NBER WORKING PAPER SERIES

IMPROVING THE NUMERICAL PERFORMANCE OF BLP STATIC AND DYNAMIC DISCRETE CHOICE RANDOM COEFFICIENTS DEMAND ESTIMATION

Jean-Pierre H. Dubé
Jeremy T. Fox
Che-Lin Su

Working Paper 14991
http://www.nber.org/papers/w14991

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
May 2009

We thank Daniel Ackerberg, Steven Berry, John Birge, Amit Gandhi, Philip Haile, Lars Hansen, Panle Jia, Kyoo il Kim, Samuel Kortum, Kenneth Judd, Sven Leyffer, Denis Nekipelov, Aviv Nevo, Jorge Nocedal, Ariel Pakes, John Rust, Hugo Salgado, Azeem Shaikh and Richard Waltz for helpful discussions and comments. We also thank workshop participants at CREST-INSEE / ENSAE, EARIE, the ESRC Econometrics Study Group Conference, the Econometric Society, the Federal Trade Commission, INFORMS, the International Industrial Organization Conference, the 2009 NBER winter IO meetings, Northwestern University, the Portuguese Competition Commission, the Stanford Institute for Theoretical Economics, the UK Competition Commission, the University of Chicago, and the University of Rochester. Dubé is grateful to the Kilts Center for Marketing and the Neubauer Faculty Fund for research support. Fox thanks the NSF, grant 0721036, the Olin Foundation, and the Stigler Center for financial support. Su is grateful for the financial support from the NSF (award no. SES-0631622) and the University of Chicago Booth School of Business. The views expressed herein are those of the author(s) and do not necessarily reflect the views of the National Bureau of Economic Research.

Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation

Jean-Pierre H. Dubé, Jeremy T. Fox, and Che-Lin Su

NBER Working Paper No. 14991
May 2009
JEL No. C01,C61,L0

ABSTRACT

The widely-used estimator of Berry, Levinsohn and Pakes (1995) produces estimates of consumer preferences from a discrete-choice demand model with random coefficients, market-level demand shocks and endogenous prices. We derive numerical theory results characterizing the properties of the nested fixed point algorithm used to evaluate the objective function of BLP's estimator. We discuss problems with typical implementations, including cases that can lead to incorrect parameter estimates. As a solution, we recast estimation as a mathematical program with equilibrium constraints, which can be faster and which avoids the numerical issues associated with nested inner loops. The advantages are even more pronounced for forward-looking demand models where Bellman's equation must also be solved repeatedly. Several Monte Carlo and real-data experiments support our numerical concerns about the nested fixed point approach and the advantages of constrained optimization.

Jean-Pierre H. Dubé
University of Chicago
Booth School of Business
5807 South Woodlawn Avenue
Chicago, IL 60637
and NBER
jdube@gsb.uchicago.edu

Jeremy T. Fox
Department of Economics
University of Chicago
1126 East 59th Street
Chicago, IL 60637
and NBER
fox@uchicago.edu

Che-Lin Su
University of Chicago
che-lin.su@chicagogsb.edu

1 Introduction

Discrete-choice demand models have become popular in the demand estimation literature due to their ability to accommodate rich substitution patterns between a large array of products. Berry, Levinsohn and Pakes (1995), hereafter BLP, made an important contribution to this literature by accommodating controls for the endogeneity of product characteristics (namely prices) without sacrificing the flexibility of these substitution patterns. Their methodological contribution comprises a statistical, generalized method-of-moments (GMM) estimator and a numerical algorithm. However, many implementations of the BLP algorithm may produce incorrect parameter estimates because they are vulnerable to numerical inaccuracy. We study the underlying numerical theory leading to some of these computational problems and propose an alternative algorithm that is robust to these sources of numerical inaccuracy.

As in Berry (1994), the evaluation of the GMM objective function requires inverting the nonlinear system of market share equations. BLP and Berry suggest nesting this inversion step directly into the parameter search. BLP propose a contraction mapping algorithm to solve this system of equations numerically. Following the publication of Nevo's (2000b) "A Practitioner's Guide" to implementing BLP, numerous studies have emerged using BLP's algorithm for estimating discrete choice demand systems with random coefficients.

Our first objective consists of exploring the numerical properties of BLP's nested contraction mapping algorithm. We refer to this approach as the nested fixed point, or NFP, approach.[1] The GMM objective function and the corresponding nested inner loop may be called hundreds or thousands of times during a numerical optimization over structural parameters. Therefore, it may be tempting to use a less stringent stopping criterion for the inner loop to speed up estimation. We derive theoretical results to show how a less stringent stopping criterion for the inner loop may cause two types of errors in the parameter estimates. First, the inner loop error propagates into the outer loop GMM objective function and its derivatives, which may cause an optimization routine to fail to converge. Second, even when an optimization run converges, it may falsely stop at a point that is not a local minimum. To illustrate these problems empirically, we construct examples based on pseudo-real data and data generated from a known model. In these examples, the errors in the parameter estimates from using loose tolerances for the NFP inner loop are large.

Our second objective is to propose a new computational algorithm for implementing the BLP estimator that eliminates the inner loop entirely and, thus, removes the potential for numerical inaccuracy discussed above. Following Su and Judd (2008), we recast the optimization of BLP's GMM objective function as a mathematical program with equilibrium constraints (MPEC). The MPEC method minimizes the GMM objective function subject to a system of

[1] We borrow the term "nested fixed point" from the work of Rust (1987) on estimation of dynamic programming models, without demand shocks.

nonlinear constraints requiring that the predicted shares from the model equal the observed shares in the data. Because both the GMM objective function and the market share equations are smooth, we solve a standard constrained optimization problem.

We prefer the MPEC approach over the existing NFP approach for four reasons. First, there is no nested inner loop and, hence, no numerical error from nested calls. This aspect eliminates the potential for an optimization routine to converge to a point that is not a local minimum of the true GMM objective function.[2] Second, by eliminating the nested calls, the procedure could be faster than the contraction mapping method proposed by BLP. Third, the MPEC algorithm allows the user to relegate all the numerical operations to a call to a state-of-the-art optimization package. The speed advantage of MPEC may be enhanced further if a Newton-based solver with a quadratic rate of convergence is used.[3] In contrast, the convergence rate of the entire NFP approach has not been established in the literature. Fourth, the MPEC approach applies to more general demand models: those where there is no contraction mapping (Gandhi 2008).

We emphasize the distinction between the estimation problems associated with poorly-implemented numerical optimization (our focus) and problems of poor identification. From a statistical perspective, the MPEC algorithm generates the same estimator as the correctly-implemented NFP approach. Conceptually, MPEC is just an alternative formulation for implementing BLP's original estimator. Therefore, the theoretical results on consistency and statistical inference in Berry, Linton and Pakes (2004) apply equally to NFP and MPEC.

To illustrate the relative performance of MPEC and NFP, we conduct a series of sampling experiments. Our first set of experiments documents the estimation problems that arise when NFP is implemented poorly. In three examples, we document cases where a loose tolerance for the contraction mapping in the NFP approach leads to incorrect parameter estimates and/or the failure of an optimization routine to report convergence. We observe this problem with optimization routines that use closed-form and numerical derivatives. The errors in the estimated own-price elasticities are also found to be large in both pseudo-real field data and in simulated data. Further, in one example we show that the parameter estimates always produce the same incorrect point (a point that is not even a local minimum), so that using multiple starting points may not be able to diagnose the presence of errors in the parameter

[2]Petrin and Train's (2008) control-function approach also avoids the inner loop by utilizing additional non-primitive assumptions relating equilibrium prices to the demand shocks. Although MPEC is computationally more intensive than the control-function approach, it nevertheless avoids the need for numerical inversion while retaining the statistical properties of BLP's original GMM estimator.

[3]Alternative methods to a contraction mapping for solving systems of nonlinear equations with faster rates of convergence typically have other limitations. For instance, the traditional Newton's method is only guaranteed to converge if the starting values are close to a solution, unless one includes a line-search or a trust-region procedure subject to some technical assumptions. In general, most practitioners would be daunted by the task of nesting a hybrid Newton method customized to a specific demand problem inside the outer optimization over structural parameters.

estimates. We also use this example to show that an alternative Nelder-Meade or simplex algorithm, which does not use gradient information, also usually converges to the wrong solution.

In a second set of sampling experiments, we explore the relative speed of the NFP approach (correctly implemented) and MPEC. We use numerical theory to show that the rate of convergence of the NFP's inner loop contraction mapping is bounded above by a function that is linear in the Lipschitz constant of the contraction mapping. We derive an analytic expression for the Lipschitz constant that depends on the data and the parameter values from the demand model. We construct several Monte Carlo experiments in which MPEC becomes several times faster than NFP when we manipulate the data in ways that increase the Lipschitz constant. As expected, MPEC's speed is relatively invariant to the value of the Lipschitz constant because the contraction mapping is not used. Some may be concerned that MPEC's speed scales poorly with the number of parameters in the estimation problem. For instance, adding more markets increases the number of demand shocks to be estimated as well as the number of constraints to be satisfied. We conduct additional sampling experiments in which we find that the relative speed advantage of MPEC over NFP is robust to substantial increases in the number of markets. Finally, we show that the relative performance of MPEC versus NFP is robust to the "data quality" in a series of sampling experiments that manipulate the power of the instruments.

The theoretical results we derive can be generalized well beyond the standard static BLP model. In particular, the numerical problems associated with NFP will be magnified in dynamic discrete-choice demand models with forward-looking consumers. Consider the recent empirical literature on durable and semi-durable goods markets, where consumers can alter the timing of their purchase decision based on expectations about future products and prices (Carranza 2008, Gowrisankaran and Rysman 2007, Hendel and Nevo 2007, Melnikov 2002, and Nair 2007). Estimating demand using the NFP algorithm now involves three numerical loops: the outer optimization routine, the inner inversion of the market share equations, and the inner evaluation of the consumers' value functions (the Bellman equations) for each of the several heterogeneous consumer types. The consumer's dynamic programming problem is typically solved with a contraction mapping with a slow rate of convergence. Furthermore, Gowrisankaran and Rysman point out that the recursion proposed by BLP may no longer be a contraction mapping for some specifications of dynamic discrete choice models. Hence, the market share inversion is not guaranteed to converge to a solution, which, in turn, implies that one may not evaluate the GMM objective function correctly.

MPEC extends naturally to the case with forward-looking consumers. We optimize the statistical objective function and impose consumers' Bellman equations and market share equations as constraints. Our approach eliminates both inner loops, thereby eliminating these two sources of numerical error when evaluating the outer loop objective function. We

produce benchmark results that show that MPEC can be faster than NFP under realistic data generating processes. More importantly, we sometimes find that NFP fails to report convergence to a local optimum, whereas MPEC routinely converges. In practice, when the researcher does not know the true parameters, it may be difficult to assess the validity of estimates from a non-converged run. We expect the relative performance of MPEC to improve for more complex dynamic demand models that nest more calls to inner loops (Lee 2008 and Schiraldi 2008).

Knittel and Metaxoglou (2008) explore the potential multiplicity of local minima of BLP's GMM objective function. To make sure our estimates are reliable, we employ multiple starting points in each run of our Monte Carlo experiments and routinely find that both NFP and MPEC recover the true structural parameters using data generated by the model. We also examine the same pseudo-real dataset used by Knittel and Metaxoglou. Using a state-of-the-art solver with 50 starting points in our implementation, we find the same local minimum each time. We briefly comment that some non-gradient-based solvers may report estimates that are not local minima.

The remainder of the paper is organized as follows. We discuss BLP's model in Section 2 and their statistical estimator in Section 3. Section 4 provides a theoretical analysis of the NFP algorithm. Section 5 presents our alternative MPEC algorithm. Section 6 presents examples of practices leading to errors in the estimates of parameters. Section 7 provides Monte Carlo evidence for the relative performances of the NFP and MPEC algorithms. Section 8 discusses the extension to the dynamic analog of BLP, where MPEC's advantages over NFP are magnified. We conclude in Section 9.

2 The Demand Model

In this section, we present the standard random coefficients, discrete choice model of aggregate demand. Consider a set of markets, $t = 1, ..., T$, each populated by a mass M_t of consumers who each choose one of the $j = 1, ..., J$ products available, or opt not to purchase. Each product j is described by its characteristics $(x_{j,t}, \xi_{j,t}, p_{j,t})$. The vector $x_{j,t}$ consists of K product attributes. Let x_t be the collection of the vectors $x_{j,t}$ for all J products. The scalar $\xi_{j,t}$ is a vertical characteristic that is observed by the consumers and firms, but is unobserved by the researcher. $\xi_{j,t}$ can be seen as a market- and product-specific demand shock that is common across all consumers in the market. For each market, we define the J-vector $\xi_t = (\xi_{1,t}, ..., \xi_{J,t})$. Finally, we denote the price of product j by $p_{j,t}$ and the vector of the J prices by p_t.

Consumer i in market t obtains the utility from purchasing product j

$$u_{i,j,t} = \beta_i^0 + x'_{j,t}\beta_i^x - \beta_i^p p_{j,t} + \xi_{j,t} + \varepsilon_{i,j,t}. \tag{1}$$

The utility of the outside good, the "no-purchase" option, is $u_{i,0,t} = \varepsilon_{i,0,t}$. The parameter vector β_i^x contains the consumer's tastes for the K characteristics and the parameter β_i^p reflects the marginal utility of income, $i's$ "price sensitivity". The intercept β_i^0 captures the value of purchasing an inside good instead of the outside good. Finally, $\varepsilon_{i,j,t}$ is an additional idiosyncratic product-specific shock. Let $\varepsilon_{i,t}$ be the vector of all $J+1$ product-specific shocks for consumer i.

Each consumer is assumed to pick the product j that gives her the highest utility. If tastes, $\beta_i = \left(\beta_i^0, \beta_i^x, \beta_i^p\right)$ and $\varepsilon_{i,t}$, are independent draws from the distributions $F_\beta(\beta;\theta)$, characterized by the parameters θ, and $F_\varepsilon(\varepsilon)$, respectively, the market share of product j is

$$s_j(x_t, p_t, \xi_t; \theta) = \int_{\left\{\beta_i, \varepsilon_{i,t} | u_{i,j,t} \geq u_{i,j',t} \, \forall \, j' \neq j\right\}} dF_\beta(\beta;\theta) \, dF_\varepsilon(\varepsilon).$$

To simplify aggregate demand estimation, we follow the convention in the literature and assume ε is distributed type I extreme value so that we can integrate it out analytically,

$$s_j(x_t, p_t, \xi_t; \theta) = \int_\beta \frac{\exp\left(\beta^0 + x_{j,t}'\beta^x - \beta^p p_{j,t} + \xi_{j,t}\right)}{1 + \sum_{k=1}^J \exp\left(\beta^0 + x_{k,t}'\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)} dF_\beta(\beta;\theta). \tag{2}$$

This assumption gives rise to the random coefficients logit model.

The empirical goal is to estimate the parameters θ characterizing the distribution of consumer random coefficients, $F_\beta(\beta;\theta)$. For practicality, BLP assume that $F_\beta(\beta;\theta)$ is the product of K independent normals, with $\theta = (\mu, \sigma)$, the vectors of means and standard deviations for each component of the K normals. However, several papers have studied the non-parametric identification of the model (Bajari, Fox, Kim and Ryan 2009, Berry and Haile 2008, Berry and Haile 2009, Chiappori and Komunjer 2009, and Fox and Gandhi 2009). If a parametric assumption is made about $F_\beta(\beta;\theta)$, the integrals in (2) are typically evaluated by Monte Carlo simulation. The idea is to generate n_s draws of β from the distribution $F_\beta(\beta;\theta)$ and to simulate the integrals as

$$\hat{s}_j(x_t, p_t, \xi_t; \theta) = \frac{1}{n_s} \sum_{r=1}^{n_s} \frac{\exp\left(\beta^{0,r} + x_{j,t}'\beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t}\right)}{1 + \sum_{k=1}^J \exp\left(\beta^{0,r} + x_{k,t}'\beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t}\right)}. \tag{3}$$

In principle, many other numerical methods could be used to evaluate the market-share integrals (Judd 1998, Chapters 7–9).

While a discrete-choice model with heterogeneous preferences dates back at least to Hausman and Wise (1978), the inclusion of the aggregate demand shock, $\xi_{j,t}$, was introduced by Berry (1994) and BLP. The demand shock $\xi_{j,t}$ is the natural generalization of demand shocks in the textbook linear supply and demand model. We can see in (2) that without the shock,

$\xi_{j,t} = 0 \, \forall \, j$, market shares are deterministic functions of the x's and p's. In consumer-level data applications, the econometric uncertainty is typically assumed to arise from randomness in consumer tastes, ε. This randomness washes out in a model that aggregates over a sufficiently large number of consumer choices (here a continuum). A model without market-level demand shocks will not be able to fit data on market shares across markets, as the model does not give full support to the data.

3 The BLP GMM Estimator

We now briefly discuss the GMM estimator typically used to estimate the vector of structural parameters, θ. Like the textbook supply and demand model, the demand shocks, $\xi_{j,t}$, force the researcher to deal with the potential simultaneous determination of price and quantity. To the extent that firms observe $\xi_{j,t}$ and condition on it when they set their prices, the resulting correlation between $p_{j,t}$ and $\xi_{j,t}$ introduces endogeneity bias into the estimates of θ.

BLP address the endogeneity of prices with a vector of D instrumental variables that do not appear in demand, $z_{j,t}$. They propose a GMM estimator based on the conditional moment condition $E\left[\xi_{j,t} \mid z_{j,t}, x_{j,t}\right] = 0$. The instruments $z_{j,t}$ can be product-specific cost shifters, although frequently other instruments are used because of data availability. Here the K non-price characteristics in $x_{j,t}$ are also assumed to be mean independent of $\xi_{j,t}$ and hence to be valid instruments, although this is not a requirement of the statistical theory. The estimator does not impose a parametric distributional assumption on the demand shocks $\xi_{j,t}$, as the identifying assumption is only $E\left[\xi_{j,t} \mid z_{j,t}, x_{j,t}\right] = 0$. Computationally, the researcher often implements the moments as $E\left[\xi_{j,t} \cdot h\left(z_{j,t}, x_{j,t}\right)\right] = 0$ for some known, vector-valued function h that gives C moment conditions. To summarize, the researcher's data consist of $\left\{\left(x_{j,t}, p_{j,t}, s_{j,t}, z_{j,t}\right)_{j=1}^{J}\right\}_{t=1}^{T}$ for J products in each of T markets.

To form the empirical analog of $E\left[\xi_{j,t} \cdot h\left(z_{j,t}, x_{j,t}\right)\right]$, the researcher needs to find the implied values of the demand shocks, $\xi_{j,t}$, corresponding to a guess for θ. The system of market shares, (2), defines a mapping between the vector of demand shocks and the market shares: $S_t = s\left(x_t, p_t, \xi_t; \theta\right)$, or $S_t = s\left(\xi_t; \theta\right)$ for short. Berry (1994), Berry and Pakes (2007), and Gandhi (2008) prove that s has an inverse, s^{-1}, such that any observed vector of shares can be explained by a unique vector $\xi_t\left(\theta\right) = s^{-1}\left(S_t; \theta\right)$. An individual demand shock $\xi_{j,t}$ given by this vector is $s_j^{-1}\left(S_t; \theta\right)$. For the random coefficients logit specification, we can compute ξ_t using the contraction mapping proposed in BLP.

A GMM estimator can now be constructed by using the empirical analog of the C moment conditions,

$$g\left(\xi\left(\theta\right)\right) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\xi_{j,t}\left(\theta\right) \cdot h\left(z_{j,t}, x_{j,t}\right) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}s_j^{-1}\left(S_t; \theta\right) \cdot h\left(z_{j,t}, x_{j,t}\right).$$

Let S be the vector of observed market shares in all markets and let $s^{-1}(S;\theta)$ be the implied demand shocks in all markets. For some weighting matrix, W, we define the GMM estimator as the vector, $\hat{\theta}$, that solves the problem

$$\min_{\theta} Q(\theta) = \min_{\theta} g\left(s^{-1}(S;\theta)\right)' W g\left(s^{-1}(S;\theta)\right). \tag{4}$$

The statistical efficiency of the GMM estimator can be improved by using more functions of $z_{j,t}$ in the vector of moments, finding more instruments, using an optimal weighting matrix in a second step, or using an efficient one-step method such as continuously-updated GMM or empirical likelihood. However, as we show in the following sections, the numerical accuracy of the algorithms used to compute $Q(\theta)$ is equally important, from a practical perspective, as matters of statistical efficiency.

4 A Theoretical Analysis of the NFP Algorithm

In this section, we analyze the numerical properties of BLP's NFP algorithm. The GMM estimator described in Section 3 consists of an outer loop to minimize the objective function, $Q(\theta)$, and an inner loop to evaluate this function by inverting the system of market shares using a contraction mapping. Therefore, each evaluation of the GMM objective function, $Q(\theta)$, nests a call to a fixed-point calculation.

From a practical perspective, the speed of the NFP algorithm is determined by the number of calls to evaluate the objective function and the computation time associated with the inner loop for each function evaluation. In the subsections below, we first provide formal results on the speed of convergence of the inner loop. We then show how numerical error from the inner loop can propagate into the outer loop, potentially leading to incorrect parameter estimates.

4.1 The Rate of Convergence for the NFP Contraction Mapping

The evaluation of the GMM objective function, $Q(\theta)$, requires us to compute the inverse: $\xi_t(\theta) = s^{-1}(S_t;\theta)$. For a given θ, the inner loop of the NFP algorithm solves the share equations for the demand shocks ξ by iterating the contraction mapping

$$\xi_t^{h+1} = \xi_t^h + \log S_t - \log s\left(\xi_t^h;\theta\right), \quad t = 1,\ldots,T, \tag{5}$$

until the successive iterates ξ_t^{h+1} and ξ_t^h are sufficiently close.[4] Formally, we choose a small number, for example 10^{-8} or 10^{-14}, for ϵ_{in} as the inner loop tolerance level and require ξ_t^{h+1}

[4] In our implementation of NFP, we iterate over $\exp(\xi)$ to speed up the computation because taking logarithms is slow. However, depending on the magnitude of ξ, the use of the exponentiated form $\exp(\xi)$ in a contraction mapping can lose 3 to 5 digits of accuracy in ξ, and as a result, introduce an additional source of numerical error. For example, if $\left|\xi_t^h\right| = -8$ and $\left|\exp\left(\xi_t^h\right) - \exp\left(\xi_t^{h+1}\right)\right| = 10^{-10}$, then $\left|\xi_t^h - \xi_t^{h+1}\right| = 2.98 \times 10^{-7}$.

and ξ_t^h to satisfy the stopping rule

$$\left\|\xi_t^h - \xi_t^{h+1}\right\| \leq \epsilon_{\text{in}} \tag{6}$$

for the iteration $h+1$ when we terminate the contracting mapping (5).[5] Let $\xi_t(\theta, \epsilon_{\text{in}})$ denote the first ξ_t^{h+1} such that the stopping rule (6) is satisfied. We then use $\xi_t(\theta, \epsilon_{\text{in}})$ to approximate $\xi_t(\theta)$.

We state the contraction mapping theorem, which provides the rate of convergence of the inner loop of the NFP algorithm.

Theorem 1. *Let $\mathcal{T}_\theta : \mathbb{R}^n \to \mathbb{R}^n$ be an iteration function and let $A_r = \left\{\xi \mid \left\|\xi - \xi^0\right\| < r\right\}$ be a ball of radius r around a given starting point $\xi^0 \in \mathbb{R}^n$. Assume that $\mathcal{T}_\theta$ is a contraction mapping in A_r, meaning*

$$\xi, \tilde{\xi} \in A_r \Rightarrow \left\|\mathcal{T}_\theta(\xi) - \mathcal{T}_\theta\left(\tilde{\xi}\right)\right\| \leq L(\theta)\left\|\xi - \tilde{\xi}\right\|,$$

where $L(\theta) < 1$ is called a Lipschitz constant. Then if

$$\left\|\xi^0 - \mathcal{T}_\theta\left(\xi^0\right)\right\| \leq (1 - L(\theta))\, r,$$

the multidimensional equation $\xi = \mathcal{T}_\theta(\xi)$ has a unique solution ξ^* in the closure of A_r, $\bar{A}_r = \{\xi \mid \|\xi - \xi_0\| \leq r\}$. This solution can be obtained by the convergent iteration process $\xi^{h+1} = \mathcal{T}_\theta\left(\xi^h\right)$, for $h = 0, 1, \ldots$ The error at the h^{th} iteration is bounded:

$$\left\|\xi^h - \xi^*\right\| \leq \left\|\xi^h - \xi^{h-1}\right\| \frac{L(\theta)}{1 - L(\theta)} \leq \left\|\xi^1 - \xi^0\right\| \frac{L(\theta)^h}{1 - L(\theta)}.$$

Theorem 1 states that at every iteration of the contraction mapping, the upper bound for the norm of the error is multiplied by $L(\theta)$. Consequently, the rate of convergence of the contraction mapping is linear and is measured by the Lipschitz constant $L(\theta)$. For a proof of the theorem, see Dahlquist and Björck (2008).

The following theorem shows how to express the Lipschitz constant for a mapping $\mathcal{T}_\theta(\xi)$ in terms of $\nabla\mathcal{T}_\theta(\xi)$, the Jacobian of $\mathcal{T}_\theta(\xi)$. We then use the Lipschitz constant result to assess an upper bound for the performance of the NFP algorithm.

Theorem 2. *Let the function $\mathcal{T}_\theta(\xi) : \mathbb{R}^n \to \mathbb{R}^n$ be differentiable in a convex set $D \subset \mathbb{R}^n$. Then $L(\theta) = \max_{\xi \in D} \|\nabla\mathcal{T}_\theta(\xi)\|$ is a Lipschitz constant for $\mathcal{T}$.*

The contraction mapping proposed by BLP to invert the market shares is

$$\mathcal{T}_\theta(\xi) = \xi + \log S - \log s(\xi; \theta).$$

[5] $\|(a_1, \ldots, a_b)\|$ is a distance measure, such as $\max(a_1, \ldots, a_b)$.

We define a Lipschitz constant for the BLP contraction mapping $\mathcal{T}$ given structural parameters θ as

$$L(\theta) \quad = \max_{\xi \in D} \left\| \nabla \mathcal{T}_\theta \left(\xi \right) \right\| = \max_{\xi \in D} \left\| I - \nabla \left(\log s \left(\xi ; \theta \right) \right) \right\|,$$

where

$$\frac{\partial \log \left(s_j \left(\xi_t ; \theta \right) \right)}{\partial \xi_{lt}} = \begin{cases} \dfrac{\sum_{r=1}^{n_s} \left[\left(\dfrac{\exp\left(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)} \right) - \left(\dfrac{\exp\left(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)} \right)^2 \right]}{\sum_{r=1}^{n_s} \frac{\exp\left(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)}}, & \text{if} \quad j = l \\ \dfrac{-\sum_{r=1}^{n_s} \left[\left(\dfrac{\exp\left(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)} \right) \left(\dfrac{\exp\left(\beta^{0,r} + x'_{l,t} \beta^{x,r} - \beta^{p,r} p_{l,t} + \xi_{l,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)} \right) \right]}{\sum_{r=1}^{n_s} \frac{\exp\left(\beta^{0,r} + x'_{j,t} \beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t} \right)}{1 + \sum_{k=1}^{J} \exp\left(\beta^{0,r} + x'_{k,t} \beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t} \right)}}, & \text{if} \quad j \neq l. \end{cases}$$

It is difficult to get precise intuition for the Lipschitz constant as it is the norm of a matrix. But, roughly speaking, the Lipschitz constant is related to the matrix of own and cross demand elasticities with respect to the demand shocks, ξ, as the jth element along the main diagonal is $\frac{\partial s_{j,t}}{\partial \xi_{j,t}} \frac{1}{s_{j,t}}$. In Section 7.2, we use the Lipschitz constant to distinguish between simulated datasets where we expect the contraction mapping to perform relatively slowly or rapidly.

4.2 Ensuring Convergence for the Outer Loop in NFP

In this subsection we show how numerical error from the inner loop propagates into the outer loop. We then characterize the corresponding numerical inaccuracy in the criterion function, $Q\left(\theta\right)$, and its gradient. This analysis gives a rate result for the tolerance level for the optimization in the outer loop to ensure that the optimization routine is able to report convergence.

We denote by $\xi\left(\theta, \epsilon_{\text{in}}\right)$ the numerical calculation of the demand shocks corresponding to a given value for θ and an inner loop tolerance ϵ_{in}. We also denote the true demand shocks as $\xi\left(\theta, 0\right)$. We let $Q\left(\xi\left(\theta, \epsilon_{\text{in}}\right)\right)$ be the programmed GMM objective function with the inner loop tolerance ϵ_{in}. In a duplication of notation, let $Q\left(\xi\right)$ be the GMM objective function for an arbitrary guess of ξ. We use big-O notation.

The following theorem characterizes the bias in evaluating the GMM objective function and its gradient at any structural parameters, θ, when there exist inner loop numerical errors.

Theorem 3. *Let $L\left(\theta\right)$ be the Lipschitz constant for the inner loop contraction mapping. For any structural parameters θ and given an inner loop tolerance ϵ_{in},*

1. $\left| Q\left(\xi(\theta, \epsilon_{\text{in}})\right) - Q\left(\xi\left(\theta, 0\right)\right) \right| = O\left(\frac{L(\theta)}{1 - L(\theta)} \epsilon_{\text{in}} \right)$

2. $\left\|\nabla_\theta Q\left(\xi\left(\theta\right)\right)\big|_{\xi=\xi(\theta,\epsilon_{\text{in}})} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\big|_{\xi=\xi(\theta,0)}\right\| = O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$,

assuming both $\left\|\frac{\partial Q(\xi)}{\partial \xi}\big|_{\xi=\xi(\theta,0)}\right\|$ *and* $\left\|\frac{\partial \nabla_\theta Q(\xi(\theta))}{\partial \xi}\big|_{\xi=\xi(\theta,0)}\right\|$ *are bounded.*

The proof is in the appendix. Theorem 3 states that the biases in evaluating the GMM objective function and its gradient at any structural parameters are of the same order as the inner loop tolerance adjusted by the Lipschitz constant for the inner loop contraction mapping.

Convergence of an unconstrained optimization problem is declared when the norm of the gradient, $\left\|\nabla Q\left(\theta\right)\right\|$, of the GMM objective function is smaller than a pre-determined outer loop tolerance level, ϵ_{out}: $\left\|\nabla Q\left(\theta\right)\right\| \leq \epsilon_{\text{out}}$. The next theorem shows that the choice of the outer loop tolerance, ϵ_{out}, should depend on the inner loop tolerance, ϵ_{in}.

Theorem 4. *Let* $\hat{\theta}(\epsilon_{\text{in}}) = \arg\max_\theta \left\{Q\left(\xi\left(\theta, \epsilon_{\text{in}}\right)\right)\right\}$. *In order for the outer loop GMM minimization to converge, the outer loop tolerance* ϵ_{out} *should be chosen to satisfy* $\epsilon_{\text{out}} = O\left(\epsilon_{\text{in}}\right)$, *assuming* $\left\|\nabla_\theta^2 Q\left(\xi\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)}\right\|$ *is bounded.*

The proof of this theorem shows that if ϵ_{in} is large (the inner loop is loose), then the gradient will be numerically inaccurate. Therefore, ϵ_{out} needs to be large (the outer loop must be loose) for an optimization routine to converge. The proof is in the appendix.

These theorems summarize the numerical concerns associated with loosening the inner loop tolerance, ϵ_{in}, to speed up the convergence of the contraction mapping. By Theorem 4, the resulting imprecision in the gradient could prevent the optimization routine from detecting a local minimum and converging. In turn, the researcher may need to loosen the outer loop tolerance to ensure the convergence of the outer loop optimization. Raising ϵ_{out} reduces precision in the estimates and, worse, could generate an estimate that is not a valid local minimum.

The default value for ϵ_{out} in most packages is a small number, such as 10^{-6} . In practice, we have found cases where BLP's approach was implemented using $\epsilon_{\text{out}} = 10^{-2}$. Alternatively, others have customized an adaptive version of the inner loop tolerance.[6] In our Monte Carlo simulations below, we will illustrate how a loose stopping criterion for the outer loop can cause the routine to terminate early and produce incorrect point estimates. In some instances, these estimates do not satisfy the first-order conditions for a local minimizer.

[6]The procedure consists of using a loose inner loop tolerance when the parameter estimates appear "far" from the solution and switching to a tighter inner loop tolerance when the parameter estimates are "close" to the solution. The switch between the loose and tight inner loop tolerances is usually based on the difference between the successive parameter iterates, e.g, if $\left\|\theta^{k+1} - \theta^k\right\| \leq 0.01$, then $\epsilon_{\text{in}} = 10^{-8}$; otherwise, $\epsilon_{\text{in}} = 10^{-6}$. Suppose that the following sequence of iterates occur: $\left\|\theta^{k+1} - \theta^k\right\| \geq 0.01$ ($\epsilon_{\text{in}} = 10^{-6}$), $\left\|\theta^{k+2} - \theta^{k+1}\right\| \leq 0.01$ ($\epsilon_{\text{in}} = 10^{-8}$), and $\left\|\theta^{k+2} - \theta^{k+1}\right\| \geq 0.01$ ($\epsilon_{\text{in}} = 10^{-6}$). The NFP objective value can oscillate because of the use of two different inner loop tolerances. This oscillation can prevent the NFP approach from converging.

4.3 Finite-Sample Bias in Parameter Estimates from the Inner Loop Numerical Error

In this section, we discuss the small-sample biases associated with inner loop numerical error. Assume, given ϵ_{in}, that we have chosen ϵ_{out} to ensure that NFP is able to report convergence. Let $\theta^* = \arg\min_{\theta} \{Q(\xi(\theta, 0)\}$ be the minimizer of the finite-sample objective function without numerical error. We study the upper bound on the bias in the final estimates, $\hat{\theta}(\epsilon_{\text{in}}) - \theta^*$, from using an inner loop-tolerance ϵ_{in}.

Theorem 5. *Assuming* $\left\| \nabla_\xi Q(\xi) \Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\|$ *is bounded, the difference between the finite-sample minimizers with and without inner loop error satisfies*

$$O\left(\left\|\hat{\theta}(\epsilon_{\text{in}}) - \theta^*\right\|^2\right) \leq \left|Q\left(\xi\left(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}\right)\right) - Q(\xi(\theta^*, 0))\right| + O\left(\frac{L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}{1 - L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}\epsilon_{\text{in}}\right).$$

The proof is in the appendix. The right side of the inequality provides an upper bound for the order of the square of the numerical error in the parameters. There are two terms in this upper bound. The first term, $\left|Q\left(\xi\left(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}\right)\right) - Q(\xi(\theta^*, 0))\right|$, is the bias in the GMM function evaluated at the finite-sample true and estimated parameter values.[7] The second term arises from $\xi\left(\hat{\theta}(\epsilon_{\text{in}}), \epsilon_{\text{in}}\right) - \xi\left(\hat{\theta}(\epsilon_{\text{in}}), 0\right)$, the bias in demand shocks with and without the inner loop error, and is of the same order as $\frac{L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}{1-L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}\epsilon_{\text{in}}$.[8]

The bound in Theorem 5 is not always sharp or large. In our Monte Carlo experiments below, we will show that the parameter errors associated with improper minimization and failure to detect convergence are more severe issues in practice.

4.4 Large Sample Bias from the Inner Loop Numerical Error

The previous section focused only on numerical errors for a finite data set. Now consider θ^0, the true parameters in the data generating process. Here we explore how numerical errors in the inner loop affect the consistency of the BLP estimator.

Recall that $\hat{\theta}(\epsilon_{\text{in}})$ corresponds to the minimizer of $Q(\xi(\theta, \epsilon_{\text{in}}))$, the biased GMM objective function with the inner loop tolerance, ϵ_{in}. Let $\bar{Q}(\xi(\theta, \epsilon_{\text{in}})) = E[Q(\xi(\theta, \epsilon_{\text{in}}))]$ be the probability limit of $Q(\xi(\theta, \epsilon_{\text{in}}))$, as either $T \to \infty$ or $J \to \infty$, as in Berry, Linton and Pakes (2004). Clearly, $\theta^0 = \arg\min \bar{Q}(\xi(\theta, 0))$ if the BLP model is identified.

[7] This term is related to a formalization of folk knowledge in the numerical-optimization literature . If a function to be optimized has error ϵ, then the minimizer of the function with error could have error of $\sqrt{\epsilon}$; see Chapter 8 in Gill, Murray and Wright (1981).

[8] Ackerberg, Geweke and Hahn (Theorem 2, 2009) studied the case where the objective function is differentiable in the equivalent of inner loop error and found a linear rate. However, in the BLP setting, the GMM objective function is not differentiable with respect to inner loop error ϵ_{in}. In our proof, we take this non-differentiability into account and obtain a square-root upper bound.

Let asymptotics be in the number of markets, T, and let each market be an iid observation. By standard consistency arguments (Newey and McFadden 1994), θ^* will converge to θ^0 if $Q\left(\xi\left(\theta,0\right)\right)$ converges to $\bar{Q}\left(\xi\left(\theta,0\right)\right)$ uniformly, which is the case with a standard GMM estimator. Further, the rate of convergence of the estimator without numerical error from the inner loop is the standard parametric rate, $\sqrt{T}$. By the triangle inequality,

$$\left\|\hat{\theta}\left(\epsilon_{\text{in}}\right)-\theta^0\right\| \leq \left\|\hat{\theta}\left(\epsilon_{\text{in}}\right)-\theta^*\right\| + \left\|\theta^*-\theta^0\right\| \leq$$
$$O\left(\sqrt{\left|Q\left(\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),\epsilon_{\text{in}}\right)\right)-Q\left(\xi\left(\theta^*,0\right)\right)\right|+\frac{L\left(\hat{\theta}\left(\epsilon_{\text{in}}\right)\right)}{1-L\left(\hat{\theta}\left(\epsilon_{\text{in}}\right)\right)}\epsilon_{\text{in}}}\right)+O\left(1/\sqrt{T}\right). \quad (7)$$

The asymptotic bias from numerical error in the inner loop persists and does not shrink asymptotically. This is intuitive: inner loop error would introduce numerical errors in the parameter estimates even if the population data were used.

4.5 Loose Inner Loop Tolerances and Numerical Derivatives

Most researchers use gradient-based optimization routines, as perhaps they should given that the GMM objective function is smooth. Gradient-based optimization requires derivative information, by definition. One approach is to derive algebraic expressions for the derivatives and then to code them manually. Our results above assume that the researcher's optimizer has information on the exact derivatives. In many applications, such as the dynamic demand model we study below, calculating and coding derivatives can be very time consuming. Instead, one may use numerical derivatives that approximate the gradient as follows

$$\nabla_d Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)=\left\{\frac{Q\left(\xi\left(\theta+de_k,\epsilon_{\text{in}}\right)\right)-Q\left(\xi\left(\theta-de_k,\epsilon_{\text{in}}\right)\right)}{2d}\right\}_{k=1}^{|\theta|}, \quad (8)$$

where d is a scalar perturbation and e_k is a vector of 0's, except for a 1 in the kth position of e_k. As $d\to 0$, $\nabla_d Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)$ converges to $\nabla Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)$, the numerically accurate gradient of $Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)$. However, the optimization of the GMM objective function requires the true derivatives without inner loop error, $\nabla Q\left(\xi\left(\theta,0\right)\right)$.

Lemma 9.1 in Nocedal and Wright (2006) shows that the numerical error in the gradient is bounded,

$$\left\|\nabla_d Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)-\nabla Q\left(\xi\left(\theta,0\right)\right)\right\|_\infty \leq O\left(d^2\right)+\frac{1}{d}O\left(\frac{L\left(\theta\right)}{1-L\left(\theta\right)}\epsilon_{\text{in}}\right).$$

There are two terms in this bound. The $O\left(d^2\right)$ term represents the standard error that arises from numerical differentiation, (8). As $d\to 0$, the $O\left(d^2\right)$ term converges to 0. The second

term $\frac{1}{d}O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$ arises from the numerical error in the objective function, for a given $\epsilon_{\text{in}} > 0$. The $O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$ term comes from part 1 of Theorem 3. If $\frac{1}{d}O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$ is relatively large, as it is when the inner loop tolerance is loose, then the bound on the error in the gradient is large. In this case, a gradient-based routine can search in the wrong direction, and end up stopping at a parameter far from a local minimum. Therefore, combining loose inner loop tolerances and numerical derivatives may produce an unreliable solver. Note that as $d \to 0$, the term $\frac{1}{d}O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right) \to \infty$. So setting d to be small exacerbates the numerical error arising from using a loose inner loop tolerance.

5 MPEC: A Constrained Optimization Approach

In this section, we propose an alternative algorithm to compute the BLP objective function based on Su and Judd's (2008) constrained optimization approach for estimating structural models, MPEC. Originally, Su and Judd use this approach to compute the equilibrium for an economic model. We use this same insight to solve for the fixed point associated with the inversion of market shares. MPEC relegates all numerical computation to a single call to a state-of-the-art optimization package, rather than the user's own customized algorithm.

Let W be the GMM weighting matrix. Our constrained optimization formulation is

$$\begin{array}{rl} \min\limits_{\theta,\xi} & g\left(\xi\right)' W g\left(\xi\right) \\ \text{subject to} & s\left(\xi;\theta\right) = S \end{array} . \tag{9}$$

The moment condition term $g\left(\xi\right)$ is simply $g\left(\xi\right) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\xi_{j,t} \cdot h\left(z_{j,t}, x_{j,t}\right)$. In MPEC, the market share equations are introduced as nonlinear constraints to the optimization problem. The objective function is specified primitively as a function of the demand shocks ξ. We optimize over both the demand shocks ξ and the structural parameters θ.

The constrained optimization problem defined by (9) can be solved using a modern nonlinear optimization package developed by researchers in numerical optimization. The defaults on feasibility and optimality tolerances in nonlinear solvers for constrained optimization are usually sufficient. In contrast, NFP requires a customized nested fixed point calculation, including the choice of tolerance, which could result in naive errors.

The following theorem shows the equivalence of the first-order conditions between NFP (4) and the constrained optimization formulation (9).

Theorem 6. *Let the BLP demand model admit a contraction mapping. The set of first-order conditions to the MPEC minimization problem in* (9) *is equivalent to the set of first-order conditions to the true (no numerical error) NFP method that minimizes* (4)*.*

The proof appears in the appendix. Theorem 6 states that any first-order stationary

point of (4) is also a stationary point of (9), and vice versa. The MPEC and NFP algorithms produce the same statistical estimator. Any statistical property of the original BLP estimator applies to the estimator when computed via MPEC. Hypothesis tests, standard errors and confidence intervals are the same for both methods. The GMM standard errors are discussed in Berry, Linton and Pakes (2004) for $J \to \infty$ and standard references for GMM for $T \to \infty$ (Newey and McFadden 1994).

The MPEC approach is theoretically superior to NFP in terms of the rate of convergence because modern constrained optimization solvers use Newton-type methods to solve the Karush-Kuhn-Tucker (KKT) system of the first-order optimality conditions. To formalize this discussion , we first define rate of convergence as follows:

Definition 1. *Let $\{\theta_k\}$ be a sequence in $\mathbb{R}^n$ and θ^* be a point in $\mathbb{R}^n$. Then we say that*

1. *θ_k converges to θ^* Q-linearly if there is a constant $r \in (0,1)$ such that $\frac{\|\theta_{k+1}-\theta^*\|}{\|\theta_k-\theta^*\|} \leq r$ for all k sufficiently large.*
2. *θ_k converges to θ^* Q-superlinearly if $\lim_{k\to\infty} \frac{\|\theta_{k+1}-\theta^*\|}{\|\theta_k-\theta^*\|} = 0$.*
3. *θ_k converges to θ^* Q-quadratically if there is a constant $K > 0$ such that $\frac{\|\theta_{k+1}-\theta^*\|}{\|\theta_k-\theta^*\|^2} \leq K$ for all k sufficiently large.*

The theoretical convergence properties of the NFP outer loop have not been studied. However, due to the numerical inversion in the inner loop, we conjecture that NFP is at best superlinearly convergent. We will verify this property in our numerical experiments. A theoretical advantage of Newton-type methods, and hence MPEC, is that they are quadratically convergent when the iterates are close to a local solution (e.g., Kelley 1995, 1999, 2003 and Nocedal and Wright 2006). We will verify this property in our Monte Carlo experiments below.

MPEC eliminates the call to the inner loop, which can create an additional theoretical speed advantage. In contrast, NFP makes a call to a linearly convergent contraction mapping in the inner loop. As we show in Section 7.2, the contraction mapping in the NFP algorithm might slow down as the Lipschitz constant approaches one. A partial solution might consist of using a Newton-type method to solve the inner loop of the NFP algorithm (Rust 1987, Bresnahan, Stern and Trajtenberg 1997, Davis 2006). However, this too would require customized coding.[9] Also, this approach does not resolve the issue of inner loop error if the inner loop tolerance is too loose.

There are several reasons why MPEC may be faster than NFP. One potential source of speed improvement comes from the fact that MPEC allows constraints to be violated during

[9]Most commercial optimization software does not permit being nested inside another call to the same software. Therefore, it would be more convenient to use MPEC and have a single call to a Newton-type solver.

optimization. In contrast, the NFP algorithm requires solving the share equation (3) *exactly* for every parameter θ examined in the optimization outer loop. Modern optimization solvers do not enforce that the constraints are satisfied at every iteration. The constraints only need to hold at the solution. This flexibility avoids wasting computational time on iterates far away from the true parameters. Another speed advantage arises from the sparsity of the market share equations, because demand shocks for a market t do not enter the constraints for other markets $t' \neq t$. The solver can exploit the sparsity of the corresponding constraint Jacobian.

We can exploit sparsity even further in the implementation of MPEC for the BLP model. By treating the moments as additional parameters, we can re-state the problem in (9) as

$$\begin{array}{rll} \min\limits_{\theta,\xi,\eta} & \eta' W \eta & \\ \text{subject to} & g\left(\xi\right) = \eta & . \\ & s(\xi;\theta) = S & \end{array} \tag{10}$$

It is easy to see that the two formulations, (9) and (10), are equivalent. The objective function in (10) is now a simple quadratic, $\eta' W \eta$, rather than a more complex function of ξ. The additional constraint $g(\xi) - \eta = 0$ is linear in both ξ and η and, hence, does not increase computational difficulty. The advantage with this alternative formation is that we increase the sparsity of the constraint Jacobian and the Hessian of the Lagrangian function by adding additional variables and linear constraints. In numerical optimization, it is often easier to solve a large but sparse problem than a small but dense problem. In our Monte Carlo experiments below, we will show that increasing the sample size and, hence, the dimension of the optimization problem do not appear to disadvantage MPEC relative to NFP.

The relative advantages of MPEC over NFP are not unique to GMM estimation. In Appendix C, we show that likelihood-based approaches require inverting the market share system and, hence, have typically been estimated using NFP.[10]

6 Parameter Errors from Loose Inner Loop Tolerances in the NFP Algorithm

In this section, we provide empirical support for the numerical problems derived in Section 4. We provide examples in which parameter errors can arise both in the context of sampling experiments and with pseudo-real field data. We use the generated data to show that a com-

[10] Nemirovsky and Yudin (1979) derive a lower bound on the number of function evaluations needed to approximate a global solution for a non-convex, unconstrained optimization problem. The bound involves a constant raised to a power in the number of parameters. However, this type of theoretical bound does not offer a full comparison between MPEC and NFP, as NFP involves solving a nested inner loop, which is not part of the analysis in Nemirovsky and Yudin. Furthermore, the evaluation error in the GMM objective function could result in inaccurate approximation of a global solution.

bination of numerical derivatives and loose inner loop tolerances can lead to grossly incorrect parameter estimates. We use the field data to show that incorrect parameter estimates can arise even with closed-form derivatives.

6.1 NFP Algorithm Implementation

For estimation, we use a one-step GMM estimator with the weighting matrix $W = (Z'Z)^{-1}$, where Z is the $TJ \times C$ matrix of instruments $h(z_{j,t}, x_{j,t})$. We also exploit the normality assumption for $F_\beta(\beta; \theta)$ to concentrate the parameters characterizing the means of the random coefficients out of the parameter search. Both decisions follow Nevo (2000b).

We compare three implementations of the NFP algorithm, each initialized with the same starting values. In the first scenario, we use a tight outer loop tolerance, $\epsilon_{\text{out}} = 10^{-6}$, and a loose inner loop tolerance, $\epsilon_{\text{in}} = 10^{-4}$. While this outer loop tolerance is tight in the sense that it is the default setting for most state-of-the-art optimization algorithms, we conjecture that the inner loop tolerance is too loose.

In the second scenario, we use a loose outer loop tolerance, $\epsilon_{\text{out}} = 10^{-2}$, and a loose inner loop tolerance, $\epsilon_{\text{in}} = 10^{-4}$. Following Theorem 4, one can think of this scenario as representing the attempt of the researcher to loosen the outer loop to promote convergence. In practice, the converged point may not actually satisfy the first-order conditions. In the third scenario, we implement the "best practice" settings for the NFP algorithm with $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$.

Our estimation code is written in the MATLAB programming environment. We use the TOMLAB interface to call the KNITRO optimization package (Byrd, Hribar, and Nocedal 1999, Byrd, Nocedal, and Waltz, 2006) in MATLAB.[11]

6.2 The Synthetic Data-Generating Process

Our first empirical example consists of a synthetic dataset based on the demand model in Section 2. We construct $T = 50$ independent markets, each with the same set of $J = 25$ products. Each product j has $K = 3$ observed, market-invariant characteristics that are generated as follows:

$$\begin{bmatrix} x_{1,j} \\ x_{2,j} \\ x_{3,j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.8 & 0.3 \\ -0.8 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix} \right).$$

[11] We found that MATLAB's included solvers, fminunc and fmincon, often fail to converge to a local minimum.

In addition, each product j has a market-specific vertical characteristic: $\xi_{j,t} \sim$ i.i.d. $N(0,1)$. Finally, each product j has a market-specific price generated as follows:

$$p_{j,t} = \left| 0.5 \cdot \xi_{j,t} + e_{j,t} + 1.1 \cdot \sum_{k=1}^{3} x_{k,j} \right|,$$

where $e_{j,t} \sim N(0,1)$ is an innovation that enters price.

For each product j in market t, there is a separate vector, $z_{j,t}$, of $D = 6$ underlying instruments generated as follows: $z_{j,t,d} \sim U(0,1) + \frac{1}{4}\left(e_{j,t} + 1.1 \cdot \sum_{k=1}^{3} x_{k,j,t}\right)$, where $U(0,1)$ is the realization of a uniform random variable and $e_{j,t}$ is the price innovation. In addition, we also use higher-order polynomial expansions of the excluded instruments, z_{jt}, and the exogenous regressors, $x_j : z_{j,t,d}^2$, $z_{j,t,d}^3$, $x_{j,k}^2$, $x_{j,k}^3$, $\prod_{d=1}^{D} z_{j,t,d}$, $\prod_{k=1}^{K} x_{j,k}$, $z_{j,t,d} \cdot x_{j1}$, and $z_{j,t,d} \cdot x_{j,t,2}$. There are 42 total moments.

There are five dimensions of consumer preference, $\beta_i = \left\{\beta_i^0, \beta_i^1, \beta_i^2, \beta_i^3, \beta_i^p\right\}$ (an intercept, $K = 3$ attributes and price), each distributed independently normal with means and variances: $E[\beta_i] = \{-1.0, 1.5, 1.5, 0.5, -3.0\}$ and $\text{Var}[\beta_i] = \{0.5, 0.5, 0.5, 0.5, 0.2\}$.

We simulate the integral in the market share equation, (3), with $n_s = 100$ independent standard normal draws. Because our focus is not on numerical integration error, we use the same set of 100 draws to compute market shares in the data generation and estimation phases.

6.3 Synthetic Data, Numerical Derivatives and False Parameter Estimates

We create one simulated synthetic dataset, using the data-generating process from Section 6.2. For this example, we use numerical derivatives. We construct 100 independent starting values for the model parameters $\text{SD}[\beta_i]$ by drawing them from a uniform distribution $U(0,7)$.[12] We run each of the three NFP implementations described in Section 6.1 for each of the 100 vectors of starting values.

For each implementation, we report the results across the 100 different starting values in Table 1. The first row reports the fraction of runs for which the routine reports convergence. Supporting Theorem 4, we find in column one that the optimization routine will never report convergence if the inner loop tolerance is loose, $\epsilon_{\text{in}} = 10^{-4}$, even when the outer loop tolerance has the default tight tolerance of $\epsilon_{\text{out}} = 10^{-6}$. In contrast, column two indicates that the algorithm is more likely to converge (54% of the runs) when we also loosen the tolerance on the outer loop. Below, we will show that convergence in this case is misleading; the estimates are far from the truth. Finally, NFP with tight tolerances converges in 95% of the runs.

To diagnose the quality of the estimates, the second row of Table 1 shows the fraction of runs where the reported GMM objective function value was within 1% of the lowest objective function that we found across all three NFP implementations and all 100 starting values (300

[12] Recall that the means $E[\beta_i]$ are concentrated out of the optimization problem (Nevo 2000b).

cases). We call this value the "global" minimum, although of course we cannot prove we have found a true global minimum. In the first two columns, corresponding to the scenario with a loose inner loop and the scenario with a loose inner and a loose outer loop, respectively, none of the 100 starting values produced the so-called global minimum. In contrast, NFP tight found the global minimum in each of the 100 runs. In general, there can be multiple local minima and NFP could have converged to any one of them.

The third and fourth rows of Table 1 provide measures to assess the economic implications of our different implementations. We use estimated price elasticities to show how naive implementations could produce misleading economic predictions. In the third row, we report the mean own-price elasticity, across all $H = 100$ starting values, all J products and all T markets:

$$\frac{1}{H}\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{j=1}^{J}\eta_{j,t}^{p}\left(\hat{\theta}^{h}\right),$$

where $\hat{\theta}^{h}$ is the vector of parameter estimates for the h^{th} starting value and $\eta_{j,t}^{p}\left(\hat{\theta}^{h}\right)$ is the own price-elasticity of firm j in market t, at those parameters. The fourth row reports the standard deviation of the mean own-price elasticities across all 100 starting values.

Referring to the third row, the final column reports the own-price elasticity of demand evaluated at the true parameter values: -5.68. As expected, NFP with a tight tolerance produces an estimate near the truth, -5.77. However, our two loose implementations of NFP produce mean elasticities that are not nearly as close to the truth. The mean of the NFP loose-inner implementation is -7.24, higher in absolute value than the true value of -5.68. The loose-both mean is -7.49. The standard deviations of own-price elasticities for the loose inner loop tolerances are huge: 5.48 and 5.55. These findings are not surprising in light of the numerical theory results about loose inner loop tolerances and numerical derivatives in Section 4.

It is true that the reported elasticities evaluated at each implementation's best estimate, in terms of the objective function values, appear to be relatively close to the truth. One should not conclude from this evidence that loose implementations can be resolved simply by using many starting values. None of the 100 runs of NFP with a loose inner loop converged, and only 54% of the runs of NFP with both a loose inner and outer loop converged. In practice, a researcher cannot prove whether non-converged runs are in fact close to a solution.

6.4 Parameter Errors with Nevo's Data and Closed-Form Derivatives

Our second empirical example consists of the cereal dataset from Nevo (2000b), which we use to study the numerical properties of NFP in the context of a pseudo-real dataset. We refer the reader to Nevo (2000b) for a description of these data. For this example, we use analytic

derivatives, which should improve the performance of all three NFP implementations.

The results in Table 2 are of the same format as Table 1. For each of the three implementations (loose inner, loose both and tight both) we use the same set of 50 starting values for the same cereal dataset. We pick our starting values by taking 50 draws from the standard normal distribution.[13] We set the inner loop tolerance to be 10^{-14}. We also report results for the Nelder-Meade or simplex algorithm, which we will discuss below. As Theorem 4 predicts, in row 1 we find that 0% of the NFP-loose inner loop starting values converge. Loosening the outer loop is one approach to finding convergence; the second column finds that 76% of starting values report convergence when this is done. 100% of the starting values converge for NFP tight. The second row shows that 100% of the NFP-tight starting values find the apparent global minimum, 0.00202, in Nevo's cereal data. None of the NFP loose tolerance implementations find the global minimum.

The loose inner loop and loose-both methods find a mean own-price elasticity of -3.82 and -3.69, respectively. This is about half the value of -7.43 found with NFP tight. Further, the estimates are all tightly clustered around the same points. With standard deviations of 0.40 and 0.07 for the loose inner loop methods, the answers are for the most part consistently wrong across runs. The fifth row shows that the smallest objective function values found by the loose-inner loop and loose-both routines are 0.00213 and 0.00683, respectively. The second result is far from the truth of 0.00202. We manually inspected all 50 starting values and found that only 1 of the 50 runs using the loose inner loop was anywhere near the apparent global minimum. More striking is the fact that the remaining 49 runs all converge to the same incorrect point as in the loose-both case. In short, the numerical imprecision of the gradient of the objective function under the loose inner loop cases causes the optimization routine to terminate at a point that is not a local minimum. This problem is not mitigated by using a professional optimization package like KNITRO. The only solution is to use a tight inner loop tolerance, such as 10^{-14}.

Knittel and Metaxoglou (2008) recently used these same data to study a related potential concern with BLP. They report that the parameter estimates are extremely sensitive to the starting values used for the NFP algorithm because many local minima could exist in the GMM objective function. Since the BLP problem is not convex, it may indeed have multiple optima. As we saw above, our NFP code finds the same objective function value, 0.00202, which is also the lowest objective value found by Knittel and Metaxoglou (2008).[14] We conclude that with multiple starting values, careful implementation of the numerical procedures (a tight inner loop), and a state-of-the-art optimization solver, the BLP estimator produces reliable

[13] We also experimented with multiplying the starting values by the solution reported in Nevo (2000b). The results were similar.

[14] We also use the MATLAB code by Nevo (2000b), except that we use the KNITRO solver as the search algorithm. For 50 starting points, KNITRO only converges for 25 of the 50 runs. However, all 25 successful runs converge to the same solution with objective value 0.00202.

estimates.

Interestingly, when we resort to non-derivative-based solvers, convergence occurs at points that are not local minima. We start with the Nelder-Meade, or simplex algorithm, using the same 50 starting values as above and a tight inner loop and tight outer loop. Results are summarized in column four of Table 2. None of the 50 runs of the simplex algorithm find the global minimum. Moreover, none of these runs satisfy the first-order optimality conditions. Further, the elasticity estimate of -3.76 is around half of the numerically correct -7.43, and the elasticity's standard deviation across the fifty starting values is a relatively tight 0.35. See McKinnon (1998) and Wright (1996) for further discussion of the problems with the Nelder-Meade algorithm.

Although not reported, we also used MATLAB's genetic algorithm routine for one run. The genetic algorithm found a point with the objective function value 0.043629, which is an order of magnitude higher than 0.00202, the minimum we found using the gradient-based method. We then started KNITRO from this point found by the generic algorithm and KNITRO found the solution with objective value 0.00202. Once again, a non-derivative-based solver fails to locate a local solution.

7 Speed Comparisons of MPEC and NFP

In this section, we use various manipulations of synthetic data and the Nevo cereal data to compare the speed of MPEC and NFP. Hereafter, we focus only on NFP tight since in Section 6 it was found routinely to recover the global optimum with multiple starting values. Our approach involves manipulating aspects of the data-generating process that influence the Lipschitz constant. A higher Lipschitz constant should slow the speed of the contraction mapping. We conjecture that the speed of MPEC should be relatively invariant to the Lipschitz constant since it does not nest a call to the contraction mapping.

7.1 NFP and MPEC Implementations

We code NFP and MPEC using closed-form, first-order derivatives. We use the quadratic objective-function form of MPEC in (10). We supply the sparsity pattern of the constraints to the optimization routine, for MPEC. For all implementations of NFP and MPEC, we use the interior point algorithm in KNITRO. Because we only supply the exact first-order derivatives, we choose options for calculating second-order derivatives based on the number of parameters in the NFP outer loop and the entire MPEC problem. To make our comparisons fair, we select the best settings for each algorithm. In particular, the KNITRO options are HESSOPT = 2 for NFP and HESSOPT = 4 for MPEC. We also use the algorithm options within the interior point method that work the best for NFP and MPEC, respectively. For

NFP, we choose the KNITRO option ALG=1, which is a direct decomposition of the first-order Karush Kuhn Tucker (KKT) matrix, because the number of parameters in the NFP outer loop is small. For MPEC, we choose the option ALG=2, which uses a conjugate gradient iteration to solve the first-order KKT matrix, because the number of variables in MPEC is usually on the order of several thousand. For detailed descriptions of algorithm options in KNITRO, see Waltz and Plantenga (2009).

An important point for our speed comparison is the choice of starting values. We always use five starting values, which are uniform random numbers. For each NFP starting value, we run the inner loop once and use this vector of demand shocks and mean taste parameters as starting values for MPEC. Effectively, we use the same starting values for both NFP and MPEC in that the two algorithms are initialized to have the same objective function value.

7.2 Lipschitz Constants

We define a base synthetic data case that will then be perturbed to vary the Lipschitz constants in the examples that follow. The model is nearly the same as Section 6.2. As before, we use $T = 50$ markets. The mean of the random coefficients is $E[\beta_i] = (0.1, 1.5, 1.5, 0.5, -3.0)$. The prices are $p_{j,t} = 3 + \xi_{j,t} \cdot 1.5 + u_{j,t} + \sum_{k=1}^{3} x_{k,j,t}$, where $u_{j,t}$ is a uniform$(0, 5)$ random variable. Likewise, $z_{j,t,d} = \tilde{u}_{j,t,d} + \frac{1}{4} \left| u_{j,t} + 1.1 \cdot \sum_{k=1}^{3} x_{k,j,t} \right|$, where $\tilde{u}_{j,t,d}$ is another uniform$(0, 1)$ random variable and $u_{j,t}$ is the same variable as before. For each table below, we calculate 20 different synthetic datasets, and reported means are across these 20 replications.

Recall that the Lipschitz constant derived in Section 4.1 is related to the demand sensitivity to the unobserved quality, $\xi_{j,t}$. Therefore, we experiment with different features of the data-generating process that affect the relative importance of $\xi_{j,t}$ for the market shares. Table 3 reports the Lipschitz constants for the base-case data-generating process. Each cell reports the mean of the Lipschitz constants evaluated at the true parameter values across 30 data sets / replications.

In our first experiment, reported in the first column of Table 3, we manipulate the scale of the parameters, β_i. We multiply the β_i of each of our n_s simulated consumers in the data-generating process by one of the constants listed in the table. We find that the Lipschitz constant is non-monotone in the scale, with the constant first falling and then rising again. This non-monotonicity comes from the fact that our manipulation also changes the levels of the market shares. Nevertheless, holding the sample size fixed, we see fairly large changes in the upper bound on the rate of convergence of the contraction mapping.

The second column of Table 3 increases the standard deviation of the product-and-market-specific demand shocks, $\xi_{j,t}$. When these shocks are more variable, products become more vertically differentiated. Over the range of values we investigate, increases in the standard deviation of the demand shocks increase the Lipschitz constant. The third column of Table

3 changes the number of markets. The number of markets has little impact on the Lipschitz constant. Finally, the fourth column of Table 3 increases the mean of the intercept, $E\left[\beta_i^0\right]$, which changes the value of the inside goods relative to the outside good. As the inside-good share increases, the Lipschitz constant increases.

7.3 Speed Comparisons of MPEC and NFP Using Synthetic Data

We now explore whether there is an implication of the Lipschitz constant for execution time. We compare performances as we vary the mean of the intercept, $E\left[\beta_i^0\right]$, from -1.9 to 3.1. For each scenario, we run 20 replications of the data. For each data replication, we estimate the GMM parameters using our two numerically-accurate algorithms, NFP with a tight inner loop and MPEC. We use five different starting values in each dataset, taking the final point estimates for each algorithm as the run with the lowest objective function value. In all cases, the lowest objective function corresponded to a case where the algorithm reported that a locally optimal solution had been found. We assess the estimates by looking at the own-price elasticities, computed as a mean across products within each market and then across markets. For each algorithm, we report the total CPU time required across all five starting values. The results are reported in Table 4. All numbers in Table 4 are means across the 20 replications.

Turning to Table 4, we can see that our numerical theory prediction holds in practice. As expected, NFP with a tight inner loop tolerance and MPEC converged in all scenarios. We also find that MPEC and NFP almost always generated identical point estimates, as one would expect since they are statistically the same estimator (Theorem 6). Across the 20 runs, MPEC and NFP produced identical estimates for the first four values of $E\left[\beta_i^0\right]$. For the last two values of $E\left[\beta_i^0\right]$, MPEC and NFP are nearly identical. With only five starting values, by happenstance in one or two of the 20 replications, MPEC and NFP found different local minima. Increasing the number of starting values to ten or fifteen would probably resolve this discrepancy. We compute the root mean-squared error (RMSE) and the bias of the own-price elasticities. For a parameter θ_1, the bias is $E\left[\hat{\theta}_1\right] - \theta_1$, where θ_1 is the true value and the expectation is taken over many estimates with independent samples. Likewise, the RMSE is $\sqrt{E\left[\left(E\left[\hat{\theta}_1\right] - \theta_1\right)^2\right]}$. In all cases, the RMSE is low and the bias is moderate at around 0.2, in comparison with a base elasticity of around -12, suggesting that the BLP estimator is capable of recovering true demand elasticities.

Run times for NFP tight vary dramatically with the level of the Lipschitz constant.[15] For the low Lipschitz case with $E\left[\beta_i^0\right] = -1.9$, the average run time across the 20 replications is roughly 17 minutes for NFP and for MPEC. As we increase the intercept, the run times

[15] We use $n_s = 100$ draws to simulate the integrals under NFP and MPEC. In a real empirical application, one would probably use around 10,000 draws, which would increase run times considerably; although most likely not the relative run times of the two algorithms.

for NFP increase, while the run times for MPEC change very little. When $E\left[\beta_i^0\right] = 3.1$, the highest Lipschitz case, a single run with five starting values of NFP takes, on average, 60 minutes, whereas MPEC takes only 13 minutes. Remarkably, in this example the speed of MPEC actually increases slightly as $E\left[\beta_i^0\right]$ decreases. The more striking effect is the decrease in the speed of NFP, as expected. The shares of the outside good range from 90% to 47%, consistent with most empirical applications of BLP. Hence, these speed results are not an artifact of unusual data.

We run an additional set of Monte Carlo experiments to ensure that the relative performance of MPEC is robust to a much larger dataset with many more markets and, hence, many more parameters. Of course, increasing the number of markets increases the number of contraction mappings at each stage of the outer loop in the NFP optimization problem. In principle, NFP might be affected more adversely as the experiment exacerbates the linear convergence of the contraction mapping, versus the quadratic convergence of MPEC.

Table 5 returns to the base specification, and varies only the number of markets, T. The mean intercept is $E\left[\beta_i^0\right] = 1.1$. As the number of markets increases, not surprisingly both methods take longer. MPEC requires only $\frac{1373}{555} = 40\%$ of the time required by NFP for $T = 25$, 41% of the time for $T = 50$, and only 27% of the time for $T = 100$. We conclude that, in this example, the performance advantage of MPEC over NFP actually increases as the number of demand shocks increase.

As Appendix F reports, we also ran a set of Monte Carlo experiments that varied the quality of our data by changing the power of the instruments. In principle, the quality of the data could influence the convexity of the objective function and, hence, the trajectory of the outer loop search. In Appendix F, the speed advantage of MPEC relative to NFP is found to be robust to the quality of the data.

As predicted by the numerical theory, it is easy to find cases where the slow rate of convergence of the inner loop slows the overall execution time of NFP. In contrast, the execution time of MPEC is fairly robust across scenarios. This relationship to run time highlights our earlier concern about the choice of the inner loop tolerance. For real applications with many more products and/or markets (e.g. 25 products and 450 market/quarters in Nevo (2000a, 2001) and 250 products and 10 years in BLP (1995)), run times could be considerably slower than in our Monte Carlo experiments with only 25 products, 50 markets, and 100 simulation draws. Because we have previously demonstrated the potential pitfalls from loosening the inner loop, we therefore recommend MPEC as a safer and more reliable algorithm for the computation of the BLP GMM estimator.

7.4 Speed Comparisons of MPEC and NFP Using Nevo's Data

As a final robustness check, we re-run the comparison of NFP and MPEC using Nevo's pseudo-real cereal data. Like NFP, MPEC converges to the same local minimum with an objective function value of 0.00202 for 48 out of 50 starting values. For only two of the runs, MPEC converges to a different local minimum with a higher objective-function value. In terms of run time for one starting value, we find that MPEC required an average CPU time of 544 seconds whereas NFP required an average CPU time of 763 seconds. In short, the relative performance of MPEC and NFP documented in our Monte Carlo experiments appears to hold in the context of field data.

Recall that part of the underlying theory for why MPEC might perform faster is the quadratic rate of convergence of the Newton-based solver. We confirm this rate of convergence by inspecting the output of the two estimation procedures. Table 6 reports the last iterations of NFP and MPEC for one starting value on the Nevo dataset. The optimality error for NFP does not decrease monotonically and takes 9 iterations to move from an optimality error of 10^{-4} to one of 10^{-7}. In contrast, MPEC takes only five iterations to move from 10^{-1} to 10^{-8}. The feasibility error (the error in the market-share constraints) also takes five iterations to move from 10^{-2} to 10^{-8}. Thus, the relative speed of MPEC versus NFP may indeed reflect, to some extent, the theoretical advantage of a quadratically convergent algorithm.

8 Extension: Dynamic Demand Models

An even more promising frontier for MPEC is in the application of dynamic demand estimation. Starting with Melnikov (2000), a new stream of literature has considered dynamic analogs of BLP with forward-looking consumers making discrete choice purchases of durable goods (Nair 2007, Gordon 2007, Carranza 2008, Gowrisankaran and Rysman 2008, Dubé, Hitsch and Chintagunta 2008, Lee 2008, Schiraldi 2008). The typical implementation involves a nested fixed point approach with two nested inner loops. The first inner loop is the usual numerical inversion of the demand system to obtain the demand shocks, ξ. The second inner loop is the iteration of the Bellman equation to obtain the consumers' value functions. In this section, we describe how MPEC can once again serve as a computationally more attractive solution than NFP.

8.1 Dynamic BLP Model and Algorithms

We specify a simple model of discrete choice demand for a durable good with falling prices over time and two competing products. There is a mass M of potential consumers at date $t = 1$. Consumers are assumed to drop out of the market once they make a purchase. Abstracting

from supply-side specifics, we assume that prices evolve over time as a function of the lagged prices of both firms according to the rule

$$p_{j,t} = \rho_{j,0} + \rho_{j,1} p_{j,t-1} + \rho_{j,2} p_{-j,t-1} + \psi_{j,t} = p'_{t-1}\rho_j + \psi_{j,t},\ j = 1, ..., 2 \tag{11}$$

where $\psi_{j,t}$ is a random supply shock. For the remainder of our discussion, we assume that this supply shock is jointly distributed with the demand shock, $(\xi_{j,t}, \psi_{j,t}) \sim N(0, \Omega)$, and is independent across time periods, firms and markets. We assume that consumers have rational expectations in the sense that they use the true price process, (11), to forecast future prices.

On the demand side, forward-looking consumers now have a real option associated with not purchasing because they can delay adoption to the future, when prices are expected to be lower. A consumer r's expected value of waiting is[16]

$$\begin{aligned} & v_0^r(p_t; \theta^r) \\ = \ & \delta \int \max \left\{ \begin{array}{c} v_0^r(p'_t\rho_j + \psi; \theta^r) + \epsilon_0 \\ \max_j \left\{ \beta_j^r - \alpha^r (p'_t\rho_j + \psi) + \xi_j + \epsilon_j \right\} \end{array} \right\} dF_\epsilon(\epsilon) dF_{\psi,\xi}(\psi, \xi) \\ = \ & \delta \int \left(\log \left(\exp\left(v_0^r(p'_t\rho_j + \psi; \theta^r)\right) + \sum_j \exp\left(\beta_j^r - \alpha^r (p'_t\rho_j + \psi) + \xi_j \right) \right) \right) dF_{\psi,\xi}(\psi, \xi). \end{aligned} \tag{12}$$

To simplify the calculation of the expected value of waiting $v_0^r(p_t; \theta^r)$, we approximate it with Chebyshev polynomials (Judd 1998). We outline the Chebyshev approximation in Appendix D. Our focus on the expected value of waiting, rather than the consumer's value function, is merely exploiting the special structure of this model.

We use a discrete distribution with R mass points to characterize the consumer population's tastes at date $t = 1$,

$$\theta = \left\{ \begin{array}{ll} \theta^1, & \Pr(1) = \lambda_1 \\ \vdots & \vdots \\ \theta^R, & \Pr(R) = 1 - \sum_{r=1}^{R-1} \lambda_r \end{array} \right. ,$$

where $\theta^r = (\alpha^r, \beta^r)$. This heterogeneity implies that certain types of consumers will systematically purchase earlier than others. The mass of consumers of a given type r at the beginning of period t, M_t^r, is

$$M_t^r = \left\{ \begin{array}{ll} M\lambda_r & ,\ t = 1 \\ M_{t-1}^r S_0^r(X_{t-1}; \theta^r) & ,\ t > 1 \end{array} \right. .$$

[16] Here we make the normalization that the location parameter of the Type I Extreme Value distribution equals -0.577.

In a given period t, the market share of product j is

$$s_j\left(p_t;\theta\right) \quad = \quad \sum_{r=1}^{R}\lambda_{t,r}\frac{\exp(\beta_j^r-\alpha^r p_{j,t}+\xi_{j,t})}{\exp\left(v_0^r(p_t;\theta^r)\right)+\sum_{k=1}^{J}\exp\left(\beta_k^r-\alpha^r p_{k,t}+\xi_{k,t}\right)},\ j=1,...,2 \tag{13}$$

where

$$\lambda_{t,r}=\begin{cases}\lambda_r & ,t=1\\ \frac{M_t^r}{\sum_r M_t^r} & ,t>1\end{cases}$$

is the probability mass associated with type r consumers still in the market at date t. The finite-types assumption eases dynamic programming because there is only one unknown value-of-waiting function for each type.

The empirical model consists of the system (11) and (13), which we write more compactly as

$$u_t \equiv \left[\begin{array}{c}\psi_t\\ \xi_t\end{array}\right]=\left[\begin{array}{c}p_t-p_{t-1}'\rho\\ s^{-1}\left(p_t,S_t;\Theta\right)\end{array}\right].$$

The multivariate normal distribution of u_t induces the density on the observable outcomes, $Y_t=(p,S_t)$,

$$f_Y\left(Y_t;\theta,\rho,\Omega\right)=\frac{1}{(2\pi)^J\left|\Omega\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}u_t'\Omega^{-1}u_t\right)\left|J_{t,u\to Y}\right|$$

where $J_{t,u\to Y}$ is the $(2J\times 2J)$ Jacobian matrix corresponding to the transformation-of-variables from u_t to Y_t. We provide the derivation of the Jacobian in Appendix E.

An NFP approach to maximum likelihood estimation of the model parameters amounts to solving the optimization problem

$$\max_{\{\theta,\rho,\Omega\}}\prod_{t=1}^{T}f_Y\left(Y_t;\theta,\rho,\Omega\right). \tag{14}$$

This problem nests two inner loops. Each stage of the outer loop maximization of the likelihood function in (14) nests a call to compute the fixed point of the contraction mapping, (12), in order to obtain the expected value of waiting. There is also a nested call to compute the fixed point of the BLP contraction mapping, (5), to compute the demand shocks ξ_t (the inversion). Numerical error from both these inner loops propagates into the outer loop. Thus, the numerical concerns regarding inner loop convergence tolerance discussed for static BLP are exacerbated with dynamic analogs of BLP.

Let D be the support of the state variables. An MPEC approach to maximum likelihood

estimation of the model parameters amounts to solving the optimization problem

$$\begin{array}{ll} \max\limits_{\{\theta,\rho,\Omega,\xi,v\}} & \prod\limits_{t=1}^{T} \frac{1}{(2\pi)^{J}|\Omega|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}u_t'\Omega_u^{-1}u_t\right)|J_{t,u\to Y}| \\ \text{subject to} & s(\xi_t;\theta) = S_t \,\forall\, t = 1,\ldots,T \\ \text{and} & v_0^r(p_d) = \delta \log\left(\begin{array}{c} \exp\left(v_0^r(p_d'\rho_j+\psi)\right)+... \\ \sum\limits_j \exp\left(\beta_j^r - \alpha^r\left(p_d'\rho_j+\psi\right)+\xi_j\right) \end{array}\right) dF_{\psi,\xi}(\psi,\xi) \\ & \forall\, d \in D,\, r = 1,\ldots,R. \end{array}$$

In this formulation, we now optimize over the demand shocks, ξ, and the expected value of waiting evaluated at each point, $v^r(p_d)$. In this case, $D \subset \mathbb{R}^2_+$ is the support of the two products' prices. While this approach increases the number of parameters in the outer loop optimization problem substantially compared to NFP, MPEC completely eliminates the two inner loops. Chebyshev approximation reduces the dimension of this problem substantially by searching over the Chebyshev weights, rather than over the value function at each point in a discretized state space.

8.2 Dynamic BLP Monte Carlo Experiments

To assess the relative performance of MPEC versus NFP in the context of our dynamic durable goods example, we construct the following Monte Carlo experiments. In the first experiment, we assume there is only a single consumer type, $R = 1$. It is easy to show that in this case, ξ_t can be computed analytically by log-linearizing the market shares, (13). We begin with this case because it only involves a nested call to the calculation of the expected value of waiting. We assume that the consumers' preferences are: $(\beta_1, \beta_2, \alpha) = (4, 3, -1)$. It is straightforward to show that the speed of the contraction mapping associated with the consumer's expected value of waiting is related to the discount factor. Therefore, we compare performance with the two different discount factors $\delta = 0.96$ and $\delta = 0.99$ corresponding roughly to an annual rate and a quarterly rate, respectively, if the interest rate is 4%. We assume that prices are generated as follows:

$$\left[\begin{array}{rcl} p_{1,t} &=& 5 + 0.8p_{1,t-1} + 0.0p_{2,t-1} + \psi_{1,t} \\ p_{2,t} &=& 5 + 0.0p_{1,t-1} + 0.8p_{2,t-1} + \psi_{2,t} \end{array}\right].$$

Finally, we assume the supply and demand shocks satisfy $(\psi_{j,t}, \xi_{j,t}) \sim N\left(0, \left[\begin{array}{cc} 1 & 0.5 \\ 0.5 & 1 \end{array}\right]\right)$ and are independent across markets and time periods. For our Chebyshev approximation, we use six nodes and a fourth-order polynomial. For the NFP algorithm, we use an inner loop tolerance of 10^{-14} for the calculation of the expected value of waiting. We use data on $M = 20$ markets and $T = 50$ time periods per market.

It is very difficult to derive analytic expressions for the Jacobian of the outer loop optimization associated with dynamic BLP, both under NFP and MPEC. As we discussed in Section 4.5, the use of numerical derivatives introduces yet another source of numerical error into the outer loop optimization. However, due to its formulation as a standard constrained-optimization problem, the MPEC algorithm can potentially exploit automatic differentiation to obtain exact derivatives for the outer loop (Griewank and Corliss 1992). To the best of our knowledge, it would be non-standard to apply automatic differentiation to an NFP problem because of the nested calls to fixed-point computations. Therefore, in our Monte Carlo experiments, we compare the performance of NFP using numerical differentiation and MPEC using automatic differentiation.[17]

Results from twenty replications of this first experiment are reported in Table 7, where we use the discount factor δ=0.96. We report the average point estimate and RMSE associated with each of the structural parameters, for MPEC and NFP respectively. With tight inner loop settings and allowing for five different starting values, we find that MPEC and NFP produce identical point estimates. Inspection of the replications found that MPEC and NFP found the same solution for all replications. However, in terms of speed, MPEC is roughly 60% faster in terms of CPU time than NFP.

In Table 8, we run another twenty replications of the same one-type model using a discount factor of $\delta = 0.99$. Even with five starting values per replication, MPEC appears to perform slightly better overall in terms of RMSE, especially for the utility intercepts. Furthermore, MPEC is just under twice as fast at NFP in terms of CPU time.

We also find that NFP fails to converge to the solution in several runs. Focusing on the last row of Table 8, for two replications of NFP, the optimization routine was not able to diagnose convergence for the greatest likelihood function value out of the five starting values. In these two replications, NFP found a worse objective function value than MPEC. All replications of MPEC converged to a local maximum. These results highlight the benefit of automatic differentiation in allowing the routine to check whether it has found a valid local maximum. [18]

We now run a second Monte Carlo experiment that allows for consumer heterogeneity, one of the main elements of the BLP model. We only consider the MPEC algorithm so as to illustrate its applicability to this more computationally challenging case. We allow for a

[17] We use the MAD (MATLAB Automatic Differentiation) package, which is part of TOMLAB.

[18] For three of the twenty replications, MPEC converged to a worse solution than NFP. For $\delta = 0.99$ and unlike the $\delta = 0.96$ case, it appears five starting values are not enough to always find the same solution with MPEC or NFP.

second type of consumer with heterogeneity in price sensitivity:

$$(\beta_1, \beta_2, \alpha)' = \begin{cases} (4, 3, -1) & , \text{ with probability } \lambda{=}0.7 \\ (4, 3, -2) & , \text{ with probability } (1-\lambda){=}0.3 \end{cases} .$$

This case now requires constraints corresponding both to the expected value of waiting for each of the $R = 2$ consumer types and to the share equations. To conserve on computer time, we use only $M = 5$ markets per replication and set $\delta = 0.90$. Results are reported in Table 9. Not only do we find that the parameters are recovered quite well, the average run time requires only 2.6 hours of CPU time. In short, these results are encouraging for MPEC as a practical approach to estimating dynamic BLP with unobserved heterogeneity in a broader context.

9 Conclusions

In this paper, we analyzed the numerical properties of the NFP approach proposed by BLP to estimate the random coefficients logit demand model. NFP is vulnerable to error due to the inner loop. In practice, the NFP approach may also be slow. We used the Lipschitz constant to establish particular data-generating processes for which the inner loop is likely to be slow. A concern is that a loose inner loop tolerance is used precisely when the speed of NFP is slow. Using numerical theory and computational examples with both pseudo-real and synthetic data, we showed that setting loose inner loop tolerances can lead to incorrect parameter estimates and a failure of an optimization routine to report convergence. Using the field data, we showed a case where the estimates with multiple starting values always reported (more or less) the same incorrect estimate.

We then proposed a new constrained optimization formulation, MPEC, which avoids the inner loop for repeatedly inverting the market share equations and, hence, eliminates the numerical error in evaluating the objective function and its gradient. It also delegates all the numerical computation to a single call to a state-of-the-art constrained optimization solver. MPEC produces good estimates relatively fast for all the data-generating processes we considered. Its speed is invariant to the Lipschitz constant, as expected.

As an extension, we adapt the MPEC approach to a new class of applications with forward-looking consumers. The relative advantage of MPEC is even stronger with dynamics because two inner loops must be solved: the dynamic programming problem and the market share inversion. This burdensome collection of three loops (optimization, market shares, dynamic programming) makes the traditional BLP approach nearly untenable in terms of computational time. Indeed, the lack of easily-derivable closed-form derivatives prevents the NFP routine from detecting convergence, in many instances. Current work (Lee 2008, Schiraldi

2008) further extends the number of inner loops being solved in estimation. As demand models become richer, the computational benefits of MPEC over NFP become greater. MPEC can also be used for demand models where there is a unique vector of demand shocks that rationalize the market shares, but no contraction mapping.

While we have conducted our analysis in the context of random coefficients demand estimation, we emphasize that our numerical theory results along with several of our insights generalize to broader empirical contexts using a nested fixed-point approach. One area where we expect our findings to generalize quite nicely is the empirical study of dynamic games. Researchers frequently solve dynamic games using iterative algorithms and, hence, we expect issues of loose tolerances to be problematic for applications nesting the solution of the game inside the estimation procedure. We also expect the speed advantages of MPEC to be important in this class of research.

References

[1] Ackerberg, D., J.. Geweke and J.. Hahn (2009): "Comments on 'Convergence Properties of the Likelihood of Computed Dynamic Models' by Fernandez-Villaverde, Rubio-Ramirez and Santos", *Econometrica*, Forthcoming.

[2] Andrews, D. W. K. (2002): "Generalized Method of Moments Estimation When a Parameter is on a Boundary," *Journal of Business and Economic Statistics,* 20 (4), 530–544.

[3] Bajari, P., J. T. Fox, K.-I. Kim and S. P. Ryan (2009): "The Random Coefficients Logit Model Is Identified," Working Paper, University of Chicago.

[4] Berry, S. (1994): "Estimating Discrete-Choice Models of Product Differentiation," *RAND Journal of Economics*, 25(2), 242–262.

[5] Berry, S. and P. A. Haile (2008): "Nonparametric Identification of Multinomial Choice Models with Heterogeneous Consumers and Endogeneity," Working Paper, Yale University.

[6] Berry, S. and P. A. Haile (2009): "Identification of Discrete Choice Demand From Market Level Data," Working Paper, Yale University.

[7] Berry, S., J. Levinsohn, and A. Pakes (1995): "Automobile Prices in Market Equilibrium," *Econometrica*, 63(4), 841–890.

[8] Berry, S., O. B. Linton, and A. Pakes (2004): "Limit Theorems for Estimating the Parameters of Differentiated Product Demand Systems," *Review of Economic Studies*, 71(3), 613–654.

[9] Berry, S. and A. Pakes (2007): "The Pure Characteristics Demand Model,"*International Economic Review*, 48(4), 1193–1225.

[10] Byrd, R. H., M. E. Hribar, and J. Nocedal (1999): "An Interior Point Method for Large Scale Nonlinear Programming." *SIAM Journal on Optimization*, 9(4), 877–990.

[11] Byrd, R. H., J. Nocedal and R. A. Waltz (1999): "KNITRO: An Integrated Package for Nonlinear Optimization," in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds,, 35–59. Springer.

[12] Carranza, J. E. (2008): "Product Innovation and Adoption in Market Equilibrium: The Case of Digital Cameras," Working Paper, University of Wisconsin.

[13] Dahlquist, G. and Å. Björck (2008): *Numerical Methods in Scientific Computing*. SIAM, Philadelphia, PA.

[14] Davis, P. J. (2006): "The Discrete Choice Analytically Flexible (DCAF) Model of Demand for Differentiated Products," CEPR Discussion Papers 5880.

[15] Dubé, J.-P., G. Hitsch and P. Chintagunta (2008): "Tipping and Concentration in Markets With Indirect Network Effects," Working Paper, University of Chicago.

[16] Fox, J. T. and A. Gandhi (2009): "Identifying Heterogeneity in Economic Choice Models," Working Paper, University of Chicago.

[17] Gandhi, A. (2008): "On the Nonparametric Foundations of Product Differentiated Demand Systems,"Working Paper, University of Wisconsin-Madison.

[18] Gill, P. E., W. Murray and M. H. Wright (1981): *Practical Optimization*, Academic Press, London.

[19] Gowrisankaran, G. and M. Rysman (2007): "Dynamics of Consumer Demand for New Durable Goods," Working Paper, The University of Arizona.

[20] Griewank, A. and G. F. Corliss, editors (1992): *Automatic Differentiation of Algorithms: Theory, Implementation, and Application.* SIAM, Philadelphia, PA.

[21] Hausman, J. A. and D. A. Wise (1976): "A Conditional Profit Model for Qualitative Choice: Discrete Decisions Recognizing Interdependence and Heterogeneous Preferences," *Econometrica*, 46(2), 403–426.

[22] Hendel, I. and A. Nevo (2007): "Measuring the Implications of Sales and Consumer Inventory Behavior," *Econometrica,* 74(16), 1637–1673.

[23] Jiang, R., P. Manchanda and P. E. Rossi (2009): "Bayesian Analysis of Random Coefficient Logit Models Using Aggregate Data," *Journal of Econometrics*, 149, 126–148.

[24] Judd, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58(2), 410–452.

[25] Judd, K. L. (1998): *Numerical Methods in Economics*. MIT Press, Cambridge, MA.

[26] Kelley, C. T. (1995): *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, PA.

[27] Kelley, C. T. (1999): *Iterative Methods for Optimization*, SIAM, Philadelphia, PA.

[28] Kelley, C. T. (2003): *Solving Nonlinear Equations with Newton's Method*. SIAM, Philadelphia, PA.

[29] Knittel, C. R. and K. Metaxoglou (2008): "Estimation of Random Coefficient Demand Models: Challenges, Difficulties and Warnings," Working Paper, U.C. Davis.

[30] Lee, R. S. (2008): "Vertical Integration and Exclusivity in Platform and Two-Sided Markets," Working Paper, New York University.

[31] McFadden, D. and K. Train (2000): "Mixed MNL Models for Discrete Response," *Journal of Applied Econometrics*, 15(5): 447–470.

[32] McKinnon, K. I. M. (1998): "Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point," *SIAM Journal on Optimization*, 9(1), 148–158.

[33] Melnikov, O. (2001): "Demand for Differentiated Durable Products: The Case of the U.S. Computer Printer Market," Working Paper, Yale University.

[34] Nair, H. (2007): "Intertemporal Price Discrimination with Forward-looking Consumers: Application to the US Market for Console Video-Games," *Quantitative Marketing and Economics*, 5(3), 239–292.

[35] Nemirovsky, A. S. and D. B. Yudin (1979 Russian, 1983): *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons.

[36] Nevo, A. (2000a): "Mergers with Differentiated Products: The Case of the Ready-to-Eat Cereal Industry," *RAND Journal of Economics*, 31(3), 395–421.

[37] Nevo, A. (2000b): "A Practitioner's Guide to Estimation of Random Coefficients Logit Models of Demand," *Journal of Economics and Management Strategy*, 9(4), 513–548.

[38] Nevo, A. (2001): "Measuring Market Power in the Ready-to-Eat Cereal Industry," *Econometrica*, 69(2), 307–342.

[39] Nocedal, J. and S. J. Wright (2006): *Numerical Optimization*. Springer, New York, NY.

[40] Petrin, A. (2002): "Quantifying the Benefits of New Products: The Case of the Minivan",*Journal of Political Economy*, 110, 705–729.

[41] Petrin, A. and K. Train (2009): "Control Function Corrections for Omitted Attributes in Differentiated Products Markets,"*Journal of Marketing Research*, Forthcoming.

[42] Rust, J. (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica,* 55(5), 999–1033.

[43] Schiraldi, P. (2008): "Automobile Replacement: a Dynamic Structural Approach," Working Paper, London School of Economics.

[44] Su, C.-L. and K. L. Judd (2008): "Constrained Optimization Approaches to Estimation of Structural Models," Working Paper, University of Chicago..

[45] Waltz, R. A. and T. D. Plantenga (2009): *KNITRO 6.0 Users's Manual.* Ziena Optimization, Inc.

[46] Wright, M. H. (1996): "Direct Search Method: Once Scorned, Now Respectable," in *Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis,* D. F. Griffiths and G. A. Watson, eds., Addison Wesley Longman, Harlow, UK, 191–208.

[47] Yang, Sha, Yuxin Chen and Greg M. Allenby (2003). "Bayesian Analysis of Simultaneous Demand and Supply," *Quantitative Marketing and Economics*, 1, 251-304.

A Proofs

In all the proofs below, we assume the sum of the second order and other higher order terms in a Taylor series expansion is bounded. This is a conventional assumption in the numerical optimization literature and allows us to use the big-O notation with a second order term, e.g., $O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right)$ or $O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right)$.

A.1 Proof of Theorem 3

By a Taylor series expansion of $Q(\xi)$ around $\xi(\theta,0)$, we have

$$\begin{aligned} & Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)-Q\left(\xi\left(\theta,0\right)\right) \\ = & \left[\frac{\partial Q(\xi)}{\partial \xi}\Big|_{\xi=\xi(\theta,0)}\right]'\left(\xi\left(\theta,\epsilon_{\text{in}}\right)-\xi\left(\theta,0\right)\right)+O\left(\left\|\xi\left(\theta,\epsilon_{\text{in}}\right)-\xi\left(\theta,0\right)\right\|^2\right) \\ & \text{and} \\ & \nabla_\theta Q\left(\xi\right)\big|_{\xi=\xi(\theta,\epsilon_{\text{in}})}-\nabla_\theta Q\left(\xi\right)\big|_{\xi=\xi(\theta,0)} \\ = & \left[\frac{\partial \nabla_\theta Q(\xi(\theta))}{\partial \xi}\Big|_{\xi=\xi(\theta,0)}\right]'\left(\xi\left(\theta,\epsilon_{\text{in}}\right)-\xi\left(\theta,0\right)\right)+O\left(\left\|\xi\left(\theta,\epsilon_{\text{in}}\right)-\xi\left(\theta,0\right)\right\|^2\right). \end{aligned}$$

Because $\left\|\xi\left(\theta,\epsilon_{\text{in}}\right)-\xi\left(\theta,0\right)\right\| \leq \frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}$ by Theorem 1, and assuming both $\left\|\frac{\partial Q(\xi)}{\partial \xi}\Big|_{\xi=\xi(\theta,0)}\right\|$ and $\left\|\frac{\partial \nabla_\theta Q(\xi(\theta))}{\partial \xi}\Big|_{\xi=\xi(\theta,0)}\right\|$ are bounded, we obtain

$$\begin{aligned} \left|Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)-Q\left(\xi\left(\theta,0\right)\right)\right| &= O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right) \\ \left\|\nabla_\theta Q\left(\xi\left(\theta\right)\right)\big|_{\xi=\xi(\theta,\epsilon_{\text{in}})}-\nabla_\theta Q\left(\xi\left(\theta\right)\right)\big|_{\xi=\xi(\theta,0)}\right\| &= O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right). \end{aligned}$$

A.2 Proof of Theorem 4

We define $\hat{\theta}(\epsilon_{\text{in}})$ to be the numerically incorrect estimates with the inner loop tolerance ϵ_{in},

$$\hat{\theta}(\epsilon_{\text{in}}) = \arg\max_{\theta}\left\{Q\left(\xi\left(\theta,\epsilon_{\text{in}}\right)\right)\right\}.$$

Because $\nabla_\theta Q\left(\xi\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}}\right)}=0$, the application of the second result in Theorem 3 at $\hat{\theta}(\epsilon_{\text{in}})$ gives

$$\left\|\nabla_\theta Q\left(\xi\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right\| = O\left(\frac{L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}{1-L\left(\hat{\theta}(\epsilon_{\text{in}})\right)}\epsilon_{\text{in}}\right). \tag{15}$$

Note that we have evaluated the GMM objective function with no numerical error at the point $\hat{\theta}(\epsilon_{\text{in}})$ that minimizes the GMM objective function with inner loop numerical error.

Let $\tilde{\theta}$ be any value of the structural parameters near $\hat{\theta}(\epsilon_{\text{in}})$. By first the inverse triangle inequality, then the regular triangle inequality, and then finally a Taylor series expansion, we have

$$\begin{aligned}
& \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},\epsilon_{\text{in}}\right)} \right\| - \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| \\
\leq & \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},\epsilon_{\text{in}}\right)} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| \\
= & \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},\epsilon_{\text{in}}\right)} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi(\tilde{\theta},0)} + \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi(\tilde{\theta},0)} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| \\
\\
\leq & \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},\epsilon_{\text{in}}\right)} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},0\right)} \right\| \\
& + \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},0\right)} - \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| \\
\leq & O\left(\tfrac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + \left\| \nabla_\theta^2 Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| \left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\| + O\left(\left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\|^2 \right).
\end{aligned}$$

As we have assumed $\left\| \nabla_\theta^2 Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)} \right\|$ is bounded, the second order term $O\left(\left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\|^2 \right)$ term can be ignored. By rearranging the above inequality, we obtain

$$\begin{aligned}
& \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\tilde{\theta},\epsilon_{\text{in}}\right)} \right\| \\
\leq & \left\| \nabla_\theta Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)} \right\| + O\left(\tfrac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + O\left(\left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\| \right) \\
= & O\left(\tfrac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))}\epsilon_{\text{in}}\right) + O\left(\tfrac{L(\tilde{\theta})}{1-L(\tilde{\theta})}\epsilon_{\text{in}}\right) + O\left(\left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\| \right) \\
= & O\left(\epsilon_{\text{in}}\right) + O\left(\left\| \tilde{\theta} - \hat{\theta}(\epsilon_{\text{in}}) \right\| \right),
\end{aligned}$$

where the first equality uses (15).

A.3 Proof of Theorem 5

We define θ^* to be the true estimate when there are no inner loop numerical errors ($\epsilon_{\text{in}} = 0$), i.e., $\theta^* = \arg\max_\theta \left\{ Q\left(\xi\left(\theta, 0\right)\right) \right\}$. First, we can quantify the bias between the numerically correct and incorrect objective function values, $Q\left(\xi(\hat{\theta}\left(\epsilon_{\text{in}}\right), \epsilon_{\text{in}}\right)$ and $Q\left(\xi(\theta^*, 0)\right)$. By two Taylor series expansions, we have

$$
\begin{aligned}
& Q\left(\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),\epsilon_{\text{in}}\right)\right)-Q\left(\xi(\theta^*,0)\right)\\
= \;& Q\left(\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),\epsilon_{\text{in}}\right)\right)-Q\left(\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),0\right)\right)+Q\left(\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),0\right)\right)-Q\left(\xi(\theta^*,0)\right)\\
= \;& \left[\nabla_\xi Q\left(\xi\left(\theta\right)\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right]'\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right)\\
& +O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right)\\
& +\left[\left(\nabla_\theta\xi\left(\theta\right)\right)'\nabla_\xi Q\left(\xi\right)\Big|_{\xi=\xi(\theta^*,0)}\right]'\left(\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right)+O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right)\\
= \;& \left[\nabla_\xi Q\left(\xi\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right]'\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right)\\
& +O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right)+O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right),
\end{aligned}
$$

because $\nabla\xi(\theta^*)'\nabla_\xi Q\left(\xi(\theta^*)\right)=0$ at the true estimates θ^*.

Rearranging the equality involving $Q\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})\right)-Q\left(\xi(\theta^*,0)\right)$ to focus on the $O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right)$ term, we have

$$
\begin{aligned}
& O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right)\\
= \;& Q\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})\right)-Q\left(\xi(\theta^*,0)\right)\\
& -\left[\nabla_\xi Q\left(\xi\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right]'\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right)\\
& -O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right)\\
\leq \;& \left|Q\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})\right)-Q\left(\xi(\theta^*,0)\right)\right|\\
& +\left\|\nabla_\xi Q\left(\xi\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right\|\left\|\xi\left(\hat{\theta}\left(\epsilon_{\text{in}}\right),\epsilon_{\text{in}}\right)-\xi\left(\hat{\theta}(\epsilon_{\text{in}}),0\right)\right\|\\
& -O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right).
\end{aligned}
$$

Because we assume $\left\|\nabla_\xi Q\left(\xi\right)\Big|_{\xi=\xi(\hat{\theta}(\epsilon_{\text{in}}),0)}\right\|$ is bounded, the second-order term $O\left(\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|^2\right)$ can be ignored. This allows us to focus on the numerical error from the NFP algorithm's inner loop and the bias in objective values. From Theorem 1, we also know $\left\|\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})-\xi(\hat{\theta}(\epsilon_{\text{in}}),0)\right\|\leq \frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))}\epsilon_{\text{in}}$. Hence, we obtain

$$
O\left(\left\|\hat{\theta}(\epsilon_{\text{in}})-\theta^*\right\|^2\right)\leq\left|Q\left(\xi(\hat{\theta}(\epsilon_{\text{in}}),\epsilon_{\text{in}})\right)-Q\left(\xi(\theta^*,0)\right)\right|+O\left(\frac{L(\hat{\theta}(\epsilon_{\text{in}}))}{1-L(\hat{\theta}(\epsilon_{\text{in}}))}\epsilon_{\text{in}}\right).
$$

A.4 Proof of Theorem 6

The NFP method (4) solves the following unconstrained problem

$$\min_{\theta} Q\left(\xi\left(\theta\right)\right). \tag{16}$$

The first-order condition of (16) is

$$\frac{\partial Q\left(\xi\left(\theta\right)\right)}{\partial\theta} = \frac{d\xi}{d\theta}'\frac{\partial Q}{\partial\xi} = 0. \tag{17}$$

The constrained optimization formulation of (16) is

$$\begin{array}{ll} \min\limits_{(\theta,\xi)} & Q\left(\xi\right) \\ \text{s.t.} & s(\xi;\theta) = S. \end{array} \tag{18}$$

The Lagrangian for (18) is $\mathcal{L}\left(\theta,\xi,\lambda\right) = Q(\xi) + \lambda^{T}\left(S - s(\xi;\theta)\right)$, where λ is the vector of Lagrange multipliers. The first-order conditions of (18) are

$$\begin{array}{rcl} \dfrac{\partial\mathcal{L}\left(\theta,\xi,\lambda\right)}{\partial\theta} & = & -\dfrac{ds(\xi;\theta)}{d\theta}'\lambda = 0 \\ \dfrac{\partial\mathcal{L}\left(\theta,\xi,\lambda\right)}{\partial\xi} & = & \dfrac{\partial Q}{\partial\xi} - \dfrac{ds(\xi;\theta)}{d\xi}'\lambda = 0 \\ \dfrac{\partial\mathcal{L}\left(\theta,\xi,\lambda\right)}{\partial\lambda} & = & S - s(\xi;\theta) = 0. \end{array} \tag{19}$$

Because the NFP inner loop is a contraction mapping, the matrix $\frac{ds(\xi;\theta)}{d\xi}'$ is invertible.[19] Solving the second set of first order conditions for λ gives $\lambda = \left(\frac{ds(\xi;\theta)}{d\xi}'\right)^{-1}\frac{\partial Q}{\partial\xi}$. Then

$$\frac{\partial\mathcal{L}}{\partial\theta} = -\frac{ds(\xi;\theta)}{d\theta}'\left(\frac{ds(\xi;\theta)}{d\xi}'\right)^{-1}\frac{\partial Q}{\partial\xi} = 0, \tag{20}$$

which is identical to (17), the first-order condition from the NFP formulation. To see the equivalence, note that the implicit function theorem (Theorem M.E.1 in Mas-Collel, Whinston and Green 1995) states

$$\frac{\partial\xi\left(\theta\right)}{\partial\theta} = -\left(\frac{ds(\xi;\theta)}{d\xi}\right)^{-1}\frac{ds(\xi;\theta)}{d\theta},$$

so by substitution, the two FOC's are identical.

[19] We thank Ken Judd and John Birge for pointing out this property.

B Gradients for the MPEC Objective Function and Constraints

Here we derive the gradients of the MPEC objective function and constraints with respect to the optimization parameters in MPEC. These gradients are an important input, for both numerical accuracy and speed. Nevo (2000b) lists the gradients for NFP. This section uses the independent normal distribution for each of the random coefficients, as in BLP (1995) and many other empirical papers.

Market Share

$$\begin{aligned} s_j(\xi_t;\theta) &= \int \frac{\exp\left(x'_{j,t}\overline{\beta}-\bar{\alpha}p_{j,t}+\xi_{j,t}+\sum_k x'_{k,j,t}\nu_k\sigma_{\beta_k}-p_{j,t}\nu_{K+1}\sigma_\alpha\right)}{1+\sum_{i=1}^{J}\exp\left(x'_{i,t}\overline{\beta}-\bar{\alpha}p_{i,t}+\xi_{i,t}+\sum_k x'_{k,i,t}\nu_k\sigma_{\beta_k}-p_{i,t}\nu_{K+1}\sigma_\alpha\right)} dF(\nu) \\ &= \int T_j(\xi_t,\nu;\theta)dF(\nu) \end{aligned}$$

$$\text{where } \theta = (\overline{\beta},\overline{\alpha},\sigma_\beta,\sigma_\alpha)', \text{ and } \nu \sim N(0, I_{K+1}).$$

MPEC Criterion Function

$$\min_{\theta,\xi} g(\xi)' W g(\xi)$$
$$\text{subject to } s(\xi;\theta) = S,$$

$$\text{where } g(\xi) = \frac{1}{T}\sum_{t=1}^{T}\xi'_t z_t.$$

Gradients for MPEC

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \overline{\beta_k}} = \int T_j(\xi_t,\nu;\theta)(x_{j,k,t} - \sum_i T_i(\xi_t,\nu;\theta)x_{k,i,t})dF(\nu)$$

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \overline{\alpha}} = \int T_j(\xi_t,\nu;\theta)(p_{j,k,t} - \sum_i T_i(\xi_t,\nu;\theta)p_{k,i,t})dF(\nu)$$

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \sigma_{\beta_k}} = \int T_j(\xi_t,\nu;\theta)(x_{j,k,t} - \sum_i T_i(\xi_t,\nu;\theta)x_{k,i,t})\nu_k dF(\nu)$$

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \sigma_\alpha} = \int T_j(\xi_t,\nu;\theta)(p_{j,k,t} - \sum_i T_i(\xi_t,\nu;\theta)p_{k,i,t})\nu_{K+1} dF(\nu)$$

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \xi_{j,t}} = \int T_j(\xi_t,\nu;\theta)(1 - T_j(\xi_t,\nu;\theta))dF(\nu)$$

$$\frac{\partial s_j(\xi_t;\theta)}{\partial \xi_{i,t}} = -\int T_j(\xi_t,\nu;\theta))T_i(\xi_t,\nu;\theta))dF(\nu)$$

$$\frac{\partial g(\xi)' W g(\xi)}{\partial \xi} = 2g(\xi)' W \frac{\partial g(\xi)}{\partial \xi}$$

C Extension: Maximum Likelihood Estimation

In this section, we outline how a researcher would adapt MPEC to a likelihood-based estimation of random-coefficients-logit demand. Some researchers prefer to work with likelihood-based estimators and, more specifically, with Bayesian MCMC estimators (Yang et al 2003 and Jiang et al. 2008) based on the joint density of observed prices and market shares.[20] Besides efficiency advantages, the ability to evaluate the likelihood of the data could be useful for testing purposes. The trade-off relative to GMM is the need for additional modeling structure which, if incorrect, could lead to biased parameter estimates. Like GMM, the calculation of the density of market shares still requires inverting the system of market share equations. Once again, MPEC can be used to circumvent the need for inverting the shares, thereby offsetting a layer of computational complexity and a potential source of numerical error. Below we outline the estimation of a limited information approach that models the data-generating process for prices in a "reduced form" (this motivation is informal as we do not specify a supply-side model and solve for a reduced form). However, one can easily adapt the estimator to accommodate a structural (full-information) approach that models the data-generating process for supply-side variables, namely prices, as the outcome of an equilibrium to a game of imperfect competition (assuming the equilibrium exists and is unique).

Recall that the system of market shares is defined in (2). We assume, as in a triangular system, that the data-generating process for prices is

$$p_{j,t} = z'_{j,t}\gamma + \eta_{j,t}, \tag{21}$$

where $z_{j,t}$ is a vector of price-shifting variables and $\eta_{j,t}$ is a mean-zero, i.i.d. shock. To capture the potential endogeneity in prices, we assume the supply and demand shocks have the following joint distribution: $(\xi_{j,t}, \eta_{j,t})' \equiv u_{j,t} \sim N(0, \Omega)$ where $\Omega = \begin{bmatrix} \sigma_\xi^2 & \sigma_{\xi,\eta} \\ \sigma_{\xi,\eta} & \sigma_\eta^2 \end{bmatrix}$. Let $\rho = \frac{\sigma_{\xi,\eta}}{\sigma_\xi \sigma_\eta}$.

The system defined by equations (2) and (21) has the joint density function

$$f_{s,p}\left(s_t, p_t; \Theta\right) = f_{\xi|\eta}\left(s_t \mid x_t, p_t; \theta, \Omega\right) \left|J_{\xi \to s}\right| f_\eta\left(p_t \mid z_t; \gamma, \Omega\right),$$

[20] One can also think of Jiang et al. (2008) as an alternative algorithm for finding the parameters. The MCMC approach is a stochastic search algorithm that might perform well if the BLP model produces many local optima because MCMC will not be as likely to get stuck on a local flat region. Because our goal is not to study the role of multiple local minima, we do not explore the properties of a Bayesian MCMC algorithm.

where $\Theta = \left(\theta, \gamma, \sigma_\xi^2, \sigma_{\xi,\eta}, \sigma_\eta^2\right)$ is the vector of model parameters, $f_{\xi|\eta}(\cdot|\cdot)$ is the marginal density of ξ conditional on η, $f_\eta(\cdot|\cdot)$ is a Gaussian density with variance σ_η^2, and $J_{\xi\to s}$ is the Jacobian matrix corresponding to the transformation of variables of $\xi_{j,t}$ to shares. The density of $\xi_{j,t}$ conditional on $\eta_{j,t}$ is

$$f_{\xi|\eta}\left(s_t \mid x_t, p_t; \theta, \Omega\right) = \prod_{j=1}^{J} \frac{1}{\sqrt{2\pi}\sigma_\xi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2}\frac{\left(\xi_{j,t} - \rho\frac{\sigma_\xi}{\sigma_\eta}\eta_{j,t}\right)^2}{\sigma_\xi^2\left(1-\rho^2\right)}\right).$$

Note that the evaluation of $\xi_{j,t}$ requires inverting the market share equations, (2).

The element $J_{j,k}$ in row l and column k of the Jacobian matrix, $J_{\xi\to s}$, is

$$J_{j,l} = \begin{cases} \int_\beta \left(1 - \frac{\exp\left(\beta^0 + x'_{j,t}\beta^x - \beta^p p_{j,t} + \xi_{j,t}\right)}{1+\sum_{k=1}^{J}\exp\left(\beta^0 + x'_{k,t}\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)}\right) \frac{\exp\left(\beta^0 + x'_{j,t}\beta^x - \beta^p p_{j,t} + \xi_{j,t}\right)}{1+\sum_{k=1}^{J}\exp\left(\beta^0 + x'_{k,t}\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)} dF_\beta\left(\beta;\theta\right) & , \quad j = l \\ -\int_\beta \frac{\exp\left(\beta^0 + x'_{j,t}\beta^x - \beta^p p_{j,t} + \xi_{j,t}\right)}{1+\sum_{k=1}^{J}\exp\left(\beta^0 + x'_{k,t}\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)} \frac{\exp\left(\beta^0 + x'_{l,t}\beta^x - \beta^p p_{l,t} + \xi_{l,t}\right)}{1+\sum_{k=1}^{J}\exp\left(\beta^0 + x'_{k,t}\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)} dF_\beta\left(\beta;\theta\right) & , \quad j \neq l \end{cases}.$$

Standard maximum likelihood estimation would involve searching for parameters, Θ^{LISML}, that maximize the following log-likelihood function

$$l\left(\Theta\right) = \sum_{t=1}^{T} \log\left(f_{s,p}\left(s_t, p_t; \Theta\right)\right).$$

This would consist of a nested inner loop to compute the demand shocks, $\xi_{j,t}$, via numerical inversion (the NFP contraction mapping).

The equivalent MPEC approach entails searching for the vector of parameters (Θ, ξ) that maximizes the constrained optimization problem

$$\begin{aligned} \max l\left(\Theta, \xi\right) \quad &= \quad \textstyle\sum_{t=1}^{T} \log\left(f_{\xi|\eta}\left(s_t \mid x_t, p_t; \theta, \Omega\right)\left|J_{\xi\to s}\right| f_\eta\left(p_t \mid z_t; \gamma, \Omega\right)\right) \\ \text{subject to} \quad &\quad s(\xi;\theta) = S. \end{aligned}$$

D Chebyshev Approximation of the Expected Value of Waiting

First, we bound the range of prices as follows, $p = \left(p_1, p_2\right)' \in [0, b] \times [0, b]$, where b is large (b is 1.5 times the largest observed price in the data). We then approximate the expected value of delaying adoption with Chebyshev polynomials, $v_0^r\left(p; \theta^r\right) \approx \gamma^{r'}\Lambda(p)$, where γ^r is a $K \times 1$ vector of parameters and $\Lambda\left(p\right)$ is a $K \times 1$ vector of K Chebyshev polynomials. Therefore, we

can rewrite the Bellman equation as

$$\gamma^{r'}\Lambda(p) = \delta \int \log\left(\exp\left(\gamma^{r'}\Lambda(p\rho+\psi)\right) + \sum_j \exp\left(\beta_j^r - \alpha^r\left(p'\rho_j+\psi\right)+\xi_j\right)\right) dF_{\psi,\xi}(\psi,\xi).$$

To solve for the Chebyshev weights, we use the Galerkin method described in Judd (1992). We define the residual function

$$\begin{aligned} R(p;\gamma) &= \gamma^{r'}\Lambda(p) - \ldots \\ &\quad \delta\int \log\left(\exp\left(\gamma^{r'}\Lambda(p\rho+\psi)\right) + \sum_j \exp\left(\beta_j^r - \alpha^r(p'\rho_j+\psi)+\xi_j\right)\right) dF_{\psi,\xi}(\psi,\xi) \end{aligned}. \tag{22}$$

Next, we let X be the matrix of K Chebyshev polynomials at each of the G points on our grid (i.e. G nodes). Our goal is to search for parameters, γ, that set the following expression to zero:

$$X'R(p;\gamma) = 0.$$

We use an iterated least squares approach for NFP.

1. Pick a starting value $\gamma^{r,0}$, $v_0^{r,0}(p;\Theta^r) = \gamma^{r,0'}\rho(p)$
2. Compute

   $$Y\left(p;\gamma^{r,0}\right) = \delta\int \log\left(\exp\left(\gamma^{r,0'}\Lambda(p\rho+\psi)\right) + \sum_j \exp\left(\beta_j^r - \alpha^r\left(p'\rho_j+\psi\right)+\xi_j\right)\right) dF_{\psi,\xi}(\psi,\xi)$$

   using quadrature
3. solve the least squares problem: $\min_\gamma R(p;\gamma)'R(p;\gamma)$ or

   $$\min_\gamma \left(X\gamma^r - Y\left(p;\gamma^{r,0}\right)\right)'\left(X\gamma^r - Y\left(p;\gamma^{r,0}\right)\right)$$

   - for which the solution is: $\gamma^{r,1} = \left(X'X\right)^{-1}X'Y\left(p;\gamma^{r,0}\right)$.
4. Compute $v_0^{r,1}(p;\Theta^r) = \gamma^{r,1'}\Lambda(p)$
5. Repeat steps 2 and 3 until convergence.

E Jacobian of the Density of (p_t, S_t) in the Dynamic BLP model

The Jacobian is defined as follows:

$$J_{t,u\to Y} = \begin{bmatrix} \frac{\partial \psi_t}{\partial p_t} & \frac{\partial \psi_t}{\partial S_t} \\ \frac{\partial \xi_t}{\partial p_t} & \frac{\partial \xi_t}{\partial S_t} \end{bmatrix}.$$

Since $\frac{\partial \psi_t}{\partial \log(p_t)} = I_J$ and $\frac{\partial \psi_t}{\partial \log(p_t)} = 0_J$ (a square matrix of zeros), we only need to compute the matrix of derivatives, $\left[\frac{\partial \xi_t}{\partial S_t}\right]$. We can simplify this calculation by applying the implicit function theorem to the following system

$$G(S_t, \xi_t) = s(p, \xi_t; \theta) - S_t = 0$$

and computing the lower block of the Jacobian as

$$\begin{aligned} J_{t,\xi\to S} &= -\left[\frac{\partial G}{\partial \xi_t}\right]^{-1}\left[\frac{\partial G}{\partial S_t}\right] \\ &= \left[\frac{\partial s}{\partial \xi_t}\right]^{-1} \end{aligned},$$

where the (j,k) element of $\frac{\partial s_{j,t}}{\partial \xi_{k,t}}$ is

$$\frac{\partial S_{j,t}}{\partial \xi_{k,t}} = \begin{cases} \sum_r \lambda_{r,t} s_j(p_t, \xi_t; \theta^r)\left(1 - s_j(p_t, \xi_t; \theta^r)\right) & \text{, if } j = k \\ -\sum_r \lambda_{r,t} s_j(p, \xi_t; \theta^r)\, s_k(p, \xi_t; \theta^r) & \text{, otherwise.} \end{cases}$$

F Monte Carlo: Varying the Quality of the Data

In principle, the quality of the data could influence the convexity of the objective function and, hence, the trajectory of the outer loop search. To assess the role of "data quality," we construct a set of sampling experiments that manipulate the power of the instruments (i.e. the correlation between the prices, p, and the instruments, z). Let $z_{j,t,d} = \tilde{u}_{j,t,d} + \nu u_{j,t}$, where $u_{j,t}$ is a random shock that also affects price and ν is a measure of the power of the instruments. Higher ν's result in more powerful instruments. By generating the prices before the instruments, we ensure that prices, shares, and product characteristics are unaffected by the power of the instruments. Thus, the NFP inner loop and the MPEC market share constraints are identical when ν varies. Only the instruments in the moment conditions vary. We use the identity matrix for the GMM weighting matrix to avoid the instrument power affecting the choice of weighting matrix.

Table 10 lists the results from five runs with differing levels of instrument power. The table lists the value of ν and the resulting R^2 from a regression of price on all excluded-from-demand

and non-excluded instruments. We see that R^2 decreases as the power of the instruments decreases. In all specifications, MPEC is faster than NFP. MPEC's speed decreases with instrument power, although the decrease from 118 to 159 seconds is not large compared to the total run time of NFP, which ranges from 342 to 619 seconds. We have no theoretical explanation for the pattern relating NFP's speed and instrument power. To some extent, the pattern is driven by the fact that NFP encounters convergence problems as we increase the power of the instruments. When $\nu = 0.5$, only 50% (10 out of 20) of the replications fail to converge for all five starting values, for NFP. Naturally, this convergence problem could be a practical concern if the researcher mistakenly interprets the point at which the solver stops as a local minimum even though the gradient-based solver is unable to detect convergence.

Table 1: Three NFP Implementations: Varying Starting Values for One synthetic Dataset, with Numerical Derivatives

	NFP Loose Inner	NFP Loose Both	NFP Tight	Truth
Fraction Convergence	0.0	0.54	0.95	
Frac.$<1\%>$ "Global" Min.	0.0	0.0	1.00	
Mean Own Price Elasticity	-7.24	-7.49	-5.77	-5.68
Std. Dev. Own Price Elasticity	5.48	5.55	~0	
Lowest Objective	0.0176	0.0198	0.0169	
Elasticity for Lowest Obj.	-5.76	-5.73	-5.77	-5.68

We use 100 starting values for one synthetic dataset. The NFP loose inner loop implementation has $\epsilon_{\text{in}} = 10^{-4}$ and $\epsilon_{\text{out}} = 10^{-6}$. The NFP loose-both implementation has $\epsilon_{\text{in}} = 10^{-4}$ and $\epsilon_{\text{out}} = 10^{-2}$. The NFP-tight implementation has $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. We use numerical derivatives using KNITRO's built-in procedures.

Table 2: Three NFP Implementations: Varying Starting Values for Nevo's Cereal Dataset, with Closed-Form Derivatives

	NFP Loose Inner	NFP Loose Both	NFP Tight	NFP Tight Simplex
Fraction Reported Convergence	0.0	0.76	1.00	1.00
Frac. Obj. Fun. $< 1\%$ Greater than "Global" Min.	0.0	0.0	1.00	0.0
Mean Own Price Elasticity Across All Runs	-3.82	-3.69	-7.43	-3.84
Std. Dev. Own Price Elasticity Across All Runs	0.4	0.07	~0	0.35
Lowest Objective Function Value	0.00213	0.00683	0.00202	0.00683
Elasticity for Run with Lowest Obj. Value	-6.71	-3.78	-7.43	-3.76

We use the same 50 starting values for each implementation. The NFP loose inner loop implementation has $\epsilon_{\text{in}} = 10^{-4}$ and $\epsilon_{\text{out}} = 10^{-6}$. The NFP loose both implementation has $\epsilon_{\text{in}} = 10^{-4}$ and $\epsilon_{\text{out}} = 10^{-2}$. The NFP tight implementation has $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. The Nelder-Meade or simplex method uses a tighter inner loop tolerance of $\epsilon_{\text{in}} = 10^{-14}$ and MATLAB's default values for the simplex convergence criteria. We manually code closed-form derivatives for all methods other than for Nelder-Meade, which does not use derivative information.

Table 3: Lipschitz Constants for the NFP Algorithm

Parameter Scale		Std. Dev. of Shocks ξ		# of Markets T		Mean of Intercept $E\left[\beta_i^0\right]$	
Altered Value	Mean Lipschitz	Altered Value	Mean Lipschitz	Altered Value	Mean Lipschitz	Altered Value	Mean Lipschitz
0.01	0.985	0.1	0.808	25	0.860	-2	0.771
0.1	0.971	0.25	0.813	50	0.871	-1	0.871
0.50	0.887	0.5	0.832	100	0.888	0	0.936
0.75	0.865	1	0.871	200	0.888	1	0.971
1	0.871	2	0.934			2	0.988
1.5	0.911	5	0.972			3	0.996
2	0.938	20	0.984			4	0.998
3	0.970						
5	0.993						

Table 4: Monte Carlo Results Varying the Lipschitz Constant

Intercept $E\left[\beta_i^0\right]$	Lipschitz Constant	Implementation	Runs Converged (fraction)	CPU Time (s)	Elasticities			Outside Share
					Bias	RMSE	Value	
-1.9	0.789	NFP tight	1	1012.9	-0.200	0.265	-12.00	0.900
		MPEC	1	981.0	-0.200	0.265	-12.00	0.900
-0.9	0.858	NFP tight	1	1365.9	-0.203	0.266	-11.98	0.845
		MPEC	1	1015.2	-0.203	0.266	-11.98	0.845
0.1	0.913	NFP tight	1	1608.4	-0.205	0.266	-11.97	0.775
		MPEC	1	1001.4	-0.205	0.266	-11.97	0.775
1.1	0.952	NFP tight	1	2057.7	-0.201	0.256	-11.96	0.687
		MPEC	1	832.4	-0.201	0.256	-11.96	0.687
2.1	0.976	NFP tight	1	2544.8	-0.202	0.256	-11.95	0.583
		MPEC	1	810.2	-0.199	0.254	-11.96	0.583
3.1	0.989	NFP tight	1	3730.3	-0.195	0.252	-11.97	0.472
		MPEC	1	767.5	-0.202	0.254	-11.96	0.472

There are 20 replications for each experiment. Each replication uses five starting values to do a better job at finding a global minimum. The NFP-tight implementation has $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The same 100 simulation draws are used to generate the data and to estimate the model.

Table 5: Monte Carlo Results Varying the Number of Markets

# Markets T	Lipschitz Constant	Implementation	Runs Converged (fraction)	CPU Time (s)	Elasticities			Outside Share
					Bias	RMSE	Value	
25	0.903	NFP tight	1	1372.9	-0.265	0.385	-12.16	0.640
		MPEC	1	555.2	-0.269	0.389	-12.16	0.640
50	0.952	NFP tight	1	2060.6	-0.201	0.256	-11.96	0.687
		MPEC	1	839.0	-0.201	0.256	-11.96	0.687
100	0.956	NFP tight	1	8068.2	-0.092	0.174	-12.30	0.893
		MPEC	1	2143.6	-0.106	0.225	-12.29	0.893

There are 20 replications for each experiment. Each replication uses five starting values to do a better job at finding a global minimum . The NFP-tight implementation has $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The same 100 simulation draws are used to generate the data and to estimate the model.

Table 6: Last iterations of runs using the Nevo dataset, for one starting value

NFP		MPEC		
Iteration	Optimality Error	Iteration	Optimality Error	Feasibility Error
115	3.5×10^{-4}	19	3.1×10^{-1}	8.6×10^{-2}
116	3.9×10^{-5}	20	3.0×10^{-2}	5.6×10^{-3}
117	3.8×10^{-4}	21	5.4×10^{-3}	3.0×10^{-4}
118	2.9×10^{-5}	22	3.3×10^{-5}	7.4×10^{-6}
119	3.9×10^{-5}	23	1.6×10^{-8}	9.1×10^{-8}
120	5.9×10^{-6}			
121	1.8×10^{-5}			
122	1.7×10^{-6}			
123	9.7×10^{-7}			

Table 7: Monte Carlo Results for Dynamic BLP with One Consumer Type for $\delta = 0.96$: NFP versus MPEC

	MPEC		NFP		
Speeds	335.55 secs.		553.50 secs.		
Parameters	Mean	RMSE	Mean	RMSE	Truth
Utility intercept product 1	3.9557	0.1780	3.9556	0.1780	4.0000
Utility intercept product 2	2.9572	0.2015	2.9572	0.2015	3.0000
Utility price coefficient, type 1	-1.0030	0.0101	-1.0030	0.0101	-1.0000
Price, product 1, constant	0.2111	0.0345	0.2111	0.0345	0.2000
Price, product 1, lagged price of product 1	0.7962	0.0136	0.7962	0.0136	0.8000
Price, product 1, lagged price of product 2	0.0026	0.0098	0.0026	0.0098	0.0000
Price, product 2, constant	0.2071	0.0378	0.2071	0.0378	0.2000
Price, product 2, lagged price of product 1	0.0037	0.0168	0.0037	0.0168	0.0000
Price, product 2, lagged price of product 2	0.7935	0.0156	0.7935	0.0156	0.8000
Demand shocks, Cholesky variance term	0.9958	0.0173	0.9958	0.0173	1.0000
Covariance btw supply and demand, Cholesky variance term	0.5015	0.0215	0.5015	0.0215	0.5000
Supply shocks, Cholesky variance term	0.8647	0.0152	0.8647	0.0152	0.8660
% of replications routine reports convergence	100%		100%		

There are 20 replications for each of MPEC and NFP. The same synthetic data are used for both MPEC and NFP. Each replication uses five starting values to do a better job at finding a global minimum. The NFP implementation has $\epsilon_{\text{in}}^{\xi} = 10^{-14}$, $\epsilon_{\text{in}}^{V} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The data have $T = 50$ periods and $M = 20$ distinct markets. Each market has two competing products. The Chebyshev regression approximation to the value function uses a fourth-order polynomial and five interpolation nodes. The numerical integration of future states uses Gauss-Hermite quadrature with three nodes. NFP uses numerical derivatives, as coding the derivatives of dynamic BLP is infeasible for many problems and it is not clear automatic differentiation works with nested inner loops. MPEC uses automatic differentiation in the form of the package MAD. The percentage of replications where the routine reports convergence is the fraction of the 20 replication where the lowest objective function coincided with an exit flag of 0 from KNITRO.

Table 8: Monte Carlo Results for Dynamic BLP with One Consumer Type for $\delta = 0.99$: NFP versus MPEC

	MPEC		NFP		
Speeds	671.49 secs.		1295.50 secs.		
Parameters	Mean	RMSE	Mean	RMSE	Truth
Utility intercept product 1	4.0684	0.7235	3.3907	1.7473	4.0000
Utility intercept product 2	3.0692	0.7316	2.3913	1.7844	3.0000
Utility price coefficient, type 1	-0.9885	0.0380	-0.9987	0.0152	-1.0000
Price, product 1, constant	0.1929	0.0682	0.2171	0.0655	0.2000
Price, product 1, lagged price of product 1	0.8170	0.0532	0.8022	0.0546	0.8000
Price, product 1, lagged price of product 2	-0.0044	0.0295	0.0000	0.0520	0.0000
Price, product 2, constant	0.1770	0.1102	0.2058	0.0813	0.2000
Price, product 2, lagged price of product 1	-0.0195	0.0557	-0.0065	0.0436	0.0000
Price, product 2, lagged price of product 2	0.8330	0.0860	0.8089	0.0585	0.8000
Demand shocks, Cholesky variance term	1.0139	0.0468	1.0053	0.0334	1.0000
Covariance btw supply and demand, Cholesky variance term	0.4985	0.0255	0.5050	0.0219	0.5000
Supply shocks, Cholesky variance term	0.8652	0.0152	0.8640	0.0159	0.8660
% of replications routine reports convergence	100%		90%		

There are 20 replications for each of MPEC and NFP. The same synthetic data are used for both MPEC and NFP. Each replication uses five starting values to do a better job at finding a global minimum. The NFP implementation has $\epsilon_{\text{in}}^{\xi} = 10^{-14}$, $\epsilon_{\text{in}}^{V} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The data have $T = 50$ periods and $M = 20$ distinct markets. Each market has two competing products. The Chebyshev regression approximation to the value function uses a fourth-order polynomial and four interpolation nodes. The numerical integration of future states uses Gauss-Hermite quadrature with three nodes. NFP uses numerical derivatives, as coding the derivatives of dynamic BLP is infeasible for many problems and it is not clear automatic differentiation works with nested inner loops. MPEC uses automatic differentiation in the form of the package MAD. The percentage of replications where the routine reports convergence is the fraction of the 20 replication where the lowest objective function coincided with an exit flag of 0 from KNITRO.

Table 9: Monte Carlo Results for Dynamic BLP with Two Consumer Types and $\delta = 0.90$ MPEC Only

	MPEC		
Speeds	9397 secs.		
Parameters	Mean	RMSE	Truth
Utility intercept product 1	3.6604	0.5719	4.0000
Utility intercept product 2	2.6980	0.5287	3.0000
Utility price coefficient, type 1	-1.0159	0.0299	-1.0000
Utility price coefficient, type 2	-2.0369	0.2623	-2.0000
Frequency, type 1	0.7907	0.1016	0.7000
Price, product 1, constant	0.1894	0.0818	0.2000
Price, product 1, lagged price of product 1	0.7919	0.0333	0.8000
Price, product 1, lagged price of product 2	-0.0013	0.0274	0.0000
Price, product 2, constant	0.2283	0.0929	0.2000
Price, product 2, lagged price of product 1	-0.0013	0.0262	0.0000
Price, product 2, lagged price of product 2	0.7919	0.0341	0.8000
Demand shocks, Cholesky variance term	0.9001	0.4386	1.0000
Covariance btw supply and demand, Cholesky variance term	0.4537	0.4609	0.5000
Supply shocks, Cholesky variance term	0.6891	0.5616	0.8660
% of replications routine reports convergence	100%		

There are 20 replications. Each replication uses two starting values to do a better job at finding a global minimum. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The data have $T = 50$ periods and $M = 5$ distinct markets. Each market has two competing products. The Chebyshev regression approximation to the value function uses a fourth-order polynomial and four interpolation nodes. The numerical integration of future states uses Gauss-Hermite quadrature with three nodes. The code uses automatic differentiation in the form of the package MAD.

Table 10: Monte Carlo Results Varying the Data Quality: The Power of Instruments

Instrument Power (ν)	R^2	Implementation	Runs Converged (fraction)	CPU Time (s)	Elasticities		
					Bias	RMSE	Value
1/2	0.75	NFP tight	0.50	619	0.028	0.173	-8.16
		MPEC	1	118	0.023	0.173	-8.16
1/4	0.69	NFP tight	1	342	0.058	0.140	-8.16
		MPEC	1	129	0.058	0.140	-8.16
1/6	0.62	NFP tight	1	447	-0.020	0.158	-8.15
		MPEC	1	135	-0.020	0.158	-8.15
1/8	0.57	NFP tight	1	376	-0.022	0.186	-8.13
		MPEC	1	135	-0.022	0.186	-8.13
1/16	0.46	NFP tight	1	512	-0.111	0.312	-8.07
		MPEC	1	159	-0.090	0.323	-8.05

There are 20 replications for each experiment. Each replication uses ten starting values to ensure a global minimum is found. The NFP-tight implementation has $\epsilon_{\text{in}} = 10^{-14}$ and $\epsilon_{\text{out}} = 10^{-6}$. There is no inner loop in MPEC; $\epsilon_{\text{out}} = 10^{-6}$ and $\epsilon_{\text{feasible}} = 10^{-6}$. The same 100 simulation draws are used to generate the data and to estimate the model. The column R^2 reports the (mean across replications) R^2 from the regression of price on all instruments, treating each product in each market as a separate observation. The meaning of ν is described in the text. These simulations were run on a different computer than the earlier simulations.