Divide and Conquer: Recursive Likelihood Function Integration for Dynamic Discrete Choice Models with Serially Correlated Unobserved State Variables

Gregor Reich*

Dept. of Business Administration
University of Zurich
Moussonstrasse 15
8044 Zurich, Switzerland
gregor.reich@uzh.ch

February 15, 2016

Abstract

This paper develops a method to efficiently estimate dynamic discrete choice models with $AR(n)$ serial correlation in the errors. To evaluate the likelihood function, we decompose the integral over the unobserved state variables into a series of lower dimensional integrals, and recursively approximate them using numerical quadrature and interpolation. Finally, we maximize the likelihood function using a nested fixed point algorithm. We show that this procedure has very favorable numerical properties: First, the computational complexity grows linearly in time, which makes the integration over hundreds and thousands of periods well feasible. Second, we prove that the numerical error is accumulated sub-linearly over time; consequently, using highly efficient and fast converging numerical integration rules for low and medium dimensions, such as Gaussian quadrature, the numerical error can be well controlled even for very large numbers of periods. We apply this method to the bus engine replacement model of Rust [Econometrica, 55 (5): 999–1033, (1987)]: first, we verify the algorithm’s ability to recover the parameters in an extensive Monte Carlo study with simulated datasets; second, we estimate the model using the original data, finding significant serial correlation for some subsamples.


Area of review: Computational Economics.

*I am heavily indebted to my advisor Karl Schmedders as well as to Ken Judd, John Rust, and Che-Lin Su for their support and guidance in this project. I would also like to thank Greg Crawford, Philipp Eisenhauer, Katharina Erhardt, Dennis Kristensen, János Mayer, Bertel Schjerning, Ole Wilms, and seminar audiences at the University of Chicago, Hoover Institution, University of Zurich, and the 68th European Meeting of the Econometric Society for helpful comments and suggestions. Finally, I thank Dave Brooks for editorial comments on the manuscript. An earlier version of this paper circulated as “Divide and Conquer: A New Approach to Dynamic Discrete Choice with Serial Correlation”.

1
1 Introduction

This paper develops a new approach to efficiently estimating dynamic discrete choice models with serially correlated unobserved state variables such as $AR(n)$ errors: First, we show how to combine some well-known methods from the literature, such as Gaussian quadrature, adaptive grid refinement, and methods for large sparse non-linear systems of equations, in order to efficiently approximate the solution to the dynamic optimization problem of the agent. Second, we develop a method to decompose and approximate the integral over the unobserved state variables that appears in the likelihood function, which has previously been considered infeasible for approximation by highly efficient deterministic integration schemes such as Gaussian quadrature; we call this procedure recursive likelihood function integration (RLI). Finally, we apply the method to the well known bus engine replacement model of Rust (1987) to estimate its parameters in the presence of serially correlated errors by using maximum likelihood, using a nested fixed point algorithm. We first apply the method to artificial data sets in order to verify the algorithm’s ability to recover the parameters of the model. Then, we estimate the model using the original data set, and we find significant serial correlation for some of the subsamples of the original dataset.

Dynamic discrete choice models (DDCMs) have become a popular instrument for the econometric analysis of decision making: First, many (individual) economic decisions we actually can observe are in fact discrete in nature, for example the choice of a brand or medical treatment. Second, the underlying utility maximization problem of the agents is often dynamic in nature: decisions made today not only influence today’s payoffs, rather they also influence future decisions and payoffs. By capturing these key facts, DDCMs have a wide range of uses; for recent surveys see, for example, Aguirregabiria and Mira (2010) and Keane et al. (2011).

The majority of contributions to the literature on the estimation of DDCMs make strong distributional assumptions about the errors, or, used synonymously, the unobserved state variables. Probably the most prominent example is extreme value type I $EV_1$ iid distributed errors; obviously implied by the $EV_1$ iid assumption, but usually stated explicitly by a conditional independence assumption (CI), the errors are assumed to be serially uncorrelated.

However, there exists a wide consensus that these assumptions are not made based on the existence of much empirical evidence, but rather for numerical tractability: $EV_1$ iid errors and CI often induce closed form solutions to potentially high dimensional integrals that arise in the solution to the dynamic optimization problem and in the choice probabilities in the likelihood function. These closed form solutions go back to the work of McFadden (1974, 1981) and Rust (1987).

While relaxing the $EV_1$ assumption has attracted some attention — for example, Larsen et al. (2012) test the statistical significance of allowing for more general distributions in the Rust (1987) model — several papers have developed integrated methods to estimate models without the CI assumption, thus allowing for a general notion of serially correlated unobserved state variables. Among those are (listed alphabetically) the expectation–maximization algorithm based on conditional choice probability estimation of Arcidiacono and Miller (2011), the particle filter method of Blevins (2011), the simulation and interpolation method of Keane and Wolpin (1994), the Markov chain Monte Carlo approaches of Norets (2009, 2012), and the application
of Gaussian quadrature and interpolation as discussed in Stinebrickner (2000).

While the approaches to DDCM estimation with serial correlation are diverse, most of them share a common challenge:

“the likelihood function for a DDCM can be thought of as an integral over latent variables (the unobserved state variables). If the unobservables are serially correlated, computing this integral is very hard.” (Norets, 2009)

This conclusion follows from the fact that the integral over serially correlated errors really has dimensionality proportional to the time horizon of the data, which itself can be arbitrarily large; moreover, no closed form solution for this integral exists in general.

A popular numerical approach to high dimensional integration is Monte Carlo integration (MC), because its approximation error is independent of the dimensionality of the integral. However, the approximation error usually decreases only very slowly as the number of integration nodes is increased: in order to reduce the estimated error by one order of magnitude, one usually has to increase the number of nodes by two orders of magnitude.$^1$ Consequently, MC is a natural choice for high dimensional integrals, but only if the integral has no structure that could potentially be exploited by more efficient methods. In contrast to MC, many quadrature rules exist that have much faster decaying errors but usually inefficient (in the worst case exponential) scaling in the dimensionality; a popular example is that of Gaussian quadrature rules, extended for multiple dimensions by the product rule.

The approach followed in this paper is to identify and exploit the structure that is present in the integral over the unobserved state variables in the likelihood function: given the serial dependence of the unobserved state variables is Markov, as, for example, with \( AR(n) \) errors, the time structure allows us to decompose the high dimensional integral over the time horizon, and rewrite it as a sequence of low dimensional integrals. Then, we can approximate this sequence to high accuracy, using highly efficient approximation schemes for low dimensional integrals, including Gaussian quadrature. We show that the computational complexity of computing this integral is linear in the time horizon.

In order to evaluate the likelihood function, we need to compute the solution to the dynamic optimization problem of the agent, namely the expected value as a function of the state variables. In the presence of serial correlation, approximating the solution to the dynamic problem involves different numerical tasks: First, taking the expectation of the value function is an integration over the unobserved state variables, for which, in contrast to the \( EV1 \ iid \) case, no closed form solution exists. Consequently, we have to approximate these integrals numerically, and we discuss how to apply Gaussian quadrature, which was first proposed and successfully implemented in the context of DDCMs by Stinebrickner (2000). Second, we have to approximate the expected value function as a continuous function of the unobserved state variables; in the \( EV1 \ iid \) case, this step was not necessary, because the unobserved state variables are integrated out in the closed form solution. Different approaches to value function approximation have been proposed (see, for example, Cai and Judd, 2013; Judd, 1998; Rust, 1996), and to stay flexible and generic we use interpolation over an adaptively refined grid, as proposed by Grüne and Semmler (2004). Third,

$^1$Formally, the variance of the Monte Carlo estimate of an integral \( I \) is proportional to \( n^{-\frac{1}{2}} \), where \( n \) is the number of integration nodes.
since the expected value function is only defined implicitly by the fixed point of the dynamic programming operator, we need to solve a non-linear system of equations in order to obtain an approximation of the expected value. While also under the EV1 iid assumption, the expected value is the solution to a fixed point problem, the system becomes much larger in the presence of serial correlation, and thus we discuss suitable methods. Finally, we solve the maximum likelihood problem using a nested fixed point (NFXP) algorithm, which is interconnected with the grid refinement process of the expected value function approximation.

As an application, we estimate the bus engine replacement model of Rust (1987) with serially correlated errors. One motivation for serial correlation in this model is a test for misspecification from the original paper, which leads to the following conclusion:

“for groups 1, 2, and 3 and the combined groups 1–4 there is strong evidence that (CI) does not hold. The reason for rejection in the latter cases may be due to the presence of ‘fixed-effects’ heterogeneity which induces serial correlation in the error terms.” (Rust, 1987)

Testing for statistical significance of serially correlated errors we find that in some subsamples of the original dataset we can reject serially uncorrelated errors. Also, the parameter estimates vary substantially; their relative sizes however are rather stable. For readability, the development of the algorithm is closely related to the model under consideration; however, note that it is generic with respect to DDCMs with Markov serial dependence in the unobserved state variables.

The remainder of this paper is organized as follows: Section 2 describes the bus engine replacement model of Rust (1987), and introduces the notion of serial correlation of the errors, which is used throughout the paper. Section 3 first develops a numerical procedure to solve the dynamic programming problem of the agent, then introduces a method, namely the recursive likelihood function integration (RLI) algorithm, to decompose the likelihood function such that it can be computed using highly efficient quadrature rules, and finally describes the likelihood maximization algorithm. Section 4 presents the estimation results. Section 5 concludes and states the agenda for future research.

2 The Bus Engine Replacement Model

In the bus engine replacement model of Rust (1987), an agent repeatedly makes decisions about the maintenance of a fleet of buses: Each period, he observes the state of each of the buses, including mileage, damage, signs of wear, etc. Based on these observations, he decides whether to do regular maintenance work only, or a general overhaul; the latter is usually referred to as a replacement of the engine. While the engine replacement causes a fixed cost of $RC$ plus some

---

2One can also think of serial correlation as a “generic feature” in this context: In optimal stopping problems, such as the bus engine replacement model, the replacement decision is expected to happen rarely. If the explanatory power of the model in terms of observed states is low, the probability of stopping is small for all possible observed states. Thus, the observed decisions are mostly driven by tail events of the unobserved state variables. However, this fact contradicts the assumption that decisions are modeled to be dynamic, because in a model without serial correlation, these events are unforeseeable, single period shocks. With the introduction of serial correlation, these shocks have persistent effects, which can be anticipated by the agent. For example, a jump in maintenance costs still comes as a surprise to the agent, but — once incurred — its effect on future periods can influence decisions to a large extent.
random component, the cost of regular maintenance is a function $c(\cdot)$ that is increasing in the current mileage state, plus some random component.

Formally, the agent faces single period costs (or negative utility) for each individual bus

$$u_θ(i, x_t) + ε_t(i), \quad u_θ(i, x_t) = \begin{cases} -RC & \text{if } i = 1 \\ -c(x_t, θ_1) & \text{if } i = 0 \end{cases}$$

(1)

where $i$ is the decision variable, with $i = 1$ indicating engine replacement, and $i = 0$ regular maintenance; $ε_t(i)$ is a random utility component that is observed by the agent for all possible choices before making the actual decision; $x_t$ is the mileage of the individual bus at time $t$, which is reset to 0 after an engine replacement. The replacement cost $RC$, as well as the cost function parameter $θ_1$, are both parameters to be estimated. The maintenance cost function is assumed to be of the form $c(x_t, θ_1) = 0.001 θ_1 x_t$. From the econometrician’s point of view, mileage at the time of decision and the decision itself are observable for each bus and each time period. The random utility component however is only observable to the agent, but not to the econometrician; consequently, it is often referred to as the unobserved state variable.

For the agent, the decision problem is how long to run a bus with regular maintenance only, with increasing costs induced by increasing mileage, and when to replace its engine, thus facing the one-time replacement cost, but at the same time reducing the maintenance costs in the future because mileage is reset to 0. Assuming that the agent behaves dynamically optimally, the Bellman equation defines the value per bus as a function of its mileage state and the random utility components

$$V_θ(x_t, ε_t) = \max_{i ∈ \{0, 1\}} \left\{ u_θ(i, x_t) + ε_t(i) + βE[V_θ(x_{t+1}, ε_{t+1})|i, x_t, ε_t] \right\}.$$

(2)

The conditional expected continuation value in (2) is defined by

$$E[V_θ(x_{t+1}, ε_{t+1})|i, x_t, ε_t] = \int_{(x_t+1, ε_{t+1})} V_θ(x_{t+1}, ε_{t+1}) Pr(x_{t+1}, ε_{t+1} | i, x_t, ε_t, θ) d(x_{t+1}, ε_{t+1})$$

(3)

with subscript $θ$ denoting the dependence of the value function on the parameter values $RC$ and $θ_1$.

The original model makes the following conditional independence (CI) assumption regarding the joint probability of the state variables:

$$Pr(x_{t+1}, ε_{t+1}|i, x_t, ε_t, θ) = Pr(ε_{t+1}|x_{t+1}) Pr(x_{t+1}|i, x_t)$$

(4)

Assumption (4) ensures that (i) the mileage state transition is — conditional on the decision $i$ — independent of the random utility component, and (ii) that the random utility components are serially uncorrelated. If the CI assumption holds, and if moreover the random utility components $ε(i)$ are distributed extreme value type I ($EV1$) iid, the integral in (3) has a closed form solution. However, in order to allow for serial correlation in $ε$, while keeping (i), we assume

$$Pr(x_{t+1}, ε_{t+1}|i, x_t, ε_t, θ) = Pr(ε_{t+1}|ε_t, x_{t+1}, θ) Pr(x_{t+1}|i, x_t).$$

(5)
Note that assumption (5) allows the transition process of the mileage state, \( Pr(x_{t+1} \mid i, x_t) \), to be estimated independently from the other model parameters — as in the original model.\(^3\) We use discretized mileage, and thus the integral over future mileage states in (3) becomes a sum:

\[
E[V_\theta(x_{t+1}, \varepsilon_{t+1}) \mid i, x_t, \varepsilon_t] = \sum_{x_{t+1}, \varepsilon_{t+1}} V_\theta(x_{t+1}, \varepsilon_{t+1}) \, Pr(d\varepsilon_{t+1} \mid \varepsilon_t, x_{t+1}, \theta) \, Pr(x_{t+1} \mid i, x_t) \quad (6)
\]

A choice for serial correlation in the unobserved state variables that is frequently used in the literature is the AR(1) process. More specifically, similar to Norets (2009), we define

\[
\varepsilon_t(0) = \rho\varepsilon_{t-1} + \tilde{\varepsilon}_t(0), \quad \tilde{\varepsilon}_t(0) \sim q(\cdot) \text{ iid}
\]

\[
\varepsilon_t(1) = \tilde{\varepsilon}_t(1), \quad \tilde{\varepsilon}_t(1) \sim q(\cdot) \text{ iid}
\]

where \( q(\cdot) \) is a density function with zero mean, and \( \rho \) is the additional parameter of the estimation; furthermore, we assume that \( \varepsilon_0(i) \) is distributed with density \( q(\cdot) \). Thus, we only assume the random utility component of regular maintenance to be serially correlated.\(^4\) It is important to note that definition (7) nests the original model for \( \rho = 0 \), and the density function \( q(\cdot) \) being extreme value type 1 \( EV1 \).

Given that mileage state \( x_t \) and decision \( i_t \) are observable for all buses, but random utility components \( \varepsilon_t \) are not, the aim is to estimate this model’s parameter \( \theta = \{\theta_1, RC, \rho\} \), given the data \( \{x_t, i_t\}^{T}_{t=0} \), by maximum likelihood estimation.

### 3 Computation and Estimation

The following subsections develop the numerical methods necessary to estimate the bus engine replacement model of Rust (1987), with serially correlated unobserved state variables: First, we show how to approximate the solution to the dynamic problem, using numerical quadrature and interpolation over an adaptively refined grid, and solving a large non-linear system of equations. Then, we develop a method to decompose and approximate the integral over the unobserved state variables that appears in the likelihood function, and approximate it using highly efficient Gaussian quadrature, using the recursive likelihood function integration (RLI) algorithm. Finally, we describe a nested fixed point algorithm to obtain maximum likelihood estimates of the model parameters.

Note that this procedure is not specific to the model under consideration, but rather is generic with respect to DDCMs with Markov serial dependence in the unobserved state variables. Also note that the methodology used to approximate the solution to the dynamic problem is independent of the likelihood function integration, for which the new RLI algorithm is proposed.

---

\(^3\)Since one can estimate the mileage transition process \( Pr(x_{t+1} \mid i, x_t) \) — referred to as parameter \( \theta_3 \) in the original model — independently from \( \theta = \{\theta_1, RC, \rho\} \), and moreover, since it is exactly the same as in Rust (1987) (because it is not affected by the serial correlation in the unobserved state variables) we ignore this aspect of the bus engine replacement model in the remainder of this paper.

\(^4\)While the assumption that serial correlation is only present for regular maintenance utility shocks is computationally more general as we will point out in Section 3.1, one can also argue that it is easier to motivate serial correlation as a feature with a real counterpart in this case, because the errors are bus specific by construction; for example, one might think of a bus having some larger damage that persistently increases maintenance costs until the next general overhaul occurs.
3.1 The Expected Value Function

From (2) it is clear that in order to obtain the value function, we need to compute its conditional expectation. In fact, the computation of the likelihood function actually requires the expected value rather than the value itself (see Section 3.2). Thus, this section describes the steps necessary to numerically approximate the expected value as a function of all possible states:

\[
EV_\theta(x, \varepsilon) = \sum_{x', \varepsilon'} \max_{i \in \{0, 1\}} \left\{ u_\theta(i, x') + \varepsilon'(i) + \beta EV_\theta(x', \varepsilon') \right\} Pr(d\varepsilon' | \varepsilon, \theta) Pr(x' | x) \equiv T(EV_\theta)(x, \varepsilon) \tag{8}
\]

Keeping the original time structure of the expectation (6) in mind, the expectation on the leftmost side of (8) is — strictly speaking — taken at time \( t \), while the one on the right hand side (within the max operator) is taken at time \( t + 1 \). But since the value function and its expectation are time invariant, given state \((x, \varepsilon)\), the same unknown function \( EV_\theta \) appears on both the left and the right sides of the equation. Therefore, \( EV_\theta \) is the solution to the functional equation

\[
EV_\theta(x, \varepsilon) = T(EV_\theta)(x, \varepsilon) \tag{9}
\]

and thus a fixed point of the non-linear operator \( T \). Moreover, since \( T \) can be shown to have the contraction mapping property (Rust, 1988), this fixed point is unique and attractive.

The numerical approximation of (8) involves three main computational tasks:

1. **Numerical integration.** In contrast to the case of extreme value type I iid distributed unobserved state variables, no closed form solution to the integral (8) exists; thus, we have to approximate it by numerical quadrature. A variety of methods for multi-dimensional integration exists; see, for example, chapter 7 of Judd (1998) for an overview, or chapter 4 of Press et al. (2007) for an implementation oriented approach. Throughout the paper, we use Gaussian quadrature, which is known to be very efficient for the integration of functions that can be well approximated by a polynomial. While this condition is obviously violated for the value function (because of the kink potentially induced by the max-operator), one can still show Gaussian schemes to be convergent for any Riemann integrable function, and, moreover, they are reported to often outperform other widely used integration schemes, even in the presence of singularities; see Judd (1998) and the literature cited therein. Also, Stinebrickner (2000) successfully applied the Gaussian quadrature rules to expected value function approximation for DDCMs with serial correlation.

2. **Maximization.** Generally, there is one more task necessary, namely maximizing the utility and continuation value, in order to obtain the current value as a function of the states. However, since the choice set is discrete and small, we carry out the maximization by complete enumeration.
The \( n \)-node Gaussian quadrature rule approximates

\[
\int_a^b f(y)w(y)\,dy \approx \sum_{i=1}^{n} \omega_i f(y_i) \tag{10}
\]

where \( w(y) \) is a non-negative weighting function with finite integral (including unity for \( |a|, |b| < \infty \)). The integration nodes \( y_i \) are the roots of the degree \( n \) polynomial of the family of polynomials that are mutually orthonormal with respect to weighting function \( w(y) \).\(^6\) The corresponding weights \( \omega_i \) are chosen such that every polynomial of degree \( 2n - 1 \) is integrated exactly; for the corresponding formulas, see, for example, Kythe and Schäferkotter (2005). Since both nodes and weights should be computed to high accuracy, they are often tabulated for some frequently used families of orthonormal polynomials.

When taking expectations of functions of continuous random variables, the integration problem (10) arises naturally, with the density function being used as weighting function \( w(x) \). Obviously, this approach requires the availability of polynomials that are orthonormal with respect to the density function in use. For some distributions, these families are well known, such as the Hermite polynomials for normally distributed random variables. For most other distributions however, the necessary polynomials (and their roots) are unknown, and have to be computed first. Alternatively, one can map the support of the corresponding density function to \([-1, 1]\) by a change of variable,\(^7\) and approximate the resulting integral using the Gaussian rule based on Legendre polynomials, which are orthonormal with respect to the unity weighting function on \([-1, 1]\). Using this procedure, we found that expectations of extreme value distributed random variables can be approximated quite efficiently.

Directly approximating (8) by Gaussian quadrature has a potential caveat, since it would require one to find polynomials that are orthonormal with respect to the conditional probabilities, \( \Pr(\varepsilon' | \varepsilon) \), and thus different nodes and weights for each \( \varepsilon \). Consequently, we reformulate the integral in (8) in terms of the unconditional probability \( \Pr(\hat{\varepsilon}'(i)) \) (or, equivalently, the density function \( q(\cdot) \) of \( \hat{\varepsilon}'(i) \)),

\[
\int_{\hat{\varepsilon}'(0)}^{\hat{\varepsilon}'(1)} \max\{ u(0, x') + \rho \varepsilon(0) + \hat{\varepsilon}'(0) + \beta \text{EV}_\theta(x', (\rho \varepsilon(0) + \hat{\varepsilon}'(0), \varepsilon'(1)),
\]

\[
\quad u(1, 1) + \hat{\varepsilon}'(1) + \beta \text{EV}_\theta(1, (0, \varepsilon'(1))))\} \Pr(d\hat{\varepsilon}'(1)) \Pr(d\hat{\varepsilon}'(0)) \tag{11}
\]

and compute (or look up) one single set of nodes and weights for weighting function \( \Pr(\hat{\varepsilon}'(i)) \).\(^8\)

Since the integration in (11) is of dimension \( N = 2 \),\(^9\) but Gaussian rules are per se one

\(^6\) A family of polynomials \( \{\varphi_k(y)\}_{k=0}^{\infty} \) with inner product \( \langle \varphi_k, \varphi_l \rangle = \int_{a}^{b} \varphi_k(y)\varphi_l(y)w(y)\,dy \) is orthonormal with respect to weighting function \( w(y) \) on \([a, b]\) if \( \langle \varphi_k, \varphi_l \rangle = 0 \quad \forall k, l : k \neq l \), and \( \langle \varphi_k, \varphi_k \rangle = 1 \quad \forall k \).

\(^7\) For example, if the inverse of the cumulative distribution of a distribution with density \( w(y) \), \( W^{-1}(y) \) exists, one can apply the following change of variables: \( \int_{-\infty}^{\infty} f(y)w(y) = \int_{-1}^{1} f(W^{-1}(y)) \).

\(^8\) Equation (11) silently assumes that after a replacement, the series of serially correlated unobserved states is reset to its mean, 0. Thus, \( \varepsilon(i) \) in the first period after an engine replacement is distributed according to density \( q(\cdot) \) again.

\(^9\) The dimension of the integration over the unobserved state variable in DDCMs is usually \((N - 1)\)-dimensional, because the decisions of the agents in the model are driven by utility differences rather than levels. In this case however, since we assume that serial correlation is only present in one dimension of the error, the reformulation of the model in terms of the differences of errors does not reduce dimensionality. Thus, the integration must be carried out over all the \( N \) dimensions.
dimensional, we use them extended to $N$ dimensions by the product rule, which generalizes (10) to $N$ dimensions by

$$
\int_{[a,b]^N} f(y^1, \ldots, y^N) \prod_{i=1}^{N} w_i(y_i) \, \text{d}(y^1, \ldots, y^N) \approx \sum_{i_1=1}^{n} \cdots \sum_{i_N=1}^{n} f(y^1_{i_1}, \ldots, y^N_{i_N}) \prod_{j=1}^{N} \omega_j^{i_j}
$$

where $f : \mathbb{R}^N \to \mathbb{R}$, $w_i : \mathbb{R} \to \mathbb{R}$ is the weighting function for dimension $i$, and $y_i^j$ and $\omega_j^i$ are the nodes and weights of the corresponding one-dimensional Gaussian rule (indexed by $j$), applied to dimension $i$.\footnote{Note that in order to use the product rule (12) to compute expectations, the dimensions of the random variable must be mutually independent. For more general multivariate distributions, see, for example, Jäckel (2005).}

**Function approximation.** Generally, the expected value function is a continuous function of $\varepsilon$, and we need to approximate it as such, but by a finite number of parameters only. Assume for the moment that we can evaluate an unknown function $f(y)$ at arbitrary points. Then, we can choose a set of nodes $y_i \in [a, b]$, and construct an interpolating function $\hat{f}(y)$, such that $f(y_i) = \hat{f}(y_i) \forall y_i$. Obviously, we want to choose $\hat{f}(y)$ such that $|f(y) - \hat{f}(y)|$ is “small everywhere”, not just at the interpolation nodes $y_i$. More formally, we want to control the interpolation error $\sup_{y \in [a, b]} |f(y) - \hat{f}(y)|$.

A general, but computationally rather expensive approach to node choice is adaptive procedures: given some interpolant $\hat{f}^{(h)}(y)$, we evaluate the quality of approximation, $|f(y) - \hat{f}^{(h)}(y)|$, at different values of the argument (different from $y_i$), and we insert new nodes where the approximation quality is poor; then, we construct a new interpolant $\hat{f}^{(h+1)}(y)$ on the set union of old and new nodes. This procedure is iterated until some convergence criterion is met. Adaptive methods are particularly well suited for functions with “difficult” shape properties, for example functions with greatly varying curvature, kinks, or discontinuities, and to explicitly control the approximation error. For the actual interpolation over such a grid, piecewise polynomial interpolation, such as piecewise linear interpolation (PLI) or higher order splines, proved to be a reliable choice.

Since we want to have direct control over the error of the approximation of $EV_\theta$, we choose an adaptive approximation method; in particular, we want to assure uniform approximation quality for different values of $\theta$, in order to compute the corresponding likelihood function values to high accuracy. Therefore, we employ the method of Grün and Semmler (2004), which repeatedly refines an interpolation grid until a global approximation error criterion is met. At this point, it is important to note that we cannot directly evaluate the true (but unknown) expected value function $EV_\theta$, because it is only implicitly defined by (9). Fortunately, to discuss this grid adaption method, it is sufficient to assume that the method is supplied with an approximation $\overline{EV}_\theta^{(h)}(\cdot; a)$ from the previous iteration of the adaption process, which is now explicitly parametrized by the finite-dimensional vector $a \in \mathbb{R}^A$. Let $\Gamma_\theta^{(h)}$ be the grid at the beginning of iteration $h$. For each cell\footnote{In this context, cell $c_l$ of an $n$-dimensional grid $\Gamma$ is defined as the hypercube spanned by $\{y_j \in \Gamma : y^*_j \leq y^k_j \leq \min\{y^*_j : y^k_j < y^k_j\}, k = 1, \ldots, n\}$, where $y^k$ is the $k$th element (dimension) of the vector $y$.} $c_l$ of grid $\Gamma_\theta^{(h)}$, we approximate the solution to the
following optimization problem:\textsuperscript{12}

\[
\eta_l = \max_{\varepsilon \in c_l} \left| \text{EV}_{\theta}^{(h)}(x, \varepsilon; a) - T(\text{EV}_{\theta}^{(h)})(x, \varepsilon; a) \right|
\]

Then, Grüne (1997) showed that the maximum error over all cells, \( \eta = \max_l \{\eta_l\} \), defines an approximation error bound by

\[
\max_{x \in X, \varepsilon \in \mathbb{R}^N} \left| \text{EV}_{\theta}(x, \varepsilon) - \hat{\text{EV}}_{\theta}^{(h)}(x, \varepsilon; a) \right| \leq \eta \frac{1}{1 - \beta}
\]

where \( \text{EV}_{\theta} \) represents the true (but unknown) expected value function. The method of Grüne and Semmler (2004) inserts new nodes into those cells \( c_l \) where the corresponding error \( \eta_l \) is larger than some threshold. Finally, we construct new interpolant \( \hat{\text{EV}}_{\theta}^{(h+1)}(\cdot; a) \) on the refined grid \( \Gamma_{\theta}^{(h+1)} \). (In order to parametrize it, we need to solve for the fixed point (9), which we will discuss shortly.) This procedure is repeated until the maximum (global) approximation error \( \eta(1 - \beta)^{-1} \) is smaller than the desired approximation error, \( \bar{\eta} \).

One particular advantage of the method of Grüne and Semmler (2004) is that it not only allows for refinement, but easily extends to grid coarsening, by identifying and removing nodes that do not increase approximation accuracy. Combining coarsening and refinement, we can construct a grid updating procedure, which can be integrated with a nested fixed point algorithm (NFXP). In NFXP, the likelihood maximization (“outer loop”) repeatedly feeds different values of \( \theta \) into the expected value function approximation (“inner loop”); thus, rather than building up from scratch an interpolant for each new value of \( \theta^{(k+1)} \), it can be obtained from updating an interpolant that has previously been built for some other value \( \theta^{(k)} \) (see Section 3.3 below).

Note that due to the fact that serial correlation is only allowed in \( \varepsilon(0) \), \( \text{EV}_{\theta}(x, \varepsilon) \) is constant in \( \varepsilon(1) \). Consequently, we only need to approximate it as a one-dimensional function of \( \varepsilon(0) \). Therefore, we can use piecewise linear interpolation to construct \( \hat{\text{EV}}_{\theta} \). However, the methodology generalizes to higher dimensions by replacing PLI with multi-dimensional interpolation.

Finally note that, since — in this formulation of the model — mileage has been discretized, we need to approximate \( \text{EV}_{\theta} \) as a separate continuous function of \( \varepsilon \) for each mileage state \( x \in X \) simultaneously; thus, \( \hat{\text{EV}}_{\theta}(\cdot; a) \) is really a set of interpolants. If, in contrast, mileage would enter the model as a continuous variable, \( \hat{\text{EV}}_{\theta}(\cdot; a) \) would rather be a single 2-dimensional interpolant. However, discrete mileage is necessary to nest the original model without serial correlation as a special case.

\textbf{Non-linear system.} The last few paragraphs discussed the choice of a function approximation scheme and interpolation grid creation, but left out how to actually evaluate the unknown function \( \text{EV}_{\theta} \), which is only implicitly defined as the fixed point of \( T \). While this fixed point is generally a continuous function, its substitution by an approximating interpolant \( \hat{\text{EV}}_{\theta}(\cdot; a) \) simplifies the problem to a non-linear system of \( D \) equations in \( A \) unknowns,

\[
\hat{\text{EV}}_{\theta}(x, \varepsilon; a) = T(\hat{\text{EV}}_{\theta})(x, \varepsilon; a) \quad \forall (x, \varepsilon) \in \Gamma_{\theta}, a \in \mathbb{R}^A
\]

\textsuperscript{12}Note that since the model is already discretized in terms of mileage state \( x \), finding the maximum error within each cell does not explicitly involve \( x \); rather, one has to carry out the error estimation for all possible mileage states independently.
where \( D \) is the number of elements in \( \Gamma_\theta \), and thus each \((x, \varepsilon) \in \Gamma_\theta \) defines one equation of (15), and the parameters \( a \) of the interpolant are the variables. From the parameter vector \( a^* \) that solves (15), we can directly construct the interpolant \( \hat{EV}_\theta(\cdot; a^*) \). This procedure is known as collocation, which is a particular variant of a projection method for the approximation of functions that are defined by functional equations; see Judd (1998), chapter 11. Finally, we compute the approximation error of \( \hat{EV}_\theta(\cdot; a^*) \) as defined by (14); if it is sufficiently small (smaller than \( \bar{\eta} \)), we accept our approximation of \( EV_\theta \); otherwise, we refine the interpolation grid \( \Gamma_\theta \), and solve (15) for the new grid.

Similar to Rust (1987), we use methods that directly solve the non-linear system

\[
\hat{EV}_\theta(x, \varepsilon; a) - T(\hat{EV}_\theta)(x, \varepsilon; a) = 0 \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A
\]

(16)
to high accuracy. Given the accuracy needs of our application, Newton (or quasi-Newton) methods are particularly interesting, because they show quadratic (superlinear) convergence close to the solution under some conditions.\(^{13}\) However, these methods require the evaluation of the Jacobian matrix \( J \) of the non-linear system (16), which is generally of size \( D^2 \), and thus can be prohibitively expensive to compute for large systems. In particular, given an adaptively refined grid, the size of \( J \) can become an issue since the number of equations of (16) is defined by the number of nodes in \( \Gamma_\theta \), and thus the system grows larger as the grid is refined. However, analogously to the original model, if the Markov transition matrix of the discrete states is sparse, \( J \) is also sparse; thus, using (quasi-)Newton methods can still be feasible because the number of non-zero elements in the Jacobian grows much more slowly than the number of grid nodes. Figure 1 illustrates the sparseness pattern of our problem.

To numerically solve the fixed point problem (9), we either use the “ipopt” package (Wächter and Biegler, 2005), in conjunction with the “pardiso” sparse linear solver (Schenk and Gärtner, 2004), or the quasi-Newton trust-region method of the R-package “nleqslv” (Hasselman, 2014), depending on the size of the problem.

Figure 2 plots an example of the expected value function, where each of the black lines represents the expected value as a function of \( \varepsilon(0) \), for a particular value \( x \). We want to emphasize again that the procedure to compute an approximation of \( EV_\theta(x, \varepsilon) \) as presented in this section easily generalizes to other models, with an arbitrary number of decisions \( N \), and serial correlation in all dimensions of the unobserved state variables, by choosing a multi-dimensional interpolation scheme.

### 3.2 The Likelihood Function

In this section, we derive the likelihood function for the bus engine replacement model with serially correlated unobserved state variables, and formulate it such that the dimensionality of the numerical integration only depends on the number of choices \( N \), and not on the time horizon of the observation, \( T \). In a second step, we provide a numerical procedure to solve this

\(^{13}\)Loosely speaking, quadratic convergence means that, close to the solution, the number of correct digits of the result roughly doubles in every Newton step. More formally, suppose that for \( f: \mathbb{R}^n \to \mathbb{R}^n \), a solution \( y^* \) to the system \( f(y^*) = 0 \) exists, the Jacobian function \( J: \mathbb{R}^n \to \mathbb{R}^{n \times n} \) is Lipschitz continuous, and the Jacobian matrix at the solution, \( J_f(y^*) \) is non-singular. Then, if \( y^{(0)} \) is sufficiently close to the solution \( y^* \), the residual decays quadratically for each Newton iteration, thus \( \exists K > 0 : \|y^{(k+1)} - y^*\| \leq K\|y^{(k)} - y^*\|^2 \).
Figure 1: Sparseness pattern of the Jacobian of the non-linear system (16).

Figure 2: The expected value function $EV_\theta(x, \varepsilon)$ for $\rho = 0.6$, $RC = 14$, $\theta_1 = 2$, and the density of $\tilde{\varepsilon}(i)$, $q(\cdot)$, being $EV1$. 
formulation by recursive likelihood function integration (RLI) to high accuracy, using standard deterministic quadrature rules.\textsuperscript{14} It is important to note that this reformulation is not specific to the Rust (1987) model, but generically applies to DDCMs with Markov serial dependence in the unobserved state variables.

The likelihood function of one individual bus derives as follows:

\[ L(\theta | \{x_t, i_t\}_{t=0}^T) = \int \cdots \int \Pr(\{x_t, i_t, \varepsilon_t\}_{t=0}^T | \theta) \, d\varepsilon_0 \cdots d\varepsilon_T \]  

(17)

The likelihood function of the full panel computes as the product of the likelihood functions of the individual buses, since the state variables are assumed to be independently distributed across buses. Incorporating the assumption that all state transitions are Markov, we can factorize the probability of observing a particular time series as

\[ \Pr(\{x_t, i_t, \varepsilon_t\}_{t=0}^T | \theta) = \prod_{t=1}^T \Pr(x_t, i_t, \varepsilon_t | x_{t-1}, i_{t-1}, \varepsilon_{t-1}, \theta). \]  

(18)

We can further decompose the joint transition probability in (18), using the fact that, given \( x_t \) and \( \varepsilon_t \), \( i_t \) is independent of \( i_{t-1}, \varepsilon_{t-1}, \) and \( x_{t-1} \), as well as incorporating assumption (5):

\[ \Pr(x_t, i_t, \varepsilon_t | x_{t-1}, i_{t-1}, \varepsilon_{t-1}, \theta) = \Pr(i_t | x_t, \varepsilon_t, \theta) \Pr(\varepsilon_t | i_{t-1}, \varepsilon_{t-1}, \theta) \Pr(x_t | x_{t-1}, i_{t-1}) \]  

(19)

For notational simplicity, we define

\[ m_{it} \equiv u_\theta(i, x_t) + \beta \mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t]. \]  

(20)

While \( \Pr(\varepsilon_t | i_{t-1}, \varepsilon_{t-1}, \theta) \) is determined by (7) and \( \Pr(x_t | x_{t-1}, i_{t-1}) \) is estimated independently (and therefore omitted from now on),\textsuperscript{15} the conditional decision probability \( \Pr(i_t | x_t, \varepsilon_t, \theta) \) is given by

\[ \Pr(i_t = 1 | x_t, \varepsilon_t(0), \varepsilon_t(1), \theta) = \mathbbm{1}(m_{1t} + \varepsilon_t(1) > m_{0t} + \varepsilon_t(0)) \]  

(21)

where \( \mathbbm{1}(\cdot) \) is the index function that is equal to one if its argument is true, and zero otherwise; note that the conditional decision probabilities are actually degenerate, because — loosely speaking — there is no randomness left, given \( \varepsilon_t \).

Finally, exploiting the Markov structure for the integration, and dropping parameter depen-

\textsuperscript{14}This is not to be confused with the recursive maximum likelihood estimation (RMLE) algorithm of Kay (1983) for the estimation of AR processes, which allows one to recursively update maximum likelihood estimates to higher order AR models.

\textsuperscript{15}Since one can estimate the mileage transition probabilities separately, they only add a multiplicative constant to the likelihood function of \( \theta = \{\theta_1, RC, \rho\} \). Thus, we omit the corresponding term of the likelihood function (and one should do so in the actual maximization for scaling reasons).
dence for better readability, we can write the likelihood function (17) as

\[
\int \cdots \int \prod_{t=1,\ldots,T-1} \Pr(i_t | x_t, \varepsilon_t) \Pr(\varepsilon_t | i_{t-1}, \varepsilon_{t-1}) \\
\int_{\varepsilon_T} \Pr(i_T | x_T, \varepsilon_T) \Pr(\varepsilon_T | i_{T-1}, \varepsilon_{T-1}) \, d\varepsilon_0 \cdots d\varepsilon_{T-1} d\varepsilon_T \quad (22)
\]

To numerically approximate (22), we define the function

\[
g_t(\varepsilon) = \begin{cases} 
1 & \text{if } t > T \\
\int_{\varepsilon'} \Pr(i_t | x_t, \varepsilon') \Pr(\varepsilon' | i_{t-1}, \varepsilon) g_{t+1}(\varepsilon') \, d\varepsilon' & \text{otherwise}
\end{cases}
\]

Now, given \(g_{t+1}(\varepsilon)\), we can numerically approximate the function \(g_t(\varepsilon)\) using both numerical integration and function approximation. Since \(g_t(\varepsilon)\) is known to be unity for \(t > T\), we can use backward iteration starting from \(g_T(\varepsilon)\) to solve for \(g_0(\varepsilon)\), which is the approximation of the likelihood function \(L(\theta | \cdot)\). Note that this procedure is analogous to solving for the value function of a finite horizon, discrete time dynamic programming problem by backward iteration. Algorithm 1 gives a formal description of the procedure.

Algorithm 1: Computation of the likelihood function (22) by recursive likelihood function integration (RLI).

1: \(\Gamma \leftarrow\) initialize grid over support of \(\varepsilon\) with \(D\) elements
2: \(\tilde{g}(\cdot) \leftarrow\) initialize interpolant with nodes \(\{(e, \tilde{g}_e)\}_{e \in \Gamma}\) to unity
3: for \(t = T, \ldots, 1\) do
4: \hspace{1em} for \(e \in \Gamma\) do
5: \hspace{2em} \(\tilde{g}_e \leftarrow\) approximate \(\int_{\varepsilon'} \Pr(i_t | x_t, \varepsilon') \Pr(\varepsilon' | i_{t-1}, e) \tilde{g}(\varepsilon') \, d\varepsilon'\)
6: \hspace{1em} end for
7: \hspace{1em} \(g(\cdot) \leftarrow\) construct interpolant with nodes \(\{(e, \tilde{g}_e)\}_{e \in \Gamma}\)
8: end for

Note that each integral over \(\varepsilon_t\) is generally still \(N\)-dimensional. Thus, the procedure decomposes the \(T \cdot N\)-dimensional integral of (17) to an \(N\)-dimensional integration that is repeated \(D \cdot T\) times, where \(D\) is the number of nodes used for function approximation of \(g_t\). Since the computational complexity of deterministic numerical integration is generally exponential in the number of dimensions, this reduction is highly desirable even for large \(D\), because it enters the complexity of the overall algorithm linearly.

\[O(\exp(T \cdot N)) \gg O(D \cdot T \exp(N)) \quad (24)\]

---

16 Recursive computation of the likelihood function for serially correlated unobserved Markov states is not a new idea in general. However, to the best of our knowledge, its application has been limited to discrete state spaces, and therefore with no need for numerical quadrature or function approximation; see, for example, Cosslett and Lee (1985) for the estimation of models with Markov regime switching.

17 Algorithm 1 is generic with respect to both the numerical integration scheme and the function approximation schemes, as long as the latter depend on function evaluations only. Also, it can be applied analogously to the case of discrete (or discretized) error processes.

18 In this context, the \(O(f(y))\) notation for the computational complexity of an algorithm reads as follows: There exists a constant \(K > 0\) such that the number of iterations needed for an algorithm to complete a task of size \(y\) is bounded by \(K \cdot f(y)\).
Given that serial correlation is only allowed in some dimensions, but not all, we can potentially replace parts of the integral in (23) by a closed form solution; this is particularly the case if the cumulative distribution of those unobserved state variables that are not serially correlated does have a closed form. Recall that the integration over $\varepsilon_t$ is really $N$-dimensional, thus 2-dimensional in the model under consideration:

$$
\int \int_{\varepsilon_t(0) \varepsilon_t(1)} Pr(\varepsilon_t(0) | i_{t-1}, \varepsilon_{t-1}(0)) \; Pr(\varepsilon_t(1)) \; Pr(i_t | x_t, \varepsilon_t(0), \varepsilon_t(1)) \; d\varepsilon_t(1)d\varepsilon_t(0) \quad (25)
$$

Using (21), we can write the integral over $\varepsilon_t(1)$ in terms of its cumulative distribution function $F$,

$$
\int_{-\infty}^{\infty} 1(\varepsilon_t(1) > m_{0t} - m_{1t} + \varepsilon_t(0)) \; Pr(\varepsilon_t(1)) \; d\varepsilon_t(1) = \int_{m_{0t} - m_{1t} + \varepsilon_t(0)}^{\infty} Pr(\varepsilon_t(1)) \; d\varepsilon_t(1) = 1 - F(m_{0t} - m_{1t} + \varepsilon_t(0)) \quad (26)
$$

which no longer involves numerical quadrature if an analytical formula for $F$ exists.

For the actual computations we use Gaussian quadrature as outlined in the previous section (in the context of expected value function approximation). Note that while we write all integrals in this section as integrals over $\varepsilon$ for simplicity, we have to reformulate them in terms of $\tilde{\varepsilon}$ by a linear change of variables in order to approximate them by Gaussian quadrature (see Section 3.1). Also, for numerical reasons, we chose a slightly different change of variables to map the integration domain from $[-\infty, \infty]$ to $[-1, 1]$, (see Judd, 1998, p. 204). Furthermore, we use Akima splines (Akima, 1970) to approximate the integral over $\varepsilon_t$ as a function of $\varepsilon_{t-1}$.

### 3.3 Likelihood Function Maximization

Obtaining the maximum likelihood estimate of $\theta$, given data $\{x_t, i_t\}_{t=0}^T$, requires us to find a solution to the following two problems simultaneously:

$$
\hat{\theta} = \arg \max_{\theta} L(\theta | \{x_t, i_t\}_{t=0}^T, \overline{EV}_\theta) \quad (27)
$$

$$
\overline{EV}_\theta(x, \varepsilon; a) = T(\overline{EV}_\theta)(x, \varepsilon; a) \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \quad (28)
$$

While there exist methods that directly solve (27) and (28) simultaneously as a constrained optimization problem, namely the mathematical programming with equilibrium constraints (MPEC) approach to DDCM estimation of Su and Judd (2012), we use the well known nested fixed point (NFXP) approach of Rust (1988).\footnote{The MPEC approach to DDCM estimation of Su and Judd (2012) “combines” the solution of the fixed point and the maximization of the likelihood by solving the original constraint formulation of the likelihood maximization problem (27). This procedure is considered to be more efficient in some cases, because it does not require one to solve the fixed point equation (9) for each parameter guess, even if it is far away from the solution; rather, it imposes the fixed point condition to hold only at the solution. However, directly integrating MPEC with adaptive interpolation grids creates two potential problems: First, adding a grid node corresponds to adding a constraint to the optimization problem, while the optimization algorithm runs. Second, adaptive methods usually} In NFXP, the likelihood maximization is performed as a
repeated two step procedure: First, given a parameter guess \( \theta^{(k)} \), one computes the expected value function \( EV_{\theta^{(k)}} \) as a fixed point of operator \( T \) by solving (28). Second, one evaluates the likelihood function for \( \theta^{(k)} \), using the approximation of \( EV_{\theta^{(k)}} \) just previously obtained. The optimization algorithm then constructs a new parameter guess \( \theta^{(k+1)} \), and the procedure starts again by approximating \( EV_{\theta^{(k+1)}} \). This is iterated until convergence of the maximization algorithm.\(^{20}\) Thus, (27) can be solved as an unconstrained problem.

Recall that the interpolation grid \( \Gamma_{\theta^{(k)}} \), over which the corresponding approximating interpolant \( \hat{EV}_{\theta^{(k)}}(\cdot; a) \) satisfies some error bound \( \bar{\eta} \), depends on \( \theta^{(k)} \). Thus, each step of the maximization routine, from \( \theta^{(k)} \) to \( \theta^{(k+1)} \), requires one to iteratively update the grid from \( \Gamma_{\theta^{(k)}} \) to \( \Gamma_{\theta^{(k+1)}} \), until the maximum approximation error of \( \hat{EV}_{\theta^{(k)}}(\cdot; a) \) is bounded by \( \bar{\eta} \) again; this procedure ensures that for each likelihood function evaluation, the approximation error of the corresponding expected value function is controlled.\(^{21}\)

Algorithm 2 summarizes the nested fixed point algorithm to solve (27).

**Algorithm 2** Nested fixed point algorithm with adaptive grid updating.

1. initialize \( \theta, \Gamma_{\theta}, a \)
2. while not converged do
   3. while \( \eta(1 - \beta)^{-1} > \bar{\eta} \) do
      4. solve \( \hat{EV}_{\theta}(x, \varepsilon; a) = T(\hat{EV}_{\theta})(x, \varepsilon; a) \) \( \forall (x, \varepsilon) \in \Gamma_{\theta}, a \in \mathbb{R}^A \)
      5. update \( \Gamma_{\theta} \) (coarsening and refinement)
   6. end while
3. evaluate \( L(\theta) \)
4. compute next \( \theta \)
5. end while

For the model under consideration, the maximization of the likelihood function is a non-linear, partially box-constrained optimization problem with three free parameters. To numerically solve this problem, we employ the model-based, derivative-free trust-region method “bobyqa” (Powell, 2009).\(^{22}\)

### 4 Estimation Results

The original dataset of Rust (1987) consists of monthly odometer readings and engine replacement decisions for a fleet of 162 buses, subdivided into 8 groups depending on their manufacturer and model. Since buses are heterogeneous across groups, it is common to create different subsamples to estimate the parameters of model (1); we follow the literature by estimating three require the approximation of an iteration to be completed in order to compute the approximation quality for the insertion decision, which in our case is not possible until (9) has been solved, which in turn contradicts the MPEC idea.

\(^{20}\)Since the fixed point of \( T \) is usually obtained using an iterative method, solving the dynamic problem is often referred to as the “inner loop” in this context, while the maximization procedure is referred to as the “outer loop”.

\(^{21}\)Controlling the maximum approximation error does not imply that it is constant over the maximization procedure. Rather, we choose \( \bar{\eta}^{(k)} \) to be decreasing in the iterations of the optimizer, in order to compute the fixed point to lower accuracy far away from the solution, but to high accuracy close to it.

\(^{22}\)According to Powell (2009), the name “bobyqa” is an acronym for “Bound Optimization BY Quadratic Approximation.”
subsamples separately, consisting of groups \{1, 2, 3\}, \{1, 2, 3, 4\}, and \{4\}. Table 1 shows the size of the panel for each group under consideration.

<table>
<thead>
<tr>
<th>Bus group</th>
<th>Number of buses (M)</th>
<th>Observation horizon (months)</th>
<th>Total number of observations</th>
<th>Number of replacements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>25</td>
<td>360</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>49</td>
<td>192</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>70</td>
<td>3,312</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>37</td>
<td>117</td>
<td>4,292</td>
<td>33</td>
</tr>
<tr>
<td>Total</td>
<td>104</td>
<td>8,156</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Number of buses, observation time horizon in months, total number of observations, and number of observed engine replacements for each bus group.

As in Rust (1987), we discretize mileage in “bins” of 5,000 miles each.\(^{23}\) The highest possible mileage state is 90 (which corresponds to 450,000 miles),\(^{24}\) formally \(x \in X = \{1, \ldots, 90\}\). We assume the mileage transition to follow a Markov process (conditional on the replacement decision), for which we estimate the parameters independently. We parametrize the discount factor by \(\beta = 0.9999\) as in the original paper.

Before presenting the results of the estimation, we verify the estimation procedure presented in Section 3: First, Table 2 presents a partial reproduction of Table IX of Rust (1987), without serial correlation, but still numerically integrating both the expected value and the likelihood function. We conclude that, for the case without serial correlation, we are well able to replicate the original estimates.

<table>
<thead>
<tr>
<th>Bus groups 1–3</th>
<th>Bus groups 1–4</th>
<th>Bus group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>11.7270</td>
<td>9.7558</td>
</tr>
<tr>
<td></td>
<td>(2.602)</td>
<td>(1.227)</td>
</tr>
<tr>
<td>(\theta_1)</td>
<td>4.8259</td>
<td>2.6275</td>
</tr>
<tr>
<td></td>
<td>(1.792)</td>
<td>(0.618)</td>
</tr>
<tr>
<td>(\rho)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L)</td>
<td>-2,708.366</td>
<td>-6,055.250</td>
</tr>
</tbody>
</table>

Table 2: Replication of Table IX of Rust (1987) for all subsamples reported therein; \(L\) is the value of the log-likelihood function at the solution; \(\beta = .9999\).

Second, we carry out an extensive Monte Carlo study, where we simulate the model from Section 2 to create many data sets of different sizes (number of buses),\(^{25}\) for both densities,

\[^{23}\text{By discretizing into bins of 5,000 miles we mean that the original mileage } \tilde{x} \text{ transforms into a mileage state } x = \lceil \tilde{x}/5,000 \rceil, \text{ with the ceiling function } \lceil \tilde{y} \rceil = \min\{y \in \mathbb{N} : y \geq \tilde{y} \}.
\[^{24}\text{If a bus ever reaches the maximum mileage state, we assume it to stay there until engine replacement. Although no bus in any of our subsamples ever reaches the maximum mileage state, it still has relevance for the solution of the dynamic problem of the agent, who takes this possibility into account when solving his infinite horizon dynamic optimization problem.}
\[^{25}\text{Note that we refer to data set size as the number of buses in a dataset, or, equivalently, as the number of replacement observations, as we simulate each bus until replacement.}
extreme value type 1 EV1 and standard normal $N(0,1)$, and estimate the parameters from these data sets using NFXP together with the RLI algorithm. The objective is to investigate the ability of the method to recover the parameters from the data, for which we know the true values in the case of simulated data. Therefore, for each data set size $M \in \{100, 1000, 10000\}$, and for both densities, we create 200 datasets; on each data set, we run an estimation with and without allowing for serial correlation (i.e. setting $\rho = 0$). Table 3 presents the results of this Monte Carlo study by reporting means and standard deviations of the respective estimates. We also report mean and standard deviation of the likelihood ratio test with the null hypothesis of absence of serial correlation, carried out on the individual data set level. Figures 3 and 4 finally plot a kernel smoothing estimation of the distribution of the estimates, together with the true parameter values, and the density of the normal distribution with mean and standard deviation as reported in Table 3.

From this Monte Carlo study we draw the following conclusions: First, while the method seems to slightly overestimate both cost parameters, the true parameters are always well within one standard deviation. Also, in case of EV1 distributed $\tilde{\varepsilon}$ where the overestimation is most apparent, the mean of the estimates clearly gets closer to the true values as we increase the data set size. For the serial correlation parameter $\rho$, we observe almost perfect recovering of the true parameter value for large data sets, in the EV1 case even for moderate data set sizes. Comparing the estimates with serial correlation to the case where serial correlation is ruled out by setting $\rho = 0$, we see that the parameter estimates vary considerably. However, looking at the (probably more relevant) ratio of the cost parameters, we find that the misspecification bias is small in the EV1 case, but moderate in the $N(0,1)$ case. Testing for the statistical significance of the increase in quality of fit by allowing for serially correlated errors using the likelihood ratio test, we find that given a data set of 100 buses (which is comparable to the largest subset of the original data presented above), it is often impossible to reject the no-serial-correlation hypothesis at a reasonable significance level, even if the true model features serial correlation as in (7). While in the EV1 case, significance increases vastly for the larger data sets under consideration, the model with normal $\tilde{\varepsilon}$ has a surprisingly low increase in quality of fit, along with relatively large $p$-values even for large data sets.

Turning our attention to Figures 3 and 4, we notice that for the smaller data sets the distribution of the parameter estimates is clearly not normal. Moreover, it even appears to be bimodal, with one solution being the no serial correlation case. However, since for the large data sets, the distributions apparently become closer to the density of the normal distribution, especially for the cost parameters in the $N(0,1)$ case, and for the serial correlation parameter in the EV1 case, it appears that the estimators might actually be asymptotically normally distributed.

At this point it is worthwhile commenting on the sources and potential impact of numerical truncation error: First, we found that the likelihood function is very flat on the right tail of the cost parameter distribution; keeping in mind that the stopping criterion of an optimization algorithm introduces a truncation error of its own, this could well explain the local modes on these tails. Second, while we use a specific Gaussian rule for the normally distributed $\tilde{\varepsilon}$, namely

---

26 The values for the parameters are chosen such that they resemble the estimates for the largest subset of the original dataset for the respective distribution, as reported below.
Figure 3: Distributions of the maximum likelihood estimates from 200 artificial data sets of different sizes, with density \( q(\cdot) \) being extreme value type 1 EV1. The bold solid vertical lines denote the true parameter value; the thin solid lines are kernel smoothing estimates of the distributions of the parameter estimates; the dash-dotted lines depict normal distributions with mean and standard deviation of the respective estimates. The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses.
Figure 4: Distributions of the maximum likelihood estimates from 200 artificial data sets of different sizes, with density $q(\cdot)$ being standard normal $N(0, 1)$. The bold solid vertical lines denote the true parameter value; the thin solid lines are kernel smoothing estimates of the distributions of the parameter estimates; the dash-dotted lines depict normal distributions with mean and standard deviation of the respective estimates. The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses.
Table 3: Mean and standard deviations of the maximum likelihood estimates from 200 artificial data sets of different sizes, density \(q(\cdot)\) being extreme value type 1 \(EV_1\) (top), and standard normal \(N(0,1)\) (bottom). The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses. \(L\) is the value of the log-likelihood function at the solution; \(p\) (LR) is the mean and the standard deviation of the \(p\)-values of the likelihood ratio test with \(H_0: \beta = .9999\).

Table 4 finally presents the estimation results using the original dataset of Rust (1987), again for both \(EV_1\) and normally distributed \(\tilde{\epsilon}\). As for the artificial data, we observe that in the \(EV_1\) case, while the parameter estimates in the presence of serial correlation are substantially different
from the estimates without serial correlation, the ratio of engine replacement cost to the regular maintenance cost parameter is relatively stable; thus, the trade-off for the decision maker has not changed much quantitatively. Performing a likelihood ratio test to compute the statistical significance of the quantitative changes induced by the introduction of serial correlation, we find that only on the largest subsample of the dataset (bus groups 1–4) can we reject the hypothesis of no serial correlation at a reasonable significance level. Also, comparing the parameter estimates from the unrestricted model to their counterparts from the restricted model individually, we observe that the difference for the cost parameters is roughly within the standard error; only the serial correlation parameter itself deviates by more than two standard errors. The case of normally distributed \( \tilde{\varepsilon}(i) \) yields similar results, with two notable differences: First, not only do the cost parameter values change substantially, but also their ratios and thus the trade-off for the decision maker. This is because the variance of the error is no longer normalized, but rather depends positively on \( \rho \), which also tends to cause identification issues. Second, while all parameter estimates from the largest subsample (bus groups 1–4) in the unrestricted case deviate by more than one standard error from their restricted counterparts, the estimates from the smaller subsamples are within the standard error even for the serial correlation parameter. Carrying out a likelihood ratio test, we cannot reject the hypothesis of no serial correlation at a reasonable significance level for any of the subsamples in the normal case.

We interpret the change of the ratio of the cost parameters in this particular model as follows (as an example, we assume the ratio in the restricted model to be larger than in the unrestricted one): If we ignore serial correlation, the relative costs of regular maintenance are underestimated. Consequently, using the true relative costs in a model without serial correlation, we would predict more (or, equivalently, earlier) engine replacement than we find in the data. Thus, allowing for serial correlation explains why we do not observe more frequent engine replacement, given the high (true) relative costs of regular maintenance. Conversely, in a model with serial correlation, but based on the biased relative costs estimates, we would predict the buses to run for too long without engine replacement.

Assessing the question of the statistical significance of the estimates from the original data set is difficult though. First, from our experiments with artificial data sets we learned that the results are rarely significant for small samples, even if the true model features serial correlation as defined by (7). Consequently, given the number of buses in the original data set, significance as for groups 1–4 with extreme value distributed \( \tilde{\varepsilon}(i) \) is not what we can generally expect. Second, we still cannot conclude that the serial correlation we found in the data is really coming from an unobserved source, as different bus groups are pooled together for two of the three subsamples, thus creating a heterogeneous sample that is treated as homogeneous by the model. Consequently, as long as we do not find the serial correlation within one single bus group to be significant, these estimations have to be taken with a grain of salt.

\[^{27}\text{For the estimation of the standard errors from the original data set, we use the inverse of the negative Hessian of the likelihood function at its maximum, } (−H(\tilde{\theta} | \{x_{i}, i\}_i^T))^{-1}, \text{ which is approximated using finite differences.}\]
Table 4: Estimation results for different subsamples of the original dataset, density \( q(\cdot) \) being extreme value type 1 \( EV1 \) (top), and standard normal \( N(0, 1) \) (bottom). \( L \) is the value of the log-likelihood function at the solution; \( p \) (LR) is the \( p \)-value of the likelihood ratio test with \( H_0 : \rho = 0; \beta = .9999. \)

5 Conclusion

This paper developed a method to efficiently estimate dynamic discrete choice models in the presence of serial correlation in the unobserved state variables. First, to approximate the expected value function of the underlying dynamic problem, we use Gaussian quadrature and interpolation over an adaptively refined grid, and solve a potentially large non-linear system. Second, to evaluate the likelihood function, we decompose the integral over the unobserved state variables in the likelihood function into a series of lower dimensional integrals, and successively approximate them using Gaussian quadrature rules; we call this procedure recursive likelihood function integration (RLI). Finally, we solve the maximum likelihood problem using a nested fixed point algorithm.

First, we verify the RLI algorithm’s ability to recover the parameters in an extensive Monte Carlo study with simulated data sets, finding that the method is indeed able to recover the parameters used for the simulation, particularly in the case of the serial correlation parameter, which is recovered to very high precision. Also, we find some evidence that the distribution of the estimates is asymptotically normal for big enough data sets. Then, we apply this method to the bus engine replacement model of Rust (1987), and find significant serial correlation for
some of the subsamples. Also, the parameter estimates vary substantially, compared to the case of serially uncorrelated errors. We wish to emphasize again that the method presented in this paper is not limited to the bus engine replacement model, but is generic with respect to DDCMs with Markov serial dependence in the unobserved state variables.

As we mention in the introductory section, the recursive likelihood function integration is not the only approach to the estimation of DDCMs with serially correlated unobserved state variables. While we cited some recent alternative methods, we did not compare to them in terms of runtimes, accuracy, or other important metrics. Rather, the goal of this paper was to show that the integration of the serially correlated variables in the computation of the likelihood function can be done with complexity that is linear in the time horizon, making the application of high performance quadrature rules such as Gaussian quadrature well feasible. For a quantitative comparison of the various methods to be insightful, a rigorous experimental design is needed, in order to compare the different aspects of computational efficiency, numerical accuracy, and scaling properties, based on unified models and environments. This in-depth comparison study is subject to future research.
A Open Source Software

This appendix lists all open source software packages used to obtain the results presented in this paper, including version information.

The main framework used to implement the method of this paper is R, version 3.0.3 (R Core Team, 2014). Time critical components are implemented in C++, and interfaced to R using the “Rcpp” package, v0.11.1 (Eddelbüttel and François, 2011). The code is parallelized on the C++ level using openMP. All interpolation on the C++ level is carried out using the respective routines of the GNU Scientific Library, v1.16 (Galassi et al., 2014). Gaussian quadrature nodes are computed using the R-packages “fastGHQuad”, v0.1-1 (Blocker, 2011), and “pracma”, v1.6.4 (Borchers, 2014). Distribution functions, quantile functions, and random number generators for the extreme value distribution are provided by the R-package “evd”, v2.3-0 (Stephenson, 2002). To numerically solve the fixed point problem (9), we use the “ipopt” package, v3.11.7 (Wächter and Biegler, 2005), in conjunction with the “pardiso” sparse linear solver, v5.0.0 (Schenk and Gärtner, 2004), interfaced by the R-package “ipoptr”, v0.8.4, by Jelmer Ypma (which is distributed as part of the ipopt package), and the quasi-Newton trust-region method of the R-package “nleqslv”, v2.1.1 (Hasselman, 2014). For the likelihood maximization problem, we employ “bobyqa” (Powell, 2009), interfaced by the “minqa” R-package, v1.2.3 (Bates et al., 2012).
References


namic Discrete Choice Models with Unobserved Heterogeneity. *Econometrica: Journal of the 

Optimization Algorithms by Quadratic Approximation*. R package version 1.2.3.


R package version 0.1-1.


Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, 

*Numerische Mathematik*, 75(3):319–337.

for Optimal Control Problems in Economics. *Journal of Economic Dynamics and Control*, 
28(12):2427–2456.


