. At the same time, second-order B-splines and Chebyshev polynomials exhibit very similar biases for a given choice of K.

The right-hand side panel of Fig. 6 shows the point-wise standard deviation across different choices of K for the three different sieve bases. Consistent with the theory, the standard deviation of the value function approximation is not very sensitive to the particular choice of the sieve basis and the number of basis functions uses. That is, the sieve basis mostly affects the bias with only minor impact on the variance.

Finally, we examine how the bias behaves as we further increase K. Fig. 7 plots $\|Bias\|_\infty$ as a function of K. Similar to Fig. 6 we see much more rapid convergence when smooth basis functions are used, and with little improvement for K greater than 9. This is not surprising given the reported shape of v. The second-order B-splines and Chebyshev basis functions produce very similar fits, even if they are evaluated on different grids and the B-splines have very different properties compared to Chebyshev polynomials. Indeed, the curves are practically overlapping. This is in accordance with the asymptotic theory that predicts that higher-order B-splines and Chebyshev polynomials should lead to similar biases. Moreover, the theory informs us that if v is analytic, and this is the case in this particular implementation, we should expect the bias to vanish with rate $O(K^{-K})$ when using polynomial interpolation. The bias indeed does go to zero very quickly and so the numerical results support the theory.

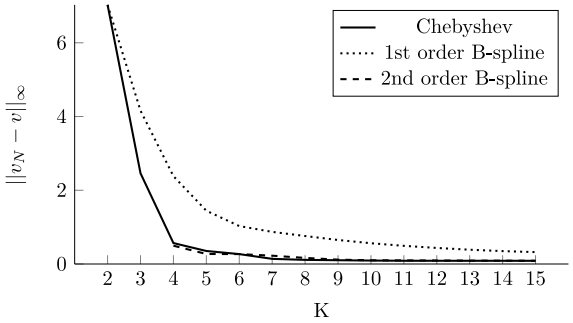

Fig. 7. Sup-norm of bias of solutions for various choices of K using various interpolation schemes, $N = 200$, $S = 200$.

Table 1
Bias, variance, and rates of convergence for various values of K.

# of basis functions, K	Sieve method 1	2	5	10	15	Self-approx. method
	σ_z=15					
$\|\|Bias\|\|_\infty$ for $N = 500$	12.743	7.029	0.348	0.016	0.003	0.203
$\sqrt{\|Var\|_\infty}$ for $N = 500$	0.000	0.020	0.063	0.066	0.066	0.450
Convergence rate for $\|\|Bias\|\|_\infty$	0.000	0.000	0.002	−0.012	−0.212	−1.290
Convergence rate for $\sqrt{\|Var\|_\infty}$	0.169	−0.500	−0.501	−0.501	−0.501	−3.618
	σ_z=100					
$\|\|Bias\|\|_\infty$ for $N = 500$	22.446	10.937	0.112	0.009	0.009	0.084
$\sqrt{\|Var\|_\infty}$ for $N = 500$	0.000	0.128	0.218	0.215	0.215	0.094
Convergence rate for $\|\|Bias\|\|_\infty$	0.000	0.000	−0.027	−0.299	−0.299	−1.331
Convergence rate for $\sqrt{\|Var\|_\infty}$	0.169	−0.500	−0.501	−0.501	−0.501	−0.543

Simulation errors, rates of convergence and asymptotic normality

We now compare the errors due to simulations and the rates with which these vanish for the two solution methods. For both methods, theory tells us that N should have a first-order effect on the variance of the approximate value function which is supposed to vanish at rate $1/N$, cf. Theorems 5 and 3. Our asymptotic theory is, on the other hand, silent about the size of simulation bias and the rate with which it should vanish with. However, we can think of both the sieve-based and self-approximating method as a nonlinear GMM-estimator where the simulated Bellman operator defines the sample moments. Importing results for GMM estimators, see, e.g., Newey and Smith (2004), we should expect the simulation bias to be of order $1/N$.

In Fig. 8 we investigate this prediction by plotting $\|Bias\|_\infty$ and $\sqrt{\|Var\|_\infty}$ for the sieve-based method (left panels) and for the self-approximating method (right panels) for two different choices of σ_ε and for across different values of N. To examine the rate with which the simulation bias and variance vanish we estimate the following exponential regressions by NLS $\sqrt{\|Var\|_\infty} = \exp(\alpha_{SD} + \rho_{SD}\ln(N))$ and $\|Bias\|_\infty = \exp(\alpha_{Bias} + \rho_{Bias}\ln(N))$ where ρ_{SD} and ρ_{Bias} measure the rate at which $\sqrt{\|Var\|_\infty}$ and $\|Bias\|_\infty$ vanish respectively. The resulting regression fit estimates are reported in both Fig. 8 as well as in Table 1. In Table 1 we present bias and standard deviation for $N = 500$ as well as their rates of convergence for both methods; with various values of K for the sieve approximation method.

According to the theory, the variance should vanish with rate $1/N$ for both methods and we therefore expect $\rho_{SD} = -0.5$ so that $\sqrt{\|Var\|_\infty}$ vanish with $1/\sqrt{N}$. For the projection based method, we see that the rate with which the standard deviation shrinks to zero is indeed close to -0.5 for all values of $K > 1$ and irrespectively of the value of σ_ε. For the self-approximating method we estimate the rate to $\rho_{SD} = -0.541$ when $\sigma_\varepsilon = 100$, which is in line with the theory. However, $\sqrt{\|Var\|_\infty}$ is found to vanish with rate $1/N^{3.6}$ for $\sigma_\varepsilon = 15$. This seems to indicate that the asymptotic theory developed in Theorems 3 and 4 does not provide a very accurate approximation of the performance of the self-approximating method for small and moderate choices of N when the support of $Z_t|(Z_{t-1} = z, d_t = 1)$ is small ($\sigma_Z = 15$). We conjecture that the discrepancy between theoretical predictions and numerical results for the self-approximating method is due to the aforementioned issues with the marginal importance sampler discussed in Section 6.2: Many of the draws are not used in the computation of the simulated Bellman operator because they fall outside the support of $Z_t|Z_{t-1} = z$ for a given choice of z. Thus, the effective number of draws is smaller than N and changes as z varies.

For the projection based method, the main source of bias is due to the sieve projection. From Fig. 6, we see that, with $N = 200$ and $K = 9$, the sieve-based methods using second-order B-splines or Chebyshev polynomials have virtually no bias, and both Fig. 8 and Table 1 also confirm that we practically eliminate by approximate the value function using Chebyshev polynomials with $K = 20$. However, there still remains a small bias that vanishes as N grows. For small K, we see that the bias is roughly independent of N. As K increases so does the dependence on N. However, even for $K = 20$

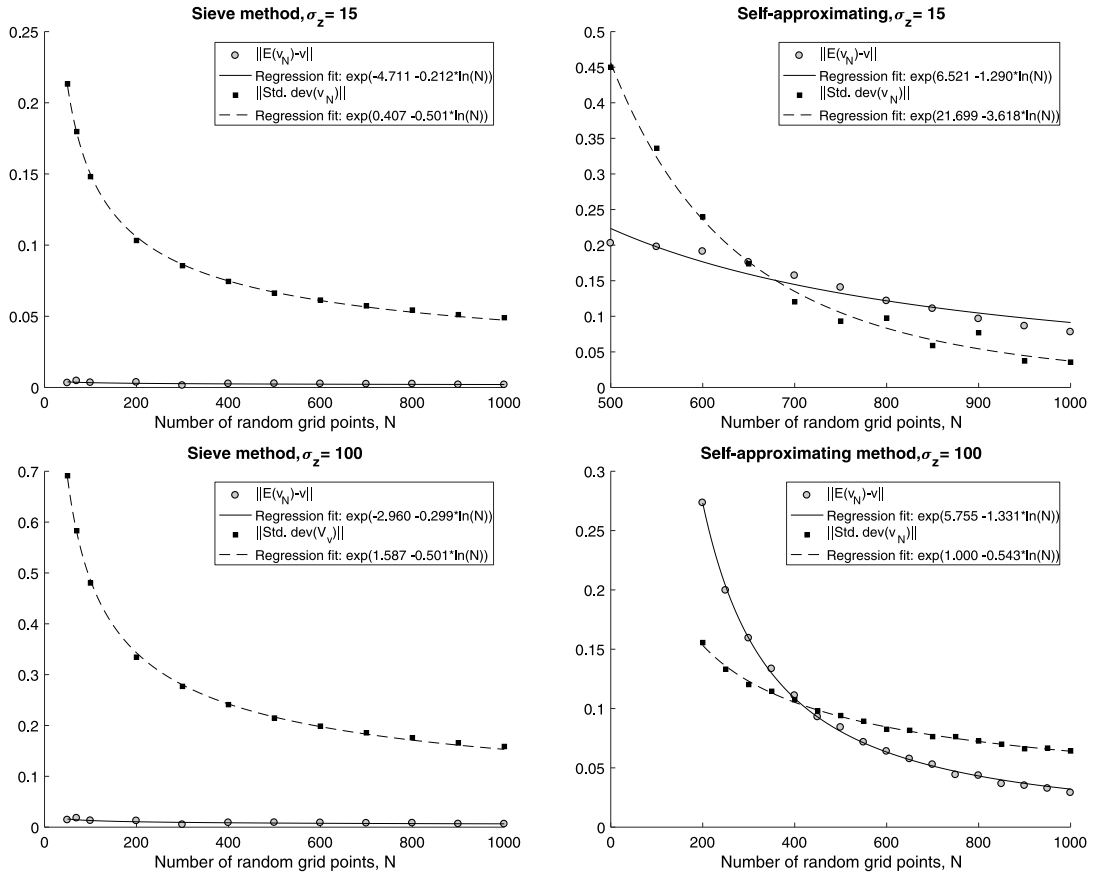

Fig. 8. Convergence results. *Notes*: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, and transition parameters are $a = 2$, $b = 5$ and $\pi = 0.000000001$. Uniform bias and variance were estimated using 500 evaluation points and $S = 2000$ implementations.

where we estimate ρ_{Bias} to be 0.21 and 0.30 for $\sigma_Z = 15$ and $\sigma_Z = 100$ respectively, the rate of convergence is far from $1/N$. This is probably due to the presence of higher-order bias components that our asymptotic theory does not account for.

For the self-approximating method, there is no sieve projection bias but a larger simulation induced bias that decreases with N. We obtain rate estimates of $1/N^{1.7}$ and $1/N^{1.4}$ for the bias when $\sigma_Z = 15$ and $\sigma_Z = 100$ respectively; these are slightly faster than expected but not too far from the theoretical predictions of $1/N$. For the self-approximating method, bias constitutes more than half of RMSE when $N < 600$ for $\sigma_Z = 15$ (or $N < 400$ for $\sigma_Z = 100$), but since $\|Bias\|_\infty$ decays faster than $\sqrt{\|Var\|_\infty}$, the simulation bias eventually becomes second order for large N.

Comparing $\|MSE\|_\infty$ for $N = 500$ we find that the sieve-based method clearly dominates the self-approximating method when $\sigma_Z = 15$, whereas the self-approximating method performs best when $\sigma_Z = 100$. This is not entirely surprising since a large value of σ_ε implies a large conditional support of Z_t in which case the draws of the marginal sampler are more likely to fall within the support, cf. the discussion in Section 6.2. Thus, the over-all error of the self-approximating method will tend to be smaller when σ_Z is large. The opposite is the case for the sieve based method which becomes more precise for smaller value of σ_Z since the variance of the simulated Bellman operator used for this method gets smaller as σ_Z gets smaller. This shows that there is considerably scope for improving the performance of the sieve-based method by more careful design of the sampling method.

Theorems 4 and 5 state that when N is large, the approximate value functions should be normally distributed. We here investigate whether this asymptotic approximation is useful in practice by looking at the pointwise distribution of the approximate solutions obtained through both methods. In Fig. 9, we plot the distribution of $(\tilde{v}(z) - E[\tilde{v}(z)]) / \sqrt{Var(\tilde{v}(z))}$ for $z = 500$, where $\tilde{v}$ denotes a given approximation method, together with the standard normal distribution. It is here important to note we do not center the estimate around $v(z)$ but instead around $E[\tilde{v}(z)]$; this is due to the sizable bias of the self-approximating method. For the sieve-based method, we see that its normalized distribution is quite close to the standard normal irrespectively of the value of σ_Z. In contrast, the normal distribution is a poor approximation for the self-approximating method when $\sigma_Z = 15$ when $N = 500$; we expect this is due to the fact that the effective number of draws is quite small and so the asymptotic approximation is poor in this case. As expected the approximation gets better as N and/or σ_Z increases.

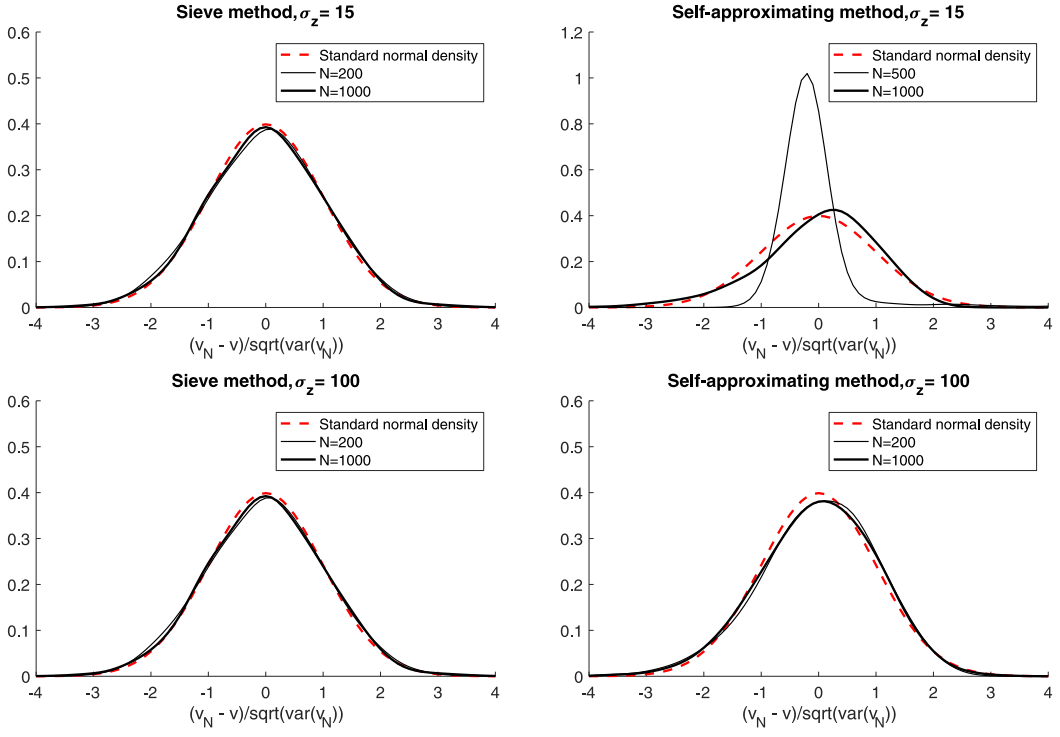

Fig. 9. Asymptotic Normality. *Notes*: Each panel shows kernel density estimates of $(\widehat{v}_N(z) - E[\widehat{v}_N(z)])/\sqrt{var(\widehat{v}_N(z))}$ for $z = 500$ based on $S = 2000$ solutions for each sample size N. Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, and transition parameters are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

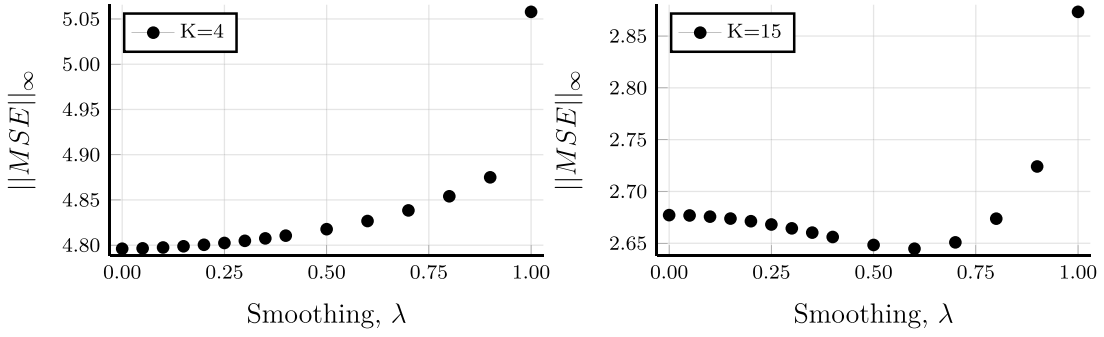

Fig. 10. Sup-norm MSE of solutions to Bellman operators with simulated taste shocks and state transitions for varying levels of smoothing, for $N = 100$.

6.5. Effect of smoothing

The results reported above did not involve any smoothing bias. We now numerically study the effect of smoothing. This is done by, instead of integrating out the i.i.d. extreme value taste shocks ε_t analytically as we have done so far, using Monte Carlo simulations to evaluate this part of the integral and then introducing the smoothing device to ensure that the simulated Bellman operator remains smooth. While this may appear somewhat artificial, the merit of doing this exercise is that we can use the same "exact" solution as benchmark as used above.

In Fig. 10 we plot the sup-norm of the mean squared error, $\|\text{MSE}\|_\infty$ as a function of λ (the smoothing scale parameter) for the sieve-based method using $K = 4$ or 8 Chebyshev polynomials (similar results were obtained for the self-approximating method and so are left out). For $K = 4$, the MSE increases monotonically as a function of λ while for $K = 15$ the bias due to smoothing is non-monotonic in λ. In both cases, at $\lambda = 0$, any remaining biases are due to either sieve-approximation or simulations. Importantly, the bias due to smoothing is negligible (relative to the other biases) for small and moderate values of λ while the variance is largely unaffected. We have no theory or heuristics for choosing

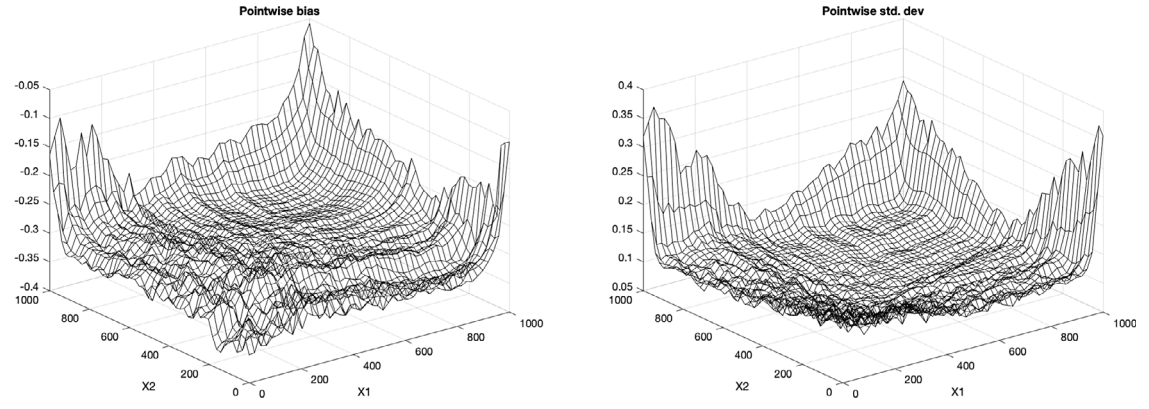

Fig. 11. Approximation errors of self-approximating method, bivariate DDP. *Notes*: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, $\lambda = 1$ and transition parameters are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$. The "exact" solution was computed by averaging over $S = 100$ solutions, each found using the smoothed random Bellman operator with $N = 3000$ pseudo random draws. Each fixed point was found using a contraction tolerance of machine precision.

an optimal λ to optimally balance bias and variance due to smoothing but the current numerical results indicate that choosing a quite small λ value works well.

6.6. Performance in the bivariate case

We now examine how the solution methods perform in the bivariate case ($d_Z = \dim(Z_t) = 2$) in order to see if there is any curse of dimensionality built into the two methods. We do this for two different models as described below.

An additive DDP

We here follow the approach of Arcidiacono et al. (2013) and Rust (1997a) and build a d_Z-dimensional model by adding up d_Z independent versions of the univariate model considered so far. That is, we choose the utilities and state dynamics as $\bar{u}(z, \varepsilon, d) = \sum_{i=1}^{d_Z} u(z_i, \varepsilon_i, d_i)$ and $\bar{F}_Z(z'|z, d) = \prod_{i=1}^{d_Z} F_Z(z_i'|z_i, d_i)$, where $z = (z_1, \ldots, z_{d_z})$ and $d = (d_1, \ldots, d_{d_Z})$, with $F_Z(z_i'|z_i, d_i)$ and $u(z_i, \varepsilon_i, d_i)$ denoting the state transition and per-period utility in the univariate case as described in Section 6.1. Note here that $(Z_{t,i}, \varepsilon_{t,i})$ and $(Z_{t,j}, \varepsilon_{t,j})$ are fully independent of each other, $i \neq j$ and the number of alternatives is 2^{d_Z}, where d_Z Thus, the model considers the joint replacement decision of d_Z assets whose stochastic usages $(Z_{t,1}, \ldots, Z_{t,d_Z})$ are mutually independent. Conveniently, the integrated value function of this multidimensional problem, $\bar{v}(z_1, \ldots, z_{d_Z})$, is simply the sum of the solutions to each of the underlying univariate models, $\bar{v}(z_1, \ldots, z_{d_Z}) = \sum_{i=1}^{d_Z} v(z_i)$, where $v(z_i)$ is the solution to the univariate model in Section 6.1. This is a rather simplistic multivariate model but it comes with the major advantage that we can obtain a very accurate approximation of the exact solution by simply adding up the "exact" solution found for the univariate case. With a more complicated multidimensional structure, the computational cost of finding the "exact" solution is much higher. However, when implementing our solution methods, we forgo the knowledge of the additive structure of the solution and so treat the above model as a "proper" multivariate problem.

Simulation error

Given the issues with the self-approximating method for small values of $\sigma_Z = 15$, we here focus exclusively on the case $\sigma_Z = 100$. To get a sense of the pointwise performance of the self-approximating method, we plot the pointwise bias and standard deviation for this method with $N = 3000$ in Fig. 11 together with the pointwise errors of the corresponding replacement (choice) probabilities. The overall shape and level of the integrated value function are quite well captured, and the same is true for the policy. However, the approximation errors tend to get larger out in the tails of the distribution and some of this comes from the fact that the issues with the marginal sampler used for the self-approximating method are amplified here. The problems are especially present in the off-grid evaluations, where we often have very few draws in a given region where we want to evaluate the value function or policies.

Next, we examine $\|Bias\|_\infty$ and $\sqrt{\|Var\|_\infty}$ for both methods as we increase N. These are plotted in Fig. 12 where it should be noted that the reported range of N reported on the x-axis of the two figures differ substantially. This is due to the fact that the self–approximating method became numerically unstable for N smaller than 1,400 while no such issues were present for the sieve-based method. As in the univariate case, both bias and variance of the two methods vanish as N increases. However, comparing Figs. 12 and 8, while the errors of the sieve-based method in the bivariate case is of a similar magnitude as in the univariate case, the errors of the self-approximating method are much larger in the bivariate case. This seems to indicate a certain type of curse-of-dimensionality in this particular application of the self-approximating method. This is caused by the issues with the marginal importance sampler employed for this method.

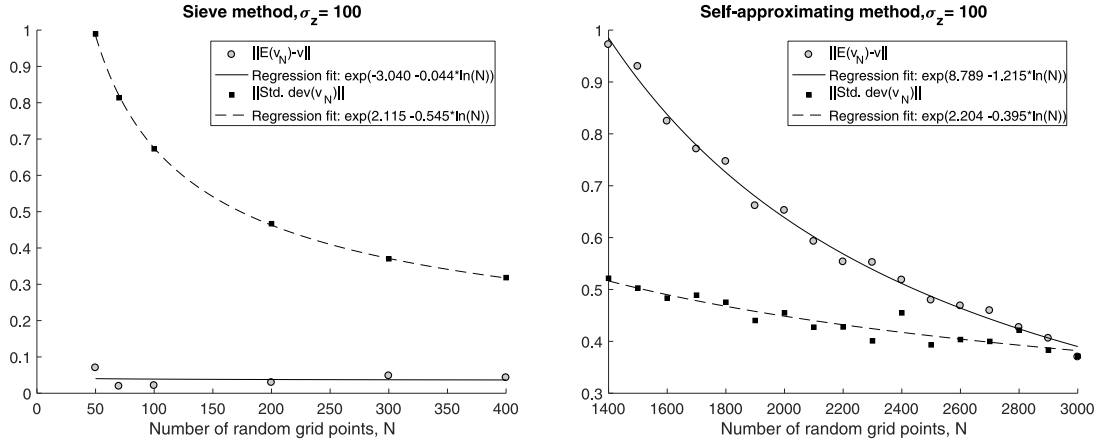

Fig. 12. Simulation errors for bivariate additive DDP. *Notes*: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, $\lambda = 1$ and parameters for transition density $f(z'|z, d)$ are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$. Point-wise bias and variance was estimated in 500 evaluation points based on $S = 200$ replications. We report the sup norm of the bias and the standard deviation for each N for both methods and NLS regression fits of $\sqrt{\|Var\|_\infty} = \exp(\alpha_{SD} + \rho_{SD} \ln(N))$ and $\|Bias\|_\infty = \exp(\alpha_{Bias} + \rho_{Bias} \ln(N))$.

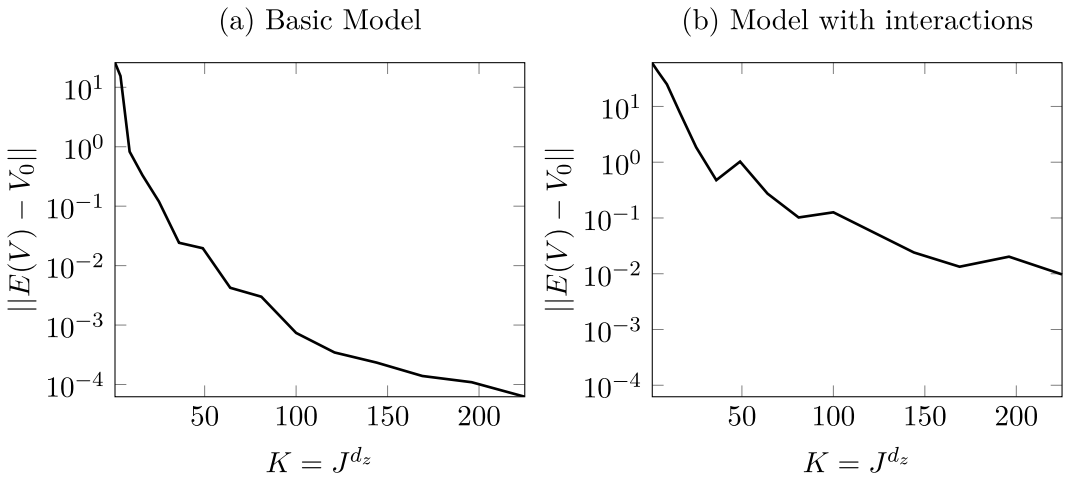

Fig. 13. Bias of value function in bivariate DDP for varying K.

Sieve approximation error

In the implementation of the sieve-based method we use as sieve basis the tensor product of univariate Chebyshev polynomials or B-splines. That is, given, say, J univariate basis functions, say, $p_1, \ldots, p_J$, we construct our bivariate basis functions as $B_{i,j}(z_1, z_2) = p_i(z_1)\, p_j(z_2)$ for $i, j = 1, \ldots, J$ yielding a total of $K = J^2$ bivariate basis functions. In particular, we do not exploit the additive structure of the problem since we are interested in the practical contents of Theorems 5 where no particular sparsity/special structure of the model is assumed to be known.

However, in practice, Chebyshev polynomials very easily pick up the additive structure and effectively sets the coefficients of the cross-product terms to zero. This is illustrated in Table 2 in Appendix C, where we report the coefficients for one particular projection-based bivariate value function estimate using a tensor product of $J = 5$ Chebyshev polynomials. However, this is due to the particular properties of the Chebyshev polynomials and is not enforced by us in the implementation. For example, if we instead use B-splines, the "estimated" coefficients of the cross-product terms were significantly different from zero, cf. Table 3 in Appendix C.

On the left-hand side panel (a) of Fig. 13, we report the uniform bias of the projection-based method with N chosen very large for the additive bivariate model. We find that the bias vanishes as K increases as in the one-dimensional model. However, convergence is now slower in K relative to the univariate case and we require $K = 50$ to obtain a sieve approximation bias of 10^{-2} while $K = 7$ sufficed in the univariate case. This is consistent with theoretical error rates for polynomial interpolation where the rate slows down as the dimension of the problem increases, cf. Section 4.3.

A non-additive DDP

One concern with the numerical results reported for the bivariate additive model in the previous section is that they may understate the curse of dimensionality of the sieve method: The true value function is by construction additive in the two state variables and so interaction terms do not appear. This in turn implies that the computational complexity of solving this particular model is relatively low; in particular, the solution should be well-approximated by lower-dimensional sieves (K small).

To investigate how the sieve method performs when applied to a more complex, non-additive model, we here consider a slightly more complicated bivariate model where we include a multiplicative interaction term so that maintenance and replacement costs of the two busses interact, $\bar{u}(z,d) = \sum_{i=1}^{2} u(z_i,d_i) - u(z_1,d_1)u(z_2,d_2)/20$. Such a structure could, for example, reflect that capacity constraints make it more costly to simultaneously replace the engines of both busses. The resulting value function will have a more complicated multidimensional structure and so we expect that the computational cost of our sieve method should be higher in this scenario.

The sieve approximation bias of our solution method for this model is reported on the right-hand side panel (b) in Fig. 13. Compared to panel (a) – the additive case – we see that more sieve terms are required in order to reach a specific absolute error level in the model with interactions. In Table 4 in Appendix C the coefficients of the first ten basis functions in each dimension and their interactions are reported. Compared to the Chebyshev-based solution earlier we see quite significant coefficients in the coefficients for the cross-terms. However, the coefficients of the basis functions tend to zero quite quickly as K increase. The sup-norm of the difference in the value function at 40.000 evaluation grids is on the order of 10^{-5} when comparing the solutions with $K = 50^2 = 2500$ and $K = 30^2 = 900$ basis functions, and individual coefficients fall below 10^{-6} for univariate basis functions and cross products beyond the 22nd univariate basis functions, and below 10^{-8} around the 30th basis functions. However, it is important to stress that a large K here only comes with a computational cost while a large K has little effect on the variance of the sieve method. All together, we find that the sieve-based solution method works well also in higher dimensions, in particular when the model has a particular structure that can be utilized in the solution method.

7. Conclusion

We have proposed two novel methods for numerical computation of either the so-called integrated or expected value function in a general class of dynamic discrete choice models. Both methods rely on a smoothed simulated version of the Bellman operators defining the integrated and expected values functions. The smoothing facilitates both the practical implementation and the theoretical analysis of the approximate value functions. Under regularity conditions, we develop an asymptotic theory for the two methods as the number of simulations used to compute the simulated Bellman operators diverges. A set of numerical experiments shows that our first method, the so-called self-approximating method can be somewhat unstable while the second one, which relies on sieve methods, appears much more numerically robust. The next step is to develop methods for choosing the number of simulations, sieve basis functions and smoothing parameter λ in a given setting so that the resulting approximate solution is of a good quality. Another area of research is to investigate how the proposed solution methods can be used for the estimation of dynamic discrete choice models.

Appendix A. Auxiliary results

We derive two general results for approximate solutions to functional fixed-points. Let $(\mathcal{X}, \|\cdot\|)$ be a normed vector space and $\Psi : \mathcal{X} \to \mathcal{X}$ be some contraction mapping w.r.t. $\|\cdot\|$ so that there exists a unique solution $x_0 \in \mathcal{X}$ to $x = \Psi(x)$. Let Ψ_N be an approximation to Ψ and let Π_K be a projection operator, $K, N \geq 1$.

Theorem A.1. *Suppose (i) $\|\Psi_N(x_0) - \Psi(x_0)\| = O_p(\rho_{\Psi,N})$ for some $\rho_{\Psi,N} \to 0$ and (ii) for some $\beta < 1$, $\|\Psi_N(x) - \Psi_N(y)\| \leq \beta\|x - y\|$ for all N large enough and all x, y. Then there exists a unique solution $x_N \in \mathcal{X}$ to $x = \Psi_N(x)$ with probability approaching one (w.p.a.1) satisfying $\|x_N - x_0\| = O_P\left(\rho_{\Psi,N}\right)$.*

Suppose furthermore (iii) $\Pi_K : \mathcal{X} \to \mathcal{X}$ satisfies $\|\Pi_K(x_N) - x_N\| = O_p(\rho_{\Pi,K})$ for some $\rho_{\Pi,K} \to 0$. Then there exists a unique solution $\hat{x}_N \in \mathcal{X}$ to $x = (\Pi_K \Psi_N)(x)$ w.p.a.1 satisfying

$$\left\|\hat{x}_N - x_0\right\| \leq \left\|\hat{x}_N - x_N\right\| + \|x_N - x_0\| = O_p(\rho_{\Pi,K}) + O_p(\rho_{\Psi,N}).$$

Proof. We first observe that due to (ii), there exists a unique solution $x_N = \Psi_N(x_N)$ for all N large enough which satisfies

$$\begin{aligned}\|x_N - x_0\| &= \|\Psi_N(x_N) - \Psi(x_0)\| \leq \|\Psi_N(x_N) - \Psi_N(x_0)\| + \|\Psi_N(x_0) - \Psi(x_0)\| \\ &\leq \beta\|x_N - x_0\| + \|\Psi_N(x_0) - \Psi(x_0)\|,\end{aligned}$$

and so $\|x_N - x_0\| \leq \|\Psi_N(x_0) - \Psi(x_0)\|/(1-\beta) = O_P\left(\rho_{\Psi,N}\right)$. Next, combining (ii) and (iii), we see that $\Pi_K \Psi_N$ is a contraction mapping w.p.a.1. with Lipschitz coefficient β, and so $\hat{x}_N$ defined in the theorem exists and is unique w.p.a.1. Moreover, by the same arguments employed in the analysis of x_N,

$$\|\hat{x}_N - x_N\| \leq \frac{\|\Pi_K \Psi_N(x_N) - \Psi_N(x_N)\|}{1-\beta} = \frac{\|\Pi_K(x_N) - x_N\|}{1-\beta} = O_p(\rho_{\Pi,K}). \quad \square$$

Theorem A.2. *Suppose the following conditions are satisfied: (i)* $\|x_N - x_0\| = O_P\left(\rho_{\psi,N}\right)$*; (ii)* $\rho_{\psi,N}^{-1}\{\Psi_N(x_0) - \Psi(x_0)\} \rightsquigarrow \mathbb{G}$ *in* $(\mathcal{X}, \|\cdot\|)$*; (iii)* $\Psi_N(x_0)$ *is Frechet differentiable at* x_0 *w.p.a.1 with Frechet differential* $\nabla\Psi_N(x_0)[\cdot] : \partial\mathcal{X} \mapsto \mathcal{X}$ *for some function set* $\partial\mathcal{X}$*, where* $x_N - x_0 \in \partial\mathcal{X}$ *w.p.a.1, such that* $\|\Psi_N(x_N) - \Psi_N(x_0) - \nabla\Psi_N(x_0)[x_N - x_0]\| = o_P(\|x_N - x_0\|)$ *and (iv)* $\sup_{dx\in\partial\mathcal{X}:\|dx\|=1} \|\{\nabla\Psi_N(x_0) - \nabla\Psi(x_0)\}[dx]\| = o_p(1)$*. Then* $\{I - \nabla\Psi(x_0)\}\left[\rho_{\psi,N}\{x_N - x_0\}\right] \rightsquigarrow \mathbb{G}$*. If furthermore (v)* $I - \nabla\Psi(x_0)[\cdot] : \partial\mathcal{X} \mapsto \mathcal{X}$ *has a continuous inverse, then* $\rho_{\psi,N}\{x_N - x_0\} \rightsquigarrow \{I - \nabla\Psi(x_0)\}^{-1}[\mathbb{G}]$*.*

Proof. To show the first claim, combine a functional Taylor expansion with conditions (i) and (iii),

$$\begin{aligned} 0 &= (I - \Psi_N)(x_N) = (I - \Psi_N)(x_0) + \{I - \nabla\Psi_N(x_0)\}[x_N - x_0] + o_P(\|x_N - x_0\|) \\ &= \Psi(x_0) - \Psi_N(x_0) + \{I - \nabla\Psi_N(x_0)\}[x_N - x_0] + o_P\left(\rho_{\psi,N}\right). \end{aligned}$$

Next, by (iv),

$$\begin{aligned} &\|\{I - \nabla\Psi_N(x_0)\}[x_N - x_0] - \{I - \nabla\Psi(x_0)\}[x_N - x_0]\| \\ &= \|\{\nabla\Psi_N(x_0) - \nabla\Psi(x_0)\}[x_N - x_0]\| \\ &\leq \sup_{dx\in\partial\mathcal{X}:\|dx\|=1} \|\{\nabla\Psi_N(x_0) - \nabla\Psi(x_0)\}[dx]\| \, \|x_N - x_0\| \\ &= o_P\left(\rho_{\psi,N}\right). \end{aligned}$$

Combining this with (ii),

$$\{I - \nabla\Psi(x_0)\}\left[\rho_{\psi,N}^{-1}\{x_N - x_0\}\right] = \rho_{\psi,N}^{-1}\{\Psi_N(x_0) - \Psi(x_0)\} + o_P(1) \rightsquigarrow \mathbb{G},$$

The second claim follows by (v) and the continuous mapping theorem. □

It is important here to note the tension between the requirement that $x_N - x_0 \in \partial\mathcal{X}$ in (iii) and $\sup_{dx\in\partial\mathcal{X}:\|dx\|=1} \|\{\nabla\Psi_N(x_0) - \nabla\Psi(x_0)\}[dx]\| = o_p(1)$. The first condition will hold if we choose $\partial\mathcal{X}$ large enough. But at the same time, we need to show uniform convergence over the same space which will generally only hold if $\partial\mathcal{X}$ is Glivenko–Cantelli. In the application to value function approximation, this is achieved by choosing $\partial\mathcal{X} = \mathbb{C}_r^1(\mathcal{Z})$ defined in Section 4.3.3 for some $r < \infty$.

Appendix B. Proofs

Proof of Theorem 1. First note that for any $V(z;\lambda) \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ and with C denoting a generic constant,

$$\begin{aligned} \left|G_\lambda\left(u_\psi(U,z) + \beta V_\psi(U;z,\lambda)\right)\right| &\leq C\left(1 + \left\|u_\psi(U,z)\right\| + \beta\left\|V_\psi(U;z,\lambda)\right\|\right) \\ &\leq C\left(1 + \bar{u}_\psi(U) + \beta\|V\|_\infty\right), \end{aligned} \tag{B.1}$$

and so, using Assumption 1,

$$\begin{aligned} \|\Gamma(V)\|_\infty &\leq \sup_{(z,\lambda)\in\mathcal{Z}\times\left(0,\bar{\lambda}\right)} E\left[\left|G_\lambda\left(u_\psi(U;z) + \beta V_\psi(U;z,\lambda)\right)\right| w_\psi(U;z)\right] \\ &\leq CE\left[\left|\left(1 + \bar{u}_\psi(U) + \beta\|V\|_\infty\right)\right| \bar{w}_\psi(U)\right] < \infty, \end{aligned}$$

which shows that $\Gamma : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. Recycling Eq. (B.1),

$$\begin{aligned} \|\Gamma_N(V)\|_\infty &\leq \frac{\sum_{i=1}^N \left\|G_\lambda\left(u_\psi(U_i;z) + \beta V_\psi(U_i;z,\lambda)\right)\right\| w_\psi(U_i;z)}{\sum_{i=1}^N w_\psi(U_i;z)} \\ &\leq C\left(\frac{\sum_{i=1}^N \left\|\bar{u}_\psi(U)\right\| w_\psi(U_i;z)}{\sum_{i=1}^N w_\psi(U_i;z)} + 1 + \beta\|V\|_\infty\right) \\ &\leq C\left(\frac{\sum_{i=1}^N \left\|\bar{u}_\psi(U_i)\right\| \bar{w}_\psi(U_i)}{\inf_{z\in\mathcal{Z}} \sum_{i=1}^N w_\psi(U_i;z)} + 1 + \beta\|V\|_\infty\right) < \infty. \end{aligned}$$

Thus, for any given $N \geq 1$, $\Gamma_N : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. To show that $\Gamma_N : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ is a contraction, use that, by quasi-linearity of $G_\lambda(r)$, for any $V_1, V_2 \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$,

$$\begin{aligned} \Gamma_N(V_1)(z,\lambda,d) &= \sum_{i=1}^N G_\lambda\left(u_\psi(U_i;z) + \beta V_{\psi,2}(U_i;z,\lambda) + \beta\left[V_{\psi,1}(U_i;z,\lambda) - V_{\psi,2}(U_i;z)\right]\right) w_{N,i}(z,d) \\ &\leq \sum_{i=1}^N G_\lambda\left(u_\psi(U_i;z) + \beta V_{\psi,2}(U_i;z,\lambda) + \beta\|V_1 - V_2\|_\infty \mathbf{1}_D\right) w_{N,i}(z,d) \end{aligned}$$

$$= \sum_{i=1}^{N} G_\lambda\left(u_\psi\left(U_i; z\right) + \beta V_{\psi,2}\left(U_i; z, \lambda\right)\right) w_{N,i}\left(z, d\right) + \beta \left\|V_1 - V_2\right\|_\infty \sum_{i=1}^{N} w_{N,i}\left(z, d\right)$$
$$= \Gamma_N(V_2)(z, \lambda, d) + \beta \left\|V_1 - V_2\right\|_\infty,$$

where $\mathbf{1}_d = (1, \ldots, 1) \in \mathbb{R}^D$ and we have used that $\sum_{i=1}^{N} w_{N,i}\left(z, d\right) = 1$ by construction. The proof of Γ being a contraction is analogous.

Next, we prove that $V_N\left(z, \lambda\right)$ is $s \geq 1$ times continuously differentiable under Assumption 2: We know that Γ_N is a contraction mapping on $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. But the set of $s \geq 0$ continuously differentiable functions $\mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ is a closed subset of $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ and so the result will follow if $\Gamma_{N,\lambda}\left(\mathbb{C}_s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D\right) \subseteq \mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. But for any $V \in \mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$, it follows straightforwardly by the chain rule in conjunction with the stated assumptions that $\Gamma_N(V)(z, \lambda) = \sum_{i=1}^{N} G_\lambda\left(u_\psi\left(U_i; z\right) + \beta V_\psi\left(U_i; z, \lambda\right)\right) w_{N,i}\left(z\right)$ is $s \geq 0$ continuously differentiable w.r.t. (z, λ). The proof of the Lipschitz property under Assumption 2(i) is similar and so left out. □

Proof of Theorem 2. We only show the result for V_0; the proof for the V_N is analogous. Applying (3.8), the following holds for any V,

$$\begin{aligned}\left|\Gamma(V)(z, 0, d) - \Gamma(V)(z, \lambda, d)\right| \leq & \int \left|\max_{d \in \mathcal{D}}\left\{u(s', d) + \beta V(z', d')\right\} - G_\lambda\left(u(s') + \beta V(z')\right)\right| dF_S(s'|z, d) \\ \leq & \sup_{r \in \mathbb{R}^D}\left|G_\lambda\left(r\right) - \max_{d \in \mathcal{D}} r\left(d\right)\right| \int_{\mathcal{Z} \times \mathcal{E}} dF_s(ds'|z, d) \\ \leq & \lambda \log D.\end{aligned}$$

The result now follows from the first part of Theorem A.1 with $\Psi_N\left(\cdot\right) = \Gamma\left(\cdot\right)\left(\cdot, \lambda_N\right)$. □

Our asymptotic analysis of V_N proceeds in two steps: First, we develop a master theorem that delivers the desired result under a set of high-level conditions on the model and chosen importance sampler. The conditions are formulated to cover a wide range of different specifications, including both the case of Z_t being continuously distributed or having countable support. Also, the master theorem allows for a wide range of the per-period utility functions and importance samplers. To state the high-level conditions, we recall the following definitions (see van der Vaart and Wellner, 1996): A class $\mathcal{F}$ of measurable functions mapping U into $\mathbb{R}$ is called P_U-Glivenko–Cantelli if $\sup_{f \in \mathcal{F}}\left|\frac{1}{N}\sum_{i=1}^{N} f\left(U_i\right) - E\left[f\left(U\right)\right]\right| \to^P 0$ and it is called P_U-Donsker if $\sup_{f \in \mathcal{F}} \frac{1}{\sqrt{N}} \sum_{i=1}^{N}\left\{f\left(U_i\right) - E\left[f\left(U\right)\right]\right\} \rightsquigarrow \mathbb{G}$ in the space of all bounded functions from $\mathcal{F}$ to $\mathbb{R}$, where $\mathbb{G}$ is a tight Gaussian process.

Theorem B.1. *(i) Suppose that Assumption 1 is satisfied and the function classes* $\mathcal{W} := \left\{U \mapsto w_\psi\left(U; z\right) \middle| z \in \mathcal{Z}\right\}$ *and*

$$\mathcal{G} = \left\{U \mapsto G_\lambda\left(u_\psi\left(U; z\right) + \beta V_0\left(\psi_Z\left(U; z\right), \lambda\right)\right) w_\psi\left(U; z\right) \middle| \left(z, \lambda\right) \in \mathcal{Z} \times \left(0, \bar{\lambda}\right)\right\} \tag{B.2}$$

are P_U-Donsker. Then the first part of Theorem 3 holds.

(ii) Suppose furthermore that $V_N - V_0 \in \partial\mathcal{V}$ where $\partial\mathcal{V}$ is P_U-Glivenko–Cantelli with an integrable envelope function and

$$\mathcal{G}' = \left\{U \mapsto \sum_{d \in \mathcal{D}} \dot{G}_{d,\lambda}\left(u_\psi\left(U; z\right) + \beta V_{\psi,0}\left(U; z, \lambda\right)\right) w_\psi\left(U; z\right) \middle| \left(z, \lambda\right) \in \mathcal{Z} \times \left(0, \bar{\lambda}\right)\right\}$$

is P_U-Glivenko–Cantelli. Then the conclusions of Theorem 4 also hold.

Proof. To show (i), we apply the first part of Theorem A.1 with $\Psi_N = \Gamma_N$, which is a contraction w.p.a.1, cf. Theorem 1. First write

$$\Gamma_N(V)(z, \lambda) = \frac{\tilde{\Gamma}_N(V)(z, \lambda)}{W_N\left(z\right)}. \tag{B.3}$$

where

$$\tilde{\Gamma}_N(V)(z, \lambda) = \frac{1}{N}\sum_{i=1}^{N} G_\lambda\left(u_\psi\left(U_i; z\right) + \beta V_\psi(U_i; z)\right) w_\psi\left(U_i; z\right), \tag{B.4}$$

$$W_N\left(z\right) = \frac{1}{N}\sum_{j=1}^{N} w_\psi\left(U_i; z\right), \tag{B.5}$$

The Donsker condition on $\mathcal{G}$ and $\mathcal{W}$ now implies that

$$\sqrt{N}\left(\tilde{\Gamma}_N(V_0) - \Gamma(V_0), W_N - 1\right) \rightsquigarrow \left(\mathbb{G}_1, \mathbb{G}_2\right) \tag{B.6}$$

on $\mathcal{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\right)$, where $(\mathbb{G}_1,\mathbb{G}_2)$ is a Gaussian process, and so

$$\begin{aligned}\sqrt{N}\left\{\Gamma_N(V_0)-\Gamma(V_0)\right\} &= \sqrt{N}\left\{\tilde{\Gamma}_N(V_0)-\Gamma(V_0)\right\}-\Gamma(V_0)\sqrt{N}\left\{W_N-1\right\}+o_P(1)\\ &\rightsquigarrow \mathbb{G}:=\mathbb{G}_1-\Gamma(V_0)\mathbb{G}_2.\end{aligned}\tag{B.7}$$

In particular, $\left\|\Gamma_N(V_0)-\Gamma(V_0)\right\|_\infty = O_P\left(1/\sqrt{N}\right)$. We conclude from Theorem A.1 that $\|V_N-V_0\|_\infty = O_p(1/\sqrt{N})$.

To show the second part, we apply Theorem A.2. Weak convergence was derived above and it is easily seen that the influence function of $\Gamma_N(V_0)$ takes the form given in Eq. (5.7) and so the Gaussian process $\mathbb{G}(z,\lambda)$ in Eq. (B.7) has covariance kernel given in (5.6). The Frechet differential $dV\mapsto\nabla\Gamma_N(V_N)[dV]$ was derived in (4.14). It is a linear operator with $\|\nabla\Gamma_N(V_N)[dV]\|\leq\beta\|dV\|$ and so $dV\mapsto\{I-\nabla\Gamma_N(V_N)\}[dV]$ has a well-defined continuous inverse. Thus, what remains is to verify (iv) of Theorem A.2. This is done by showing uniform convergence of $dV\mapsto\nabla\tilde{\Gamma}_N(V_0)[dV]$ over $\mathcal{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\times\partial\mathcal{V}\right)$. But

$$\nabla\tilde{\Gamma}_N(V_0)[dV](z)=\frac{\beta}{N}\sum_{i=1}^{N}\sum_{d\in\mathcal{D}}\dot{G}_{d,\lambda}\left(u_\psi(U_i;z)+\beta V_{\psi,0}(U_i;z,\lambda)\right)dV_\psi(U_i;z,\lambda,d)\,w_\psi(U_i;z)\tag{B.8}$$

where $V_\psi(U;z,\lambda,d)\in\partial\mathcal{V}_\psi$ with

$$\partial\mathcal{V}_\psi=\left\{U\mapsto dV(\psi_Z(U;z),\lambda)\,|\,(z,\lambda,dV)\in\mathcal{Z}\times\left(0,\bar{\lambda}\right)\times\partial\mathcal{V}\right\}$$

which is Glivenko–Cantelli since $\partial\mathcal{V}$ and $\{U\mapsto\psi_Z(U;z)\,|\,z\in\mathcal{Z}\}$ both have this property. Since $\mathcal{G}'$ is also Glivenko–Cantelli, it now follows from Theorem 3 in van der Vaart and Wellner (2000) that $\mathcal{G}'\cdot\partial\mathcal{V}_\psi$ is Glivenko–Cantelli as well which yields the desired result. □

Proof of Theorem 3. To show $\|V_N-V_0\|_\infty=O_P\left(1/\sqrt{N}\right)$, we verify the conditions of part (i) in Theorem B.1. First observe that $V_0(z,\lambda)$ is Lipschitz in (z,λ), cf. Theorem 1, and that $r\mapsto G_\lambda(r)$ is also Lipschitz uniformly in $\lambda\in(0,\bar{\lambda})$. Next, we show that $G_\lambda(r)$ is also Lipschitz w.r.t. λ uniformly in r by verifying that $\partial G_\lambda(r)/(\partial\lambda)$ is bounded uniformly in $\lambda\in(0,\bar{\lambda})$: Write

$$G_\lambda(r)=\lambda\log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{r(d)}{\lambda}\right)\right]=\max_{d\in D}r(d)+\lambda\log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\right],$$

where $\bar{r}(d)=r(d)-\max_{d\in\mathcal{D}}r(d)\leq 0$, $d\in\mathcal{D}$, to obtain

$$\dot{G}_\lambda^{(\lambda)}(r)=\frac{\partial G_\lambda(r)}{\partial\lambda}=\log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\right]-\frac{\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\frac{\bar{r}(d)}{\lambda}}{\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)}.\tag{B.9}$$

Since $1\leq\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\leq D$ and $-De^{-1}\leq\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\frac{\bar{r}(d)}{\lambda}\leq 0$ for all $\lambda>0$ and all $r\in\mathbb{R}^D$, we conclude that $\left|\dot{G}_\lambda^{(\lambda)}(r)\right|\leq\log(D)+De^{-1}$ and so. Next,

$$\begin{aligned}&\left|G_\lambda\left(u_\psi(U;z)+\beta V_0(\psi_Z(U,z),\lambda)\right)-G_{\lambda'}\left(u_\psi(U;z')+\beta V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\right)\right|\\ \leq\;&\left|G_\lambda\left(u_\psi(U;z)+\beta V_0(\psi_Z(U,z),\lambda)\right)-G_{\lambda'}\left(u_\psi(U;z)+\beta V_0(\psi_Z(U,z),\lambda)\right)\right|\\ &+\left|G_{\lambda'}\left(u_\psi(U;z)+\beta V_0(\psi_Z(U,z),\lambda)\right)-G_{\lambda'}\left(u_\psi(U;z')+\beta V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\right)\right|\\ \leq\;&C\left\{\left|\lambda-\lambda'\right|+\left\|u_\psi(U;z)-u_\psi(U;z')\right\|+\left\|V_0(\psi_Z(U,z),\lambda)-V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\right\|\right\},\end{aligned}$$

and it now follows that under Assumption 2 together with the Lipschitz property of V_0 that $\mathcal{G}$ as defined in Eq. (B.2) is Type IV class under P_U with index 2 according to the definition on p. 2278 in Andrews (1994) which yields the first part of the theorem.

Next, we analyze $\partial V_N/\left(\partial z_j\right)$, $j=1,\ldots,d_Z$. Since $\sup_{z\in\mathcal{Z}}\left|\sum_{i=1}^N w(S_i(z,d)\,|z,d)/N-1\right|=O_P\left(1/\sqrt{N}\right)$, we replace $w_{N,i}(z,d)$ by $w_\psi(U_i;z,d)/N$ in the following. Now, taking derivatives w.r.t. z_j, $j=1,\ldots,d_Z$, on both sides of Eq. (4.1),

$$\frac{\partial V_N(z,\lambda)}{\partial z_j}=\nabla\Gamma_N(V_N)\left[\frac{\partial V_N}{\partial z_j}\right](z,\lambda)+\Gamma_{N,j}^{(z)}(V_N)(z,\lambda),\tag{B.10}$$

where $\nabla \Gamma_N$ was defined in (4.14) and

$$\dot{\Gamma}^{(z)}_{N,j}(V_N)(z,\lambda,d) = \frac{1}{N}\sum_{i=1}^{N}\sum_{d\in\mathcal{D}} \dot{G}^{(r)}_{\lambda,d}\left(u_\psi(U_i;z,d)+\beta V_{\psi,N}(U_i;z,\lambda)\right)\frac{\partial u_\psi(U_i;z,d)}{\partial z_j} w_\psi(U_i;z) + \frac{1}{N}\sum_{i=1}^{N} G_\lambda\left(u_\psi(U_i,d)+\beta V_{\psi,N}(U_i;z,\lambda)\right)\frac{\partial w_\psi(U_i;z)}{\partial z_j}.$$

where $\dot{G}^{(r)}_{\lambda,d}(r)$ was defined in (4.12). Similarly,

$$\frac{\partial V_N(z,\lambda)}{\partial \lambda} = \nabla\Gamma_N(V_N)\left[\frac{\partial V_N}{\partial \lambda}\right](z,\lambda) + \dot{\Gamma}^{(\lambda)}_{N,j}(V_N)(z,\lambda), \tag{B.11}$$

where

$$\dot{\Gamma}^{(\lambda)}_{N,j}(V_N)(z,\lambda,d) = \frac{1}{N}\sum_{i=1}^{N}\dot{G}^{(\lambda)}_{\lambda}\left(u_\psi(U_i;z,d)+\beta V_{\psi,N}(U_i;z,\lambda)\right)w_\psi(U_i;z),$$

and

$$\dot{G}^{(\lambda)}_{\lambda}(r) = \log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{r(d)}{\lambda}\right)\right] - \frac{\sum_{d\in\mathcal{D}}\exp\left(\frac{r(d)}{\lambda}\right)\frac{r(d)}{\lambda}}{\sum_{d\in\mathcal{D}}\exp\left(\frac{r(d)}{\lambda}\right)} \tag{B.12}$$

The mapping $dV \mapsto \nabla\Gamma_N(V_N)[dV]$ is a bounded linear operator with $\|\nabla\Gamma_N(V_N)[dV]\| \le \beta\|dV\|$ and so

$$\frac{\partial V_N(z,\lambda)}{\partial z_j} = \{I-\nabla\Gamma_N(V_N)\}^{-1}\left[\dot{\Gamma}^{(z)}_{N,j}(V_N)\right](z,\lambda).$$

Thus,

$$\left\|\frac{\partial V_N}{\partial z_j}\right\|_\infty = \left\|\{I-\nabla\Gamma_N(V_N)\}^{-1}\left[\Gamma^{(z)}_{N,j}(V_N)\right]\right\|_\infty \le \frac{\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty}{1-\beta},$$

where,

$$\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty \le \frac{1}{N}\sum_{i=1}^{N}\left\|\frac{\partial u_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty \left\|w_\psi(U_i;\cdot)\right\|_\infty + \frac{1}{N}\sum_{i=1}^{N}\left\{\left\|u_\psi(U_i;\cdot)\right\|_\infty+\beta\|V_N\|_\infty\right\}\left\|\frac{\partial w_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty.$$

We know $\|V_N\|_\infty \to^P \|V\|_\infty$ and, under Assumption 2, we can appeal to the ULLN to obtain

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\frac{\partial u_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty\left\|w_\psi(U_i;\cdot)\right\|_\infty \to^P E\left[\left\|\frac{\partial u_\psi(U;\cdot)}{\partial z_j}\right\|_\infty\left\|w_\psi(U;\cdot)\right\|_\infty\right],$$
$$\frac{1}{N}\sum_{i=1}^{N}\left\|u_\psi(U_i;\cdot)\right\|_\infty\left\|\frac{\partial w_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty \to^P E\left[\left\|u_\psi(U;\cdot)\right\|_\infty\left\|\frac{\partial w_\psi(U;\cdot)}{\partial z_j}\right\|_\infty\right].$$

We conclude that $\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty$ and therefore also $\left\|\partial V_N/(\partial z_j)\right\|_\infty$ are bounded w.p.a.1. Similarly, it follows that $\|\partial V_N/(\partial\lambda)\|_\infty$ is bounded w.p.a.1. and so $V_N \in \mathbb{C}^1_r\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\right)$ w.p.a.1 for some fixed $r<\infty$.

Finally, observe that $dV \mapsto \Psi_{N,j}(dV)(z,\lambda) = \nabla\Gamma_N(V_N)[dV](z,\lambda) + \Gamma^{(z)}_{N,j}(V_N)(z,\lambda)$ is a contraction mapping on $\mathbb{B}\left(\mathcal{Z}\times\left[0,\bar{\lambda}\right]\right)^D$ and so we can apply Theorem A.1. First, we expand each of the two terms w.r.t. V_N,

$$\begin{aligned}&\nabla\Gamma_N(V_N)[dV](z;\lambda)-\nabla\Gamma_N(V_0)[dV](z;\lambda)\\&=\beta\sum_{i=1}^{N}\sum_{d_1,d_2\in\mathcal{D}}\ddot{G}^{(r)}_{\lambda,d_1,d_2}\left(u_\psi(U_i;z)+\beta\bar{V}_{\psi,N}(U_i;z,\lambda)\right)\left\{V_{\psi,N}(U_i;z,\lambda,d_2)-V_{\psi,0}(U_i;z,\lambda,d)\right\}\\&\times dV(U_i;z,\lambda,d_1)\,w_\psi(U_i;z),\end{aligned}$$

where $\ddot{G}^{(r)}_{\lambda,d_1,d_2}(r)=\frac{\partial^2 G_\lambda(r)}{\partial r(d_1)\partial r(d_1)}$. It is easily checked that $\left|\ddot{G}^{(r)}_{\lambda,d_1,d_2}(r)\right|\leq C/\lambda$ for some $C<\infty$ and so the right hand side in the above equation is bounded by $C/\lambda\,\|V_N-V_0\|_\infty\,\|dV\|_\infty=O_P\left(\sqrt{N}/\lambda\right)$ for any given $dV\in\mathbb{B}\left(\mathcal{Z}\times\left[0,\bar{\lambda}\right]\right)^D$. By similar arguments, we can show that $\left\|\dot{\Gamma}^{(z)}_{N,j}(V_N)-\dot{\Gamma}^{(z)}_{N,j}(V_0)\right\|_\infty=O_P\left(\sqrt{N}/\lambda\right)$ and $\left\|\dot{\Gamma}^{(\lambda)}_{N,j}(V_N)-\dot{\Gamma}^{(\lambda)}_{N,j}(V_0)\right\|_\infty=O_P\left(\sqrt{N}/\lambda\right)$. Theorem A.1 now yields the second part of the theorem. □

Proof of Theorem 4. We verify the conditions in part (ii) of Theorem B.1 with $\partial\mathcal{V}=\mathbb{C}^1_r(\mathcal{Z})^D$ and $r<\infty$ given in Theorem 3. First, by arguments similar to the ones in the analysis of $\mathcal{G}$ in the proof of Theorem 3, $\mathcal{G}'$ is Glivenko–Cantelli due to the Lipschitz property of $V_0(\psi_Z(U;z),\lambda)$ and the other components entering the function set under Assumption 2. Second, $\mathbb{C}^1_r(\mathcal{Z})^D$ has finite Bracketing number according to Theorem 2.7.1 in van der Vaart and Wellner (1996) and so is also Glivenko–Cantelli.

The rate result is an immediate consequence of Theorem A.1 together with Assumption 3. For the weak convergence result, we use the decomposition (5.8) where $\left\|\hat{V}_N-V_N\right\|_\infty=O_p(\rho_{\Pi,K})=o_P\left(1/\sqrt{N}\right)$ while the second term converges weakly according to Theorem 4. □

Appendix C. Additional numerical details for sieve method

Chebyshev basis functions

Chebyshev polynomials of the first kind have well-known good properties for approximating functions on bounded intervals. Recall that Chebyshev polynomials are defined on $[-1,1]$. We then choose $-\infty<z^{\min}<z^{\max}<\infty$ and define the kth basis function as follows for any $z\in\mathbb{R}$:

$$B_{c,k}(z)=\begin{cases}\cos\left((k-1)\arccos(T(z))\right), & |T(z)|\leq 1\\ (\text{sign}(T(z)))^k, & |T(z)|>1,\end{cases}$$

where $T(z)=2\frac{z-z^{\min}}{z^{\max}-z^{\min}}-1$ maps z into the interval $[-1,1]$. In particular, the basis functions are "truncated" and are set to one outside the interval $\left[z^{\min},z^{\max}\right]$. This is done to avoid any erratic extrapolation. We then choose the grid points $z_1,\ldots,z_M$ in (4.7) as the Chebyshev nodes in order to minimize the presence of Runge's phenomenon. Thus, $M=K$ in this case.

B-splines

We use cardinal B(asis)-splines to form our B-spline spaces, so they are represented by a knot vector with equidistant entries $(0,\frac{1}{M+1},\frac{2}{M+1},\ldots,\frac{M}{M+1},1)$, and the Cox–de Boor recursion

$$\begin{aligned}\bar{B}_{i,0}(z)&=\begin{cases}1 & \text{if } t_i\leq z<t_{i+1}\\ 0 & \text{otherwise}\end{cases}\\ \bar{B}_{i,k}(z)&=\frac{z-t_i}{t_{i+k}-t_i}\bar{B}_{i,k-1}(z)+\frac{t_{i+k+1}-z}{t_{i+k+1}-t_{i+1}}\bar{B}_{i+1,k-1}(z).\end{aligned}$$

For interpolation purposes we use the so-called Universal (Parameters) Method by Tjahjowidodo et al. (2017). This amounts to choosing the M grid points as the unique maximizers of all B-splines of degree $k\geq 1$, or any point if $k=0$ in which case we set it to the first K elements of the knot vector. The above are defined on the unit interval $[0,1]$ and so the final basis functions are chosen as

$$B_{c,k}(z)=\begin{cases}\bar{B}_k(T(z)) & 0\leq T(z)\leq 1\\ (\text{sign}(T(z))) & otherwise.\end{cases}$$

where now $T(z)=\frac{z-z^{\min}}{z^{\max}-z^{\min}}$

Table 2
Coefficients of tensor product Chebyshev basis functions in the 2D model of engine replacement for $K = J^2 = 25$, $N = 200$.

$J_1 \backslash J_2$	1	2	3	4	5
1	−38.4713	−4.4754	1.6176	−0.256420	−0.064960
2	−4.4754	1.9662e−14	−6.2341e−15	−1.1318e−15	2.5392e−15
3	1.6176	7.2256e−15	5.6179e−14	−2.0548e−14	−3.3049e−15
4	−0.2564	−1.0110e−14	−8.8673e−15	7.5672e−15	−2.9838e−15
5	−0.0649	4.0869e−15	−1.5251e−14	3.4571e−15	1.4218e−15

Table 3
Coefficients of tensor product 2nd order B-Spline basis functions in the 2D model of engine replacement for $K = J^2 = 25$, $N = 200$.

$J_1 \backslash J_2$	1	2	3	4	5
1	−21.9800	−27.1194	−30.0583	−31.1025	−31.506
2	−27.1194	−32.2589	−35.1977	−36.2419	−36.6455
3	−30.0583	−35.1977	−38.1366	−39.1808	−39.5844
4	−31.1025	−36.2419	−39.1808	−40.2250	−40.6285
5	−31.5060	−36.6455	−39.5844	−40.6285	−41.0321

Table 4
Coefficients of the ten basis functions, and their products, upon convergence with $K = 50^2$.

$J_1 \backslash J_2$	1	2	3	4	5	6	7	8	9	10
1	−52.200	−5.070	2.700	−1.170	0.478	−0.145	−0.006	0.038	−0.024	0.007
2	−5.070	−1.560	0.135	0.225	−0.057	−0.023	0.013	−0.006	0.004	0.001
3	2.700	0.135	−0.176	0.032	0.046	−0.021	−0.004	0.005	−0.002	0.001
4	−1.170	0.225	0.032	−0.087	0.019	0.019	−0.012	0.002	0.001	−0.001
5	0.478	−0.057	0.046	0.019	−0.038	0.010	0.009	−0.008	0.002	0.000
6	−0.145	−0.023	−0.021	0.019	0.010	−0.018	0.005	0.005	−0.004	0.002
7	−0.006	0.013	−0.004	−0.012	0.009	0.005	−0.009	0.003	0.002	−0.003
8	0.038	−0.006	0.005	0.002	−0.008	0.005	0.003	−0.005	0.002	0.001
9	−0.024	0.004	−0.002	0.001	0.002	−0.004	0.002	0.002	−0.003	0.001
10	0.007	0.001	0.001	−0.001	0.0002	0.002	−0.003	0.001	0.001	−0.002

References

Arcidiacono, P., Bayer, P., Bugni, F.A., James, J., 2013. Approximating high-dimensional dynamic models: Sieve value function iteration. Adv. Econom. 31, 45–95.
Bowman, A., Hall, P., Prvan, T., 1998. Bandwidth selection for the smoothing of distribution functions. Biometrika 85, 799–808.
Brumm, J., Scheidegger, S., 2017. Using adaptive sparse grids to solve high-dimensional dynamic models. Econometrica 85, 1575–1612.
Cai, Y., Judd, K.L., 2013. Shape-preserving dynamic programming. Math. Methods Oper. Res. 77 (3), 407–421.
Chen, V.C., 1999. Application of orthogonal arrays and MARS to inventory forecasting stochastic dynamic programs. Comput. Statist. Data Anal. 30, 317–341.
Chen, X., 2007. Large sample sieve estimation of semi-nonparametric models. In: Heckman, J.J., Leamer, E.E. (Eds.), Handbook of Econometrics, vol. 6. Elsevier.
Chen, X., White, H., 1999. Improved rates and asymptotic normality for nonparametric neural network estimators. IEEE Trans. Inform. Theory 45 (2), 682–691.
Fermanian, J.-D., Salanie, B., 2004. A nonparametric simulated maximum likelihood estimation method. Econometric Theory 20, 701–734.
Iskhakov, F., Jørgensen, T.H., Rust, J., Schjerning, B., 2017. The endogenous grid method for discrete-continuous dynamic choice models with (or without) taste shocks. Quant. Econ. 8 (2), 317–365.
Judd, K.L., Maliar, L., Maliar, S., Valero, R., 2014. Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. J. Econom. Dynam. Control 44, 92–123.
Keane, M., Wolpin, K.I., 1994. The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte Carlo evidence. Rev. Econ. Stat. 76, 648–672.
Kristensen, D., Salanie, B., 2017. Higher order properties of approximate estimators. J. Econometrics 198, 189–208.
Kristensen, D., Shin, Y., 2012. Estimation of dynamic models with nonparametric simulated maximum likelihood. J. Econometrics 167, 76–94.
Lizotte, D.J., 2011. Convergent fitted value iteration with linear function approximation. In: NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems, pp. 2537–2545.
Lumsdaine, R., Stock, J., Wise, D., 1992. Three models of retirement: Computational complexity versus predictive validity. In: Wise, D. (Ed.), Topics in the Economics of Aging. NBER: University of Chicago Press, pp. 21–60.
Ma, Q., Stachurski, J., 2020. Dynamic programming deconstructed: Transformations of the bellman equation and computational efficiency. Oper. Res. forthcoming.
McFadden, D., 1989. A method of simulated moments for estimation of discrete response models without numerical integration. Econometrica 57, 995–1026.
Munos, R., Szepesvari, C., 2008. Finite-time bounds for fitted value iteration. J. Mach. Learn. Res. 1, 815–857.
Newey, W.K., Smith, R.J., 2004. Higher order properties of GMM and generalized empirical likelihood estimators. Econometrica 72, 219–255.

Norets, A., 2010. Continuity and differentiability of expected value functions in dynamic discrete choice models. Quant. Econom. 1, 305–322.
Norets, A., 2012. Estimation of dynamic discrete choice models using artificial neural network approximations. Econometric Rev. 31, 84–106.
Pal, J., Stachurski, J., 2013. Fitted value function iteration with probability one contractions. J. Econom. Dynam. Control 37, 251–264.
Rivlin, T.J., 1990. Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory. Wiley-Interscience, New York.
Robert, C., Casella, G., 2013. Monte Carlo Statistical Methods. Springer Science & Business Media.
Rust, J., 1987. Optimal replacement of GMC bus engines: An empirical model of harold zurcher. Econometrica 55, 999–1033.
Rust, J., 1988. Maximum likelihood estimation of discrete control processes. SIAM J. Control Optim. 26 (5), 1006–1024.
Rust, J., 1997a. A comparison of policy iteration methods for solving continuous-state, infinite-horizon Markovian decision problems using random, quasi-random, and deterministic discretizations.
Rust, J., 1997b. Using randomization to break the curse of dimensionality. Econometrica 65, 487–516.
Rust, J., 2008. Dynamic programming. The New Palgrave Dictionary of Economics: Vol. 1–8. Springer, pp. 1471–1489.
Rust, J., Traub, J., Wozniakowski, H., 2002. Is there a curse of dimensionality for contraction fixed points in the worst case? Econometrica 70, 285–329.
Schumaker, L.L., 2007. Spline Functions: Basic Theory, third ed. Cambridge University Press.
Tjahjowidodo, T., et al., 2017. A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting. PLoS One 12 (3), e0173857.
van der Vaart, A.W., Wellner, J.A., 1996. Weak Convergence and Empirical Processes. Springer.
van der Vaart, A., Wellner, J., 2000. Preservation theorems for glivenko-cantelli and uniform glivenko-cantelli classes. In: Gine E., W.J. (Ed.), High Dimensional Probability II. Progress in Probability, vol. 47.