Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coe□cients Demand Estimation

Jean-Pierre Dubé
Graduate School of Business
University of Chicago

Jeremy T. Fox
Department of Economics
University of Chicago and
NBER

Che-Lin Su
Graduate School of Business
University of Chicago

October 2008

Abstract

The widely-used estimator of Berry, Levinsohn and Pakes (1995) produces consistent instrumental variables estimates of consumer preferences from a discrete-choice demand model with random coe□cients, market-level demand shocks and potentially endogenous regressors (prices). The nested fixed-point algorithm typically used for estimation is computationally intensive, largely because a system of market share equations must be repeatedly numerically inverted. We provide numerical theory results that characterize the properties of typical nested fixed-point implementations. We use these results to discuss several problems with typical computational implementations and, in particular, cases which can lead to incorrect parameter estimates. As a solution, we introduce a new computational formulation of the estimator that recasts estimation as a mathematical program with equilibrium constraints (MPEC). In many instances, MPEC is faster than the nested fixed point approach. It also avoids the numerical issues associated with nested inner loops. Several Monte Carlo experiments support our numerical concerns about NFP and the advantages of MPEC. We also discuss estimating static BLP using maximum likelihood instead of GMM. Finally, we show that MPEC is particularly attractive for forward-looking demand models where both Bellman's equation and the market share equations must be repeatedly solved.

We thank John Birge, Lars Hansen, Kyoo il Kim, Kenneth Judd, Sven Ley□er, Aviv Nevo, Jorge Nocedal, Hugo Salgado and Richard Waltz for helpful discussions and comments. Thanks to workshop participants at Chicago, INFORMS, the International Industrial Organization Conference, Northwestern, the Portuguese Competition Commission, Rochester and the Stanford Institute for Theoretical Economics. 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 Chicago GSB. Our email addresses are jdube@chicagogsb.edu, fox@uchicago.edu and Che-Lin.Su@chicagogsb.edu.

1 Introduction

The discrete choice class of demand models has become popular in the demand estimation literature due to the models' ability to accommodate rich substitution patterns between a potentially large array of products. The simulated method of moments estimator developed in 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. BLP consider a random coefficients discrete choice model with market-level demand shocks that correlate with prices. They construct moment conditions with which they can address the price endogeneity using standard instrumental variables methods. The approach has had a large impact: as of October 2008, BLP generated of 1000 citations in Google Scholar and the approach has been used in many important empirical studies. However, the estimator is difficult to program and can take a long time to run on a desktop computer. More importantly, some current implementations of the estimator are sufficiently vulnerable to numerical inaccuracy that they may produce incorrect parameter estimates. We summarize some of these computational problems and propose an alternative procedure that is robust to these sources of numerical inaccuracy.

An important component of BLP's contribution consists of a computationally feasible approach to constructing the moment conditions. As in Berry (1994), the main idea is to invert the non-linear system of market share equations. BLP and Berry suggest nesting this inversion step directly into the parameter search. For complex specifications such as random coefficients, this inversion step may not have an analytic inverse and numerical inversion can be prohibitively slow. BLP propose a contraction-mapping routine to solve this system of equations. This step nests an inner loop contraction mapping into the parameter search. Following the publication of Nevo's (2000b) "A Practitioner's Guide" to implementing BLP, numerous studies have emerged using the BLP approach to estimating discrete choice demand systems with random coefficients.

Our first objective consists of exploring the numerical properties of BLP's contraction mapping approach. The GMM objective function can be called hundreds of times during a numerical optimization over structural parameters; each call to the objective function requires a call to the inner loop. Therefore, it may be tempting to use a less stringent stopping criterion for the inner loop in order to speed up estimation. We show theoretically that any numerical error in the contraction mapping is magnified when considering the numerical error to the overall GMM objective function. Running the inner contraction mapping using a loose stopping criteria propagates numerical error into the GMM objective function, which can cause a smooth optimization routine to stop early and produce parameter estimates that are not a true local minimum. Also, numerical error may prevent the optimization routine from being able to diagnose convergence. The main concern is that researchers may

try to increase the speed of the inner loop by using a looser convergence tolerance. This may lead, unfortunately, to incorrect parameter estimates.

Our second objective consists of proposing a new computational method for implementing the BLP estimator that eliminates the inner loop entirely and, thus, eliminates the potential for numerical inaccuracy discussed above. Following Su and Judd (2007), we recast the BLP problem as a Mathematical Program with Equilibrium Constraints (MPEC). The MPEC method minimizes the GMM objective function subject to a system of nonlinear constraints requiring that the predicted shares from the model equal the observed shares in the data. The minimization of an objective function subject to

not nest a contraction mapping. One concern with MPEC may be the large number of parameters in the optimization problem. We increase the number of markets and show that the comparison of the performance of MPEC and NFP does not change as the number of parameters in the optimization problem increases.

computations than Gowrisankaran and Rysman (2007). The more complicated the model of consumer demand, the greater the advantage of MPEC over traditional inner loop approaches.

Another stream of literature, concerned by the statistical e□ciency of GMM estimators, has explored likelihood-based approaches that use additional structure on the joint-distribution of demand and supply (Villas-Boas and Winer 1999; Villas-Boas and Zhao 2005). Jiang et al (2008) propose an alternative Bayesian approach using Markov Chain Monte Carlo methods. In general, likelihood-based approaches still require the numerical inversion of the system of market shares,[1] subjecting them to this

likelihood estimation, where the need to compute the Jacobian makes MPEC especially useful. Second, we discuss the burgeoning literature on dynamic consumer demand.

2 The Demand Model

In this section, we present the standard random coe□cients discrete choice demand model. In most empirical applications, the researcher has access to market shares for each of the available products, but does not have consumer-level information.[2] The usual modeling solution is to build a system of market shares that is consistent with an underlying population of consumers independently making discrete choices among the various products. The population is in most instances assumed to consist of a continuum of consumers with known mass.

Formally, each market $t = 1;:::;T$ has 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 $(x_{j;t};\ _{j;t};p_{j;t})$: The vector $x_{j;t}$ consists of K product attributes. The scalar $_{j;t}$ is a vertical characteristic that is observed by the consumers and firms, but is unobserved by the researcher. $_{j;t}$ 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 $_t = (\ _{1;t};:::;\ _{J;t})^{0}$. Finally, we denote the price of product j by $p_{j;t}$.

Consumer i in market t

$$s_j(x_t, p_t, \xi_t; \theta) = \int \frac{\exp\left(\delta_{j,t} + x_{j,t}^{(2)} \Sigma x_{j,t}^{(2)} + \pi p_{j,t} + \xi_{j,t}\right)}{1 + \sum_{k=1}^{J} \exp\left(\delta_{k,t} + x_{k,t}^{(2)} \Sigma x_{k,t}^{(2)} + \pi p_{k,t} + \xi_{k,t}\right)} dF(\nu; \theta). \quad (2)$$

This is the random coefficient logit model.

In BLP, the goal is to estimate the parameters θ characterizing the distribution of consumer random coefficients, $F(\nu; \theta)$. McFadden and Train (2000) prove that a flexible choice of the family $F(\nu; \theta)$ (combined with a polynomial in $x_{j,t}$ and $p_{j,t}$) allows the random coefficient logit model to approximate arbitrarily any vector of choice probabilities (market shares) originating from a random utility model with an observable linear index (meaning no $\xi_{j,t}$ term). Bajari, Fox, Kim and Ryan (2008) prove the nonparametric identification (no finite-dimensional parameter θ) of $F(\nu)$ in the random coefficient logit model without aggregate demand shocks, using data on market shares and product characteristics. Berry and Haile (2008) prove the nonparametric identification of the entire BLP demand model, including allowing for aggregate shocks. Fox and Gandhi (2008) have an alternative identification proof for heterogeneity that can be adapted for market level demand shocks in the same way as Berry and Haile. However, in most applications, more structure is imposed on the family of distributions characterizing $F(\nu; \theta)$ through the choice of the family $F(\nu; \theta)$, with each family member indexed by the estimable finite vector of parameters θ. For example, BLP assume that $F(\nu; \theta)$ 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.

Typically, the integrals in (2) are evaluated by Monte Carlo simulation. The idea is to generate ns draws from $f(\nu; \theta)$

shares across markets, as the model does not give full support to the data. In the next section, we discuss estimation challenges that arise when $\xi_{j;t}$ is included in the model.

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

until the successive iterates $\exp(\delta_t^{h+1})$ and $\exp(\delta_t^{h})$ are sufficiently close.[4] Formally, we choose a small number, for example 10^{-8} or 10^{-10}, for ε_{in} as the inner loop tolerance level and require $\exp(\delta_t^{h+1})$ and $\exp(\delta_t^{h})$ to satisfy the stopping rule

$$\left| \exp(\delta_t^{h}) - \exp(\delta_t^{h+1}) \right| < \varepsilon_{in} \tag{6}$$

for the iteration $h+1$ where we terminate the contracting mapping (5).[5] Let $\psi_t(\cdot\,; \varepsilon_{in})$ denote the first $\exp(\delta_t^{h+1})$ such that the stopping rule (6) is satisfied. The researcher then uses $\psi_t(\cdot\,; \varepsilon_{in})$ to approximate $\psi_t(\cdot)$:

Researchers often find it tempting to loosen the inner loop tolerance if the NFP contraction mapping is slow. Below, we derive formally the theoretical rate of convergence of the inner loop call to the contraction mapping in terms of the economic parameters of the BLP demand model. Numerical theory proves that the convergence of a contraction mapping is linear at best. Linearly convergent algorithms are typically considered to be slow compared to alternative methods, such as Newton's

bound for the norm of the error is multiplied by a factor equal to L. A proof of this theorem can be found in many textbooks, such as Dahlquist and Björck (2008). The following theorem shows how a Lipschitz constant for a mapping $T(x)$ can be expressed in terms of $r_T(x)$, the Jacobian of T. We then use the Lipschitz constant result to assess an upper bound for the performance of the BLP NFP estimator.

Theorem 2. *Let the function $T(\,): R^n \to R^n$ be differentiable in a convex set $D \subseteq R^n$. Then $L = \max_{\in D} \|r_T(\,)\|$ is a Lipschitz constant for T.*

The contraction mapping in the BLP estimator is

$$T(\,) = \quad + \log S \quad \log s(\,;\,):$$

We define a Lipschitz constant for the BLP contraction mapping T given structural parameters as

$$L(\,) = \max_{\in D} \|r_T(\,)\| = \max_{\in D} \|I \quad r(\log s_j(x_t; p_t;\ _t;\ ))\|;$$

where

$$\frac{\partial \log s_j(x_t; p_t;\ _t;\ )}{\partial\ _{lt}} = \begin{cases} \sum_{r=1}^{ns} \left(\frac{\exp(\ _{0;r} + x^0_{j;t}\ _{x;r} \quad p_{t;r}\ _{p_{j;t}} + \ _{j;t})}{1 + \sum_{k=1}^{J} \exp(\ _{0;r} + x^0_{k;t}\ _{x;r} \quad p_{t;r}\ _{p_{k;t}} + \ _{k;t})} \right) \left(\frac{\exp(\ _{0;r} + x^0_{j;t}\ _{x;r} \quad p_{t;r}\ _{p_{j;t}} + \ _{j;t})}{1 + \sum_{k=1}^{J} \exp(\ _{0;r} + x^0_{k;t}\ _{x;r} \quad p_{t;r}\ _{p_{k;t}} + \ _{k;t})} \right) ; & \text{if } j = l \\ \sum_{r=1}^{ns} \left(\frac{\exp(\ _{0;r} + x^0_{j;t}\ _{x;r} \quad p_{t;r}\ _{p_{j;t}} + \ _{j;t})}{1 + \sum_{k=1}^{J} \exp(\ _{0;r} + x^0_{k;t}\ _{x;r} \quad p_{t;r}\ _{p_{k;t}} + \ _{k;t})} \right) \left(\frac{\exp(\ _{0;r} + x^0_{l;t}\ _{x;r} \quad p_{t;r}\ _{p_{l;t}} + \ _{l;t})}{1 + \sum_{k=1}^{J} \exp(\ _{0;r} + x^0_{k;t}\ _{x;r} \quad p_{t;r}\ _{p_{k;t}} + \ _{k;t})} \right) ; & \text{if } j \neq l: \end{cases}$$

For a given vector of structural parameters , $L(\,)$ is the Lipschitz constant for the NFP inner loop. It is difficult to get precise intuition for this expression 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 for the demand shocks, , as the jth element along the main diagonal is $\frac{\partial s_{j;t}}{\partial\ _{j;t}} \frac{1}{s_{j;t}}$. These expressions are, in turn, related to the degree of asymmetry in the market shares. In section 7.3 below, we use the Lipschitz constant to distinguish between simulated datasets where we expect the contraction mapping to perform relatively slow or fast.

4.2 Determining the Stopping Criteria for the Outer Loop in NFP

This subsection provides guidance on how to select the outer loop tolerance to ensure the outer loop will converge for a given inner loop tolerance. In particular, we show how numerical error from the

inner loop can propagate into the outer loop. We characterize the corresponding numerical inaccuracy in the criterion function, $Q(\ )$; and its gradient. This analysis then informs the decision of what tolerance to use for the outer-optimization loop to ensure that the optimization routine is able to report convergence. This subsection focuses on ensuring the outer loop can actually converge given the numerical inaccuracy of the inner loop. In a later section, we show how this numerical inaccuracy in $Q(\ )$ and its gradient can generate numerical inaccuracy in the parameter estimates of . In some instances, this inaccuracy could imply that the reported estimates are not a true local minimum of $Q(\ )$.

Recall that the outer loop of the BLP estimator consists of minimizing the GMM objective function (4). The convergence of this outer loop depends on the choice of an outer loop tolerance level, denoted by $_{out}$. In theory, $_{out}$ should be set to a small number, such as 10^{-5} or 10^{-6}. In practice, we have found cases in the BLP literature where 10^{-2} was used, possibly to offset the slow performance or non-convergence of the minimization routine. As we illustrate in our Monte Carlo simulations below,

rate of convergence of a contraction mapping. 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. Recall that a smooth optimization routine convergences when the gradient of the objective function is close to zero, by some metric. In the next theorem, we analyze the numerical properties of the gradient. The theorem indicates circumstances in which the outer loop might report convergence despite a numerically inaccurate inner loop.[6] We also show that the choice of the outer-loop tolerance, $_{out}$, should depend on the inner-loop tolerance $_{in}$ and the Lipschitz constant L. This is important because the outer loop tolerance determines the number of significant digits for the solution. Using a tight outer loop tolerance also helps eliminate spurious local minima.

Theorem 4. *Let $L(\ )$ be the Lipschitz constant of the inner-loop contraction mapping for a given and let $_{in}$ be the inner-loop tolerance. Let $\hat{}$ = arg max $\{Q(\ (\ ;\ _{in}))\}$: In order for the outer-loop GMM minimization to converge, the outer-loop tolerance $_{out}$ should be chosen to satisfy $_{out} = O\left(\frac{L(\hat{})}{1\ L(\hat{})}\ _{in}\right)$; assuming $r^2Q(\ )_{\ =(\hat{};0)}$ $\hat{}$ for in a neighborhood of $\hat{}$ is bounded.*

The function $\frac{L(\hat{})}{1\ L(\hat{})}$ is increasing on $[0;1]$, the set of valid Lipschitz constants for a contraction mapping. Therefore, if $_{in}$ is large (the inner loop is loose), then $_{out}$ must also be large (the outer loop must be loose) for the optimization routine to converge. If the inner loop is slow because L is close to 1, then for a fixed $_{in}$, $_{out}$ should be even larger to ensure convergence. The proof is in the appendix.

An immediate consequence of these results is that the researcher may be tempted to select tolerances based on the convergence of the algorithms, rather than the precision of the estimates themselves. In situations where the inner-loop is slow, a researcher may loosen the inner loop tolerance, $_{in}$, to speed convergence of the contraction-mapping. By Theorem 4, the resulting imprecision in the gradient could prevent the optimization routine from detecting a (possibly incorrect) local minimum and converging. In turn, the researcher may be tempted to loosen the outer loop tolerance to ensure convergence of the minimization routine. Besides concerns about imprecision in the estimates, raising $_{out}$ could also generate an estimate that is not in fact a local minimum.

[6] The numerical error in the gradient convergence test may encourage some researchers to use non-smooth optimization methods. Our experiments with MATLAB's version of a genetic algorithm and the simplex method on the BLP NFP problem suggest that both non-smooth optimizers can report convergence to a point that is not a local minimum, even with a tight inner loop tolerance $_{in}$ and tight outer-loop tolerance $_{out}$. We can verify whether a point is a true local minimum by starting a high-quality smooth optimization routine at that point. If it is a local minimum, the smooth routine will immediately report convergence. For these reasons, we focus on smooth optimizers in this paper.

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 $_{in}$, that we have chosen $_{out}$ to ensure that the algorithm is able to report convergence. Let $^{?} = \arg\max \{Q(\ ; 0)\}$ be the maximizer of the finite-sample objective function without numerical error. As economists, now we are interested in the errors in the final estimates, $\hat{}$, from using a loose inner loop.

Theorem 5. *Assume that* $r \ Q(\ )_{= (\hat{};0)}$ *is bounded and that*

$$O \quad \hat{}; _{in} \quad \hat{};0 \quad ^{2} \quad Q \quad \hat{}; _{in} \quad Q(\ (\ ; 0)) + r \ Q(\ )_{= (\hat{};0)} \quad \hat{}; _{in} \quad \hat{};0 \quad :$$

The di erence between the finite-sample maximizers with and without numerical error satisfies

$$O \quad \hat{} \quad ^{2} \quad Q \quad \hat{}; _{in} \quad Q(\ (\ ; 0)) + O \quad L(\hat{})$$

4.4 Large Sample Bias from the Inner-Loop Numerical Error

The previous section focused only on numerical errors for a finite data set. We now use statistical theory to examine the large-sample properties of the BLP estimator using the NFP algorithm. Before, $^{?}$ was the true minimizer of the finite-sample GMM objective function without any inner-loop numerical errors. Now instead consider ^{0}, the true parameters in the data generating process. Even a researcher with a perfect computer program will not be able to recover ^{0} because of statistical sampling error. Here we explore how numerical errors in the inner loop affect the consistency of the BLP estimator.

Recall that $\hat{}$ corresponds to the minimizer of $Q\quad\hat{};\ _{\text{in}}\ ;$ the biased GMM objective function with the inner-loop tolerance $_{\text{in}}$. Let $Q(\ (\ ;0)) = E[Q(\ (\ ;0))]$ be the probability limit of $Q(\ (\ ;0))$, as either $T\ !\ 1$ or $J\ !\ 1$, as in Berry, Linton and Pakes (2004). Let be the minimizer of $Q(\ (\ ;\ _{\text{in}}))$, the population objective function with the inner-loop tolerance $_{\text{in}} > 0$. Clearly, $^{0} = \arg\min Q(\ (\ ;0))$ if the BLP model is identified.

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), will converge to ^{0} if $Q(\ (\ ;0))$ converges to $Q(\ (\ ;0))$ uniformly, which is usually 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,

$$\hat{}\quad{}^{0}\quad\hat{}\quad+\quad{}^{0} = O\left(\sqrt{\frac{L(\hat{})}{1\quad L(\hat{})}}\ _{\text{in}}\right) + O\left(1=\sqrt{T}\right);\tag{7}$$

where $\hat{}\quad = O\left(\sqrt{\frac{L(\hat{})}{1\quad L(\hat{})}}\ _{\text{in}}\right)$ because we showed $O\left(\hat{}\quad^{2}\right)\quad O\left(\frac{L(\hat{})}{1\quad L(\hat{})}\ _{\text{in}}\right)$ in the previous subsection. These results suggest that the asymptotic bias due to 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

numerical derivatives. The gradient is approximated by

$$r_d Q(\ (\ ;\ _{\text{in}})) = \frac{Q(\ (\ + de_k;\ _{\text{in}}))\quad Q(\ (\quad de_k;\ _{\text{in}}))}{}$$

5.1 NFP Algorithm Implementations

For all NFP implementations, we examine the one-step GMM estimator with Nevo's (2000) suggestion of using the weighting matrix $W = (Z'Z)^{-1}$, where Z is the $TJ \times D$ matrix of instruments $z_{j;t;k}$.[8] We use one fake data set and one real data set to show that NFP with loose inner loop tolerances can lead to incorrect parameter estimates.

We use three implementations of NFP for our real data and fake data tests. We use the same data and set of starting values for all three implementations. We use our numerical theory results from section 4 to guide us in the selection of inner and outer loop tolerances for the NFP algorithm. To assess the importance of those findings, we construct three scenarios which we examine for each Monte Carlo experiment. In the first scenario, we explore the implications of a tight outer loop tolerance, set at $_{out} = 10^{-6}$; and a loose inner loop tolerance, set at $_{in} = 10^{-4}$. The former outer loop tolerance is the default setting for most state-of-the-art optimization algorithms. However, from our numerical theory results, we know the latter inner loop tolerance is too large. One could think of this scenario as representing the "frustrated researcher" who loosens the inner loop to speed the apparent rate of convergence. In the second scenario, we explore the results from Theorem 4, whereby the loose inner loop tolerance could, in turn, prevent the outer loop from converging. Specifically, we keep $_{in} = 10^{-4}$ and set $_{out} = 10^{-2}$. One can think of this scenario as representing the attempt of the researcher to loosen the outer loop to force it to converge, even though in practice the converged point may not actually satisfy the first-order conditions. In our third scenario, we implement the "best practice" settings for the NFP algorithm with $_{in} = 10^{-14}$ and $_{out} = 10^{-6}$.

For all implementations of NFP, we use the same programming environment (MATLAB) and the same optimization package (KNITRO using the TOMLAB interface). We selected MATLAB because this is a commonly-used software package among practitioners. We also selected the KNITRO optimization package instead of MATLAB's built-in optimization routines as the former is a highly-respected, state-of-the-art solver in the optimization community (Byrd, Nocedal and Waltz 1999). For our fake data example, we use numerical derivatives. For our real data example, we also supply derivatives for each algorithm because all local optimization methods improve if the user supplies exact derivatives of the objective function.[9]

We also customized several aspects of the NFP algorithm to increase speed. In the case of NFP, the

[8] We choose a simple weighting matrix because our focus is on comparing algorithms, not finding the most statistically efficient estimator.

[9] Another option is to use automatic differentiation software. Automatic differentiation software is automatically used by some languages, such as AMPL, and can be accessed with the TOMLAB interface for MATLAB. Our experience has been that automatic differentiation is very slow for NFP. Also, software packages like AMPL are impractical for NFP algorithms because AMPL is a problem definition language, not a general purpose programming language like MATLAB. Therefore, we use MATLAB for all our empirical analysis. However, in practice, many users may find AMPL more convenient for the MPEC implementation. One warning: the automatic differentiation overhead in AMPL uses lots of computer memory.

most notable speed improvements came from exploiting as much as possible the built-in linear algebra operations ("vectorization") for the inner loop. In addition, we exploited the normality assumption for $F(\ ;\ )$ to concentrate the means out of the parameter search under the NFP algorithm, as suggested in Nevo (2000b). Therefore, the NFP algorithm can be recast to search only over the standard deviation of the random coefficients, rather than both the means and standard deviations. Relaxing the normality assumption would prevent the use of this simplification (except perhaps in other location and scale families), which could improve the relative speed performance of MPEC over NFP even further.The Fake Data Generating Process

We use the demand model from section 2. In this section we describe a data generating process for a base case. The individual experiments perturb aspects of the data generating process from this base case. We allow for $K = 3$ observed characteristics, in addition to prices. We also estimate a random coefficient on the intercept, β_i^0, which models the relative attractiveness of purchasing any of the products instead of the outside good. β_i^p, the price coefficient, is also random.

We focus on markets with a fairly large number of products, $J = 75$, to ensure that our results are not due to sampling error. We also consider an intermediate number of statistically independent markets, here $T = 25$. Although not reported, we noticed large biases in the mean and standard deviation of the intercept, β_i^0; as well as functions of the parameters (like price elasticities) when a small number of markets was used. Intuitively, the moments of β_i^0 are identified in part from the share of the outside good, and more markets are needed to observe more variation in the outside good's shares.

For product j in market t, let

$$\begin{bmatrix} x_{1;j;t} \\ x_{2;j;t} \\ x_{3;j;t} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \begin{bmatrix} 1 & 0{:}8 & 0{:}3 \\ 0{:}8 & 1 & 0{:}3 \\ 0{:}3 & 0{:}3 & 1 \end{bmatrix} \right);$$

Likewise, $\xi_{j;t} \sim N(0; \sigma_\xi^2)$, with the default $\sigma_\xi^2 = 1$. Price is

$$p_{j;t} = |0{:}5\ \xi_{j;t} + e_{j;t}| + 1{:}1 \sum_{k=1}^{3} x_{k;j;t};$$

where $e_{j;t} \sim N(0; 1)$ is an innovation that enters only price. Prices are always positive. Prices are endogenous as $\xi_{j;t}$ enters price. For each product j in market t, there is a separate vector $z_{j;t}$ of $D = 6$

cients. To maintain consistency with the application in BLP (1995) and the related empirical literature, we assume independent normal random coe□cients on each product characteristic and the intercept. Thus, $F(\ ;\ )$ is the product of five independent normal distributions ($K = 3$ attributes, price and the intercept) characterized by means and standard deviations contained in . The true values of the moments of the random coe□cients $_i = \ _i^0;\ _i^1;\ _i^2;\ _i^3;\ _i^p$ are $E[\ _i] = f\ 1;\ 1{:}5;\ 1{:}5;\ 0{:}5;$

interpretation of their findings. With multiple starting values, careful implementation of the numerical procedures, and state-of-the-art optimization solvers, the BLP GMM estimator appears to produce reliable estimates. [13]

5.3 Fake Data, Numerical Derivatives and False Parameter Estimates

For NFP, the numerical theory in section 4 raises several concerns about the common practice of setting the tolerance, $_{in}$; too high (too loose). Section 4.5 shows that a combination of a loose inner loop, numerical derivatives and a smooth optimization routine can produce incorrect parameter estimates. Also recall that Theorem 4 shows that if $_{in}$ is too loose, $_{out}$ must be set to be too loose in order for the routine to be able to report convergence.

In this subsection, we explore empirically the problems with loose inner loop tolerances and numerical derivatives. We create one simulated fake dataset, using the data generating process from section 5.1. Holding the simulated data fixed, we first compare the estimates produced from 100 randomly-chosen starting values for the own-price demand elasticities. We run each of the three NFP implementations described in section 5.1 for each of the 100 vectors of starting values.Table 1 reports the results for the 100 different starting values. The first row reports the fraction of runs for which the routine reports convergence. As Theorem 4 shows, if the inner loop tolerance is a loose $_{in} = 10^{-4}$ and the outer loop tolerance a standard value of $_{out} = 10^{-6}$, the routine will never report convergence. Column one confirms this finding as only 2% of the runs with a loose inner loop and tight outer loop converge. In contrast, column two indicates that the algorithm is more likely to converge (30% of the runs) when we also loosen the tolerance on the outer loop. As we will show below, this semblance of convergence is merely an artifact of numerical imprecision that leads to misleading estimates. 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 numerically found across all three NFP implementations and all 100 starting values (300 cases).

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

	NFP Loose Inner	NFP Loose Both	NFP Tight	Truth
Fraction Reported Convergence	0.02	0.30	0.95	
Frac. Obj. Fun. < 1% Greater than "Global" Min.	0.0	0.0	0.25	
Mean Own Price Elasticity Across All Runs	-12.28	-12.30	-5.77	-5.68
Std. Dev. Own Price Elasticity Across All Runs	19.44	19.43	0.0441	
Lowest Objective Function Value	0.0217	0.0327	0.0169	
Elasticity for Run with Lowest Obj. Value	-5.89	-5.63	-5.77	-5.68

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

NFP tight should not find the global minimum every time, because a gradient-based optimization routine may indeed converge to a local minimum.

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 100 starting values, all $J = 25$ products and all $T = 75$ markets:

$$\frac{1}{100}\sum_{h=1}^{100}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{j=1}^{J} \hat{p}_{j;t}(\hat{\theta}^h);$$

where $\hat{\theta}^h$ is the vector of parameter estimates for the hth starting value and $\hat{p}_{j;t}(\hat{\theta}^h)$ 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 elasticity across all 100 runs: $\frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{j=1}^{J}\hat{p}_{j;t}(\hat{\theta}^h)$.

Beginning with the third row, first note that in the final column we report the own-price demand elasticity[6]. We use evaluated at the true parameter values: -5.68.

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
Fraction Reported Convergence	0.0	0.81	1.00
Frac. Obj. Fun. < 1% Greater than "Global" Min.	0.0	0.0	1.00
Mean Own Price Elasticity Across All Runs	-3.75	-3.69	-7.43
Std. Dev. Own Price Elasticity Across All Runs	0.03	0.08	~0
Lowest Objective Function Value	15.3816	15.4107	4.5615

The results in Table 2 are of the same format as Table 1. 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 81% 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 global minimum, 4.5615, 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 -3.75 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 0.03 and 0.08 for the loose inner loop methods, the answers are consistently wrong across runs. The fifth row shows the smallest objective function values found by the loose inner loop and loose both routines are 15.38 and 15.41, respectively. These are far from the true "global" minimum of 4.56.

These results show that a naive but otherwise careful researcher might feel that his or her estimates were correct because even trying 25 different starting values always produce around the same estimates. Even if the researcher correct coded the derivatives in closed form and used a high-quality, professional optimizer like KNITRO, the NFP loose inner and loose both implementations can consistently converge to the wrong elasticity, and the elasticity can be half of the true value. Thus, there is no diagnostic that a researcher can do that will detect all types of numerical error. With Nevo's cereal dataset, an inner loop tolerance that is too loose will lead to consistent but consistently wrong own-price elasticity estimates. Only using an a priori theoretically correct setting, like a tight inner loop tolerance, will avoid these errors.

6 A Constrained Optimization Approach to Improve Speed

We have established that only NFP with a tight inner loop tolerance can produce reliable parameter estimates. According to Theorem 5, if we wish to achieve the default numerical precision in the outer loop of 10^{-6}, we need to set the NFP inner loop tolerance to 10^{-12} or tighter, for reliable parameter estimates. Using a tight inner loop means NFP may be slow. Further, in the previous section, we established that the NFP method's inner loop converges linearly and can be slow when the Lipschitz constant is close to 1. A slow inner loop might cause researchers to choose loose tolerances for the inner loop, which might lead to problems in establishing the convergence of the outer loop as well as errors in the reported parameter estimates.[15]

[15] 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 line-search or trust-region procedure subject to some technical assumptions. In general, most practitioners would be daunted by the task of nesting a hybrid Newton

In this section, we propose an alternative algorithm based on Su and Judd's (2007) constrained optimization approach for estimating structural models. Below we show that the MPEC approach generates the same solution as NFP. MPEC can save computation time while completely avoiding issues of numerical precision by eliminating the inner loop of the NFP algorithm. In their original paper, Su and Judd focus more on solving for the unknown variables in economic models, such as value functions in single-agent dynamic programming problems and the entry probabilities of rival firms in static Bertrand entry games with multiple equilibria. We apply this insight to the recovery

The constrained optimization defined by (9) can be solved using modern nonlinear optimization solvers developed by researchers in numerical optimization. Unlike the NFP algorithm, where users need to exercise caution in the choice of tolerance levels for both inner and outer loops, the defaults on feasibility and optimality tolerances in nonlinear optimization solvers for constrained optimization are usually su□cient. These default tolerances have been established to work well in hundreds or thousands of papers in the numerical analysis literature. The default tolerances are usually su□cient because the market share equations and GMM objective function (without an inner loop) are exposed to the optimization routine. In short, MPEC lets a state-of-the-art optimization algorithm handle all of the computational aspects of the problem. In contrast, with NFP, the researcher needs to customize a nested-fixed-point calculation, which could result in naive errors.

In addition to simplifying implementation, bypassing the inner loop reduces several sources of numerical error that could, possibly, lead to non-convergence. We have detected some common practices with the coding of the inner loop that could naively lead to numerical error. These include loose choices of thepn78nerpn78npn78nIacould opti-317(s(pn78n)o7some)(sol8)28(e78nvi((whmizationdur-27(de78n)-1(er)-3380(-27(de

routine is exposed to the constraints, the derivatives of the constraints and of the objective function, and the sparsity pattern of the constraints. On sparsity, recall that demand shocks for market t do not enter the constraints for market $t + 1$. Therefore, this constrained optimization problem is highly sparse.

Most constrained optimization solvers are based on sequential quadratic programming or interior point methods. As stated earlier, these solvers use Newton-based methods. Economists are often skeptical about Newton's method because it might not converge if the starting point is far away from the solution. While this perception is true for the purest textbook version of Newton's method, modern Newton-like methods incorporate a line-search or a trust-region strategy to give more robustness to the choice of starting values. We refer readers to Nocedal and Wright (2006) and Kelley (1995, 1999, 2003) for further details on modern optimization methods for smooth objectives and constraints.

Finally, our implementation of MPEC for the BLP model is slightly more sophisticated than the simple explanation in (9). We actually treat the moments as separate parameters, so that the problem being solved is

$$\begin{aligned} \min_{;\ ;} \quad & {}^{0}W \\ \text{subject to} \quad & g(\ ) = \quad : \\ & s(\ ;\ ) = S \end{aligned} \tag{10}$$

The solution to this new problem is the same as (9). The objective function is now a simple quadratic, ${}^{0}W$, rather than a more complex, direct function of ; the additional constraint $g(\ ) \quad = 0$ is linear in both and and, hence, does not add additional di□culties to the original problem. Computationally, the advantage with this equivalent formation is that we increase the sparsity of the constraint Jacobian and the Hessian of the Lagrangian function by adding the additional variables and constraints. In numerical optimization, it is often easier to solve a large but sparse problem than a small but dense problem. Another advantage of MPEC over NFP is that the objective function and constraints in MPEC are likely more "smooth" or less "nonlinear" in the unknowns than the NFP objective function is in . In NFP, the mapping from to the objective function value uses the very

7 Speed Comparisons of MPEC and NFP

NFP with a tight inner loop will produce correct parameter estimates if many starting values are used. However, NFP can be slow on some datasets. This section uses fake data and the Nevo cereal to compare the speed of MPEC and NFP. We present examples where MPEC performs better than NFP. This is not meant to be a theorem: there could be cases where NFP is faster than MPEC. We now show that, in many situations, NFP may be computationally impractical in terms of execution time. In contrast, we will show that MPEC's execution time appears to be relatively invariant across these situations. Our approach exploits the Lipschitz constant for the BLP contraction mapping derived in section 4.1. We conjecture that data with a higher Lipschitz constant, and hence a higher upper bound on the rate of convergence of the inner loop, may slow NFP estimation. The idea will be to manipulate various components of the data-generating process in order to measure their respective impact on the Lipschitz constant. We have no reason to believe cases exist where MPEC grows really slow with some equivalent of a Lipschitz constant. Therefore, we suspect that MPEC will be more robust against extremely slow performance. Keep in mind that is these slow-performing cases where a researcher will be tempted to loosen the inner loop tolerance, leading to the problem of incorrect parameter estimates that we earlier highlighted.

7.1 NFP and MPEC Implementations

We code NFP and MPEC using closed-form derivatives. As the proof of Theorem (6) shows, the components of these derivatives are the same for both methods. We the quadratic form of MPEC in (10). We give the sparsity pattern of the constraints tomore

the two algorithms are initialized to have the same objective function value.[18] 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. This is our attempt to equalize the starting values across NFP and MPEC.[19]

Table 3: Lipschitz Constants for the NFP Algorithm

Parameter Scale		Std. Dev. of Shocks		# of Markets T		Mean of Intercept $E\left[\begin{smallmatrix}0\\i\end{smallmatrix}\right]$	
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						

products within each market and then across markets. For each algorithm, we report the total CPU time required for all 10 runs. 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 converge in all scenarios. We also find that MPEC and NFP generate identical point estimates, as one would expect since they are statistically the same estimator (Theorem 6). We compute the root mean-squared error (RMSE) and the bias of the own-price elasticities. For a parameter

Table 4: Monte Carlo Results Varying the Lipschitz Constant

Intercept	Lipschitz	Implementation	Runs Converged	CPU Time (s)	Elasticities	
$E\left[\begin{smallmatrix}0\\j\end{smallmatrix}\right]$	Constant		(fraction)		Bias	RMSE
-2	0.806	NFP tight	1	1184.1	0.026	0.254
		MPEC	1	1455.1	0.026	0.254
-1	0.895	NFP tight	1	1252.8	0.029	0.258
		MPEC	1	1528.4	0.029	0.258
0	0.950	NFP tight	1	1352.5	0.029	0.265
		MPEC	1	1564.1	0.029	0.265
1	0.978	NFP tight	1	1641.1	0.029	0.270
		MPEC	1	1562.5	0.029	0.270
2	0.991	NFP tight	1	2498.1	0.030	0.273
		MPEC	1	1480.7	0.030	0.273
3	0.997	NFP tight	1	5128.1	0.031	0.276
		MPEC	1	1653.9	0.030	0.278
4	0.999	NFP tight	1	9248.5	0.032	0.279
		MPEC	1	1881.8	0.031	0.279

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

the inner loop, it avoids all the potential risks of naive implementations with loose tolerances. We therefore recommend MPEC as a safer and more reliable algorithm for the estimation of the BLP GMM estimator.

7.5 Varying the Number of Markets

In the previous section, we demonstrated that MPEC has a speed advantage over NFP when the Lipschitz constant is high. However, some readers may be concerned that MPEC may not be practical as one increases the number of products or the number of markets. The reason is that there is one nuisance optimization parameter, $_{j;t}$, for each product j and market t combination. As the number of markets T (or the number of products J) increases, there will be more $_{j;t}$s over which to optimize and, correspondingly, more constraints. The next set of Monte Carlo experiments compare estimation with di□ering numbers of markets to see whether MPEC's speed advantage is related to having a small number of demand shocks.

Table 5 returns to the base specification, and varies only the number of markets, T. As the number of markets increases, not surprisingly both methods take longer. MPEC and NFP with tight

Table 5: Monte Carlo Results Varying the Number of Markets

# Markets	Lipschitz	Implementation	Runs Converged	CPU Time (s)
T	Constant		(fraction)	
25	0.937	NFP Tight	1	258.5
		MPEC	1	226.8
50	0.944	NFP Tight	1	780.0

8 Other Computational Issues with BLP

8.1 Simulating Market Shares

The times for all methods reported in Table 4, the Monte Carlo results, are lower bounds on the actual speeds of these methods in applications. By shutting down simulation error, we were able to get by with *ns* = 100 simulation draws in the market share equations, (3). Our experiments with data generated using many more draws suggests that perhaps 10,000 draws might be appropriate to eliminate most simulation error, for models with five independent normal random coe□cients. Using

can easily adapt the estimator to accommodate a structural (full-information) approach that models

maximize the following log-likelihood function

$$l(\quad) = \sum_{t=1}^{\bar{X}} \log\left(f_{s;p}(s_t; p_t; \quad)\right):$$

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

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

$$\begin{aligned} l^{\text{MPEC}}(\ ;\ ) \quad &= \quad \sum_{t=1}^{T} \log \quad f_j \ (s_t \, j \, x_t; p_t; \ ;\ ) \, jJ \ _{!\,s} j \, f \ (p_t \, j \, z_t; \ ;\ ) \\ \text{subject to} \qquad & s(\ ;\ ) = S \end{aligned} \qquad : \tag{13}$$

10 Extension: Dynamic Demand Models

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, : The second inner loop is the iteration of the Bellman equation to obtain the consumer's value function

consumer r's expected value of waiting is

$$v_0^r(p_t;\ ^r) = \int_{\ } \max\left\{ v_0^r(p_t^0{}_{\ j} + \ ;\ ^r) + {}_0 \ ;\ \max_j \ {}_j^r \ {}^r(p_t^0{}_{\ j} + \ + \ ) + {}_j + {}_j \right\} dF_{\ }(\ )dF_{\ ;\ }(\ ;\ )$$
$$= \log \exp(v_0^r(P_j(p_t;\ _p) + \ ;\ ^r)) + \sum_j \exp \ {}_j^r \ {}^r(P_j(p_t;\ _p) + \ ) + {}_j \ dF_{\ ;\ }(\ ;\ ): \tag{15}$$

To simplify the calculation of the expected value of waiting, we approximate it with Chebyshev polynomials (Judd 1998).[22] We outline the Chebyshev approximation in Appendix C.

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

$$\begin{bmatrix} {}_h \\ {}_h \end{bmatrix} = \begin{cases} {}^1; & \Pr(1) = {}_1 \\ \vdots & \vdots \\ {}^R; & \Pr(R) = 1 \ {}_{r=1}^{R\ 1} \ {}_r \end{cases}:$$

This heterogeneity implies that certain types of consumers will systematically purchase earlier than others. Thus, the mass of remaining consumers of a given type r, M_t^r; evolves over time as follows:

$$M_t^r = \begin{cases} M_{\ r} & ;\ t = 0 \\ M_t^r{}_{\ 1} S_0^r(X_{t\ 1};\ ^r) & ;\ t > 0 \end{cases}:$$

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

$$s_j(p_t;\ ) = \sum_{r=1}^{R} {}_{t;r} \frac{\exp(\ {}_j^r \ {}^r p_{j;t} + \ _{j;t})}{\exp\left(v_0^r(p_t;\ ^r)\right) + \sum_{k=1}^{J} \exp\left(\ {}_k^r \ {}^r p_{k;t} + \ _{k;t}\right)}; \tag{16}$$

where

$${}_{t;r} = \begin{cases} {}_r & t = 0 \\ \dfrac{M_t^r}{\sum_r M_t^r} & ;t > 0 \end{cases}$$

is the proportion of type r consumers still in the market at date t:

The empirical model consists of the system (14) and (16), which we write more compactly as

$$\begin{bmatrix} u_t \\ {}_t \end{bmatrix} \begin{bmatrix} \log(p_t) \\ p_t^0{}_{\ 1} \end{bmatrix} k$$

parameters. The multivariate normal distribution of ($_{j;t}$; $_{j;t}$) induces the density on the observable outcomes, ($p; S_t$),

$$f_{p;S}(p_t; S_t; \ ; \ ; \ ) = \frac{1}{(2\ )^{\frac{3J}{2}} j\ \ j^{\frac{1}{2}}} \exp \quad \frac{1}{2} u_t^{\ 0}\ \ _u{}^{1} u_t \quad j J_{t;u\ !\ Y} j$$

where $J_{t;u\ !}$

assume there is only a single consumer type, $R = 1$. It is easy to show that in this case, $_t$ can be computed analytically by log-linearizing the market shares, (16).[23] We begin with this case because it only involves a nested call to the calculation of the expected value of waiting. Below we will allow for more consumer types to see what happens when we also require a nested call to the numerical inversion of the shares. We assume that the consumers' preferences are: $(_1; _2;) = (4; 1; :15)$ and the discount factor is $= 0:99$.[24] We assume that the density of prices has the transition rules

$$\begin{bmatrix} p_{1;t} \\ p_{2;t} \end{bmatrix} = \begin{bmatrix} 5 + :8p_{1;t\ 1} + :2p_{2;t\ 1} + _{1;t} \\ 5 + :1p_{1;t\ 1} + :55p_{2;t\ 1} + _{2;t} \end{bmatrix};$$

Note how the lagged price of product 2 effects the price of product 1, and vice versa. Finally, we assume the supply and demand shocks satisfy $(_{j;t}; _{j;t}) \quad N\left(0; \begin{bmatrix} 1 & 0:5 \\ 0:5 & 1 \end{bmatrix}\right)$ and are independent across markets and time periods. For our Chebyshev approximation, we use 6 grid points and a 6th order polynomial. For the NFP algorithm, we use an inner loop tolerance of 10^{-14} for the calculation of the expected value of waiting.

Results from 25 replications of this first experiment are reported in Table 6. We report the bias and RMSE associated with each of the structural parameters, for MPEC and NFP respectively. Interestingly, MPEC seems to produce estimates that, on average, have lower bias while NFP seems to produce lower RMSE. This may be a consequence of using only one starting value per replication. More importantly, the average CPU time for MPEC is just over 25% of the CPU time for NFP.

Now we run a second Monte Carlo experiment where we allow for two types of consumers.

11 Conclusions

In this paper, we analyzed the numerical properties of the NFP approach proposed by BLP to estimate the random coefficients logit demand model. Theoretically, the NFP approach may be slow, as NFP's inner loop is only linearly convergent and and NFP is more vulnerable to error due to the inner loop. We showed the Lipschitz constant is a measure of an upper bound to the convergence rate of NFP's inner loop's contraction mapping. We numerically evaluated the Lipschitz constant for particular

	Bias		RMSE	
: taste parameters	MPEC	NFP	MPEC	NFP
$_1$: 4	7.5E-03	4.6E-02	1.7E-01	1.5E-01
$_2$: -1	6.2E-03	3.7E-02	1.5E-01	1.2E-01
: -0.15	-1.1E-04	-2.9E-04	8.0E-04	5.4E-04
: price transitions				
int_1 : 5	9.4E-03	1.9E-02	4.9E-02	4.6E-02
$_{1;1}$: 0.8	9.5E-05	-2.1E-04	1.2E-03	1.2E-03
$_{1;2}$: 0.2	-1.6E-04	-3.8E-05	1.5E-03	1.7E-03
int_2 : 5	8.9E-03	6.6E-04	5.9E-02	3.2E-02
$_{2;1}$: 0.1	-7.0E-05	1.5E-04	1.1E-03	5.6E-04
$_{2;2}$: 0.55	-6.5E-05	-4.5E-04	1.4E-03	8.8E-04
: variances of shocks				
1	-4.1E-03	-4.5E-03	1.7E-02	1.7E-02
0.866	-1.7E-03	-5.5E-04	1.5E-02	1.4E-02
0.5	-7.9E-04	-2.4E-03	2.0E-02	1.9E-02
Avg CPU time (sec)	4579.3	16,971		

Table 6: Monte Carlo Results for Dynamic BLP with One Consumer Type: NFP versus MPEC

coe□cient logit demand model. MPEC is quicker to compute and avoids numerical errors because it avoids repeatedly inverting the market shares equations numerically. It also allows the researcher to access state-of-the-art constrained optimization solvers.

To assess the practical aspects of MPEC versus NFP, we conducted a number of Monte Carlo

collection of three loops (optimization, market shares, dynamic programming) makes the traditional BLP approach nearly untenable in terms of computational time. 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.

References

[1] Ackerberg, D.. and M.. Rysman (2005), Unobserved Product Di□erentiation in Discrete Choice Models: Estimating Price Elasticities and Welfare E□ects, *RAND Journal of Economics*, 36 (4): 771–788.

[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 and S. P. Ryan (2007), "Linear Regression Estimation of Discrete Choice Models with Nonparametric Distributions of Random Coe□cients." *American Economic Review*, 97(2), 459-463

[4] Bajari, P., J. T. Fox, K.-I. Kim and S. P. Ryan (2008), A Simple Nonparametric Estimator for the Distribution of Random Coe□cients in Discrete Choice Models. Working paper.

[5] Berry, S. (1994), Estimating Discrete-Choice Models of Product Di□erentiation. *RAND Journal of Economics*, 25(2): 242–262.

[6] Berry, S. and P. A. Haile (2008), Nonparametric Identification of Multinomial Choice Models with Heterogeneous Consumers and Endogeneity, working paper.

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

[8] Berry, S., J. Levinsohn, and A. Pakes (2004), "Di□erentiated Products Demand Systems from a Combination of Micro and Macro Data: The New Car Market." *Journal of Political Economy*, 112(1):68-105.

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

[10] Byrd, R. H., J. Nocedal and R. A. Waltz (1999), An Interior Point Method for Large Scale Nonlinear Programming. SIAM J. Optimization, 9, 4, 877-990.

[11] Carranza, Juan Esteban. (2008) Product innovation and adoption in market equilibrium: The case of digital cameras. Working paper.

[12] Dahlquist, Germund and Åke Björck. (2008) *Numerical Methods in Scientific Computing*. SIAM, Philadelphia.

[13] Davis, Peter J. (2006) "The Discrete Choice Analytically Flexible (DCAF) Model of Demand for

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

[27] Knittel, Christopher R. and Konstantinos Metaxoglou (2008), "Estimation of Random Coe□cient Demand Models: Challenges, Diculties and Warnings." Working paper.

[28] Lee, Robin S. "Vertical Integration and Exclusivity in Platform and Two-Sided Markets." Working paper.

[29] Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green. (1995) *Microeconomic Theory.* Oxford University Press.

[30] McFadden, D. and K. Train (2000), Mixed MNL models for discrete response. *Journal of Applied Econometrics*, 15(5): 447–470.

[31] Melnikov, Oleg (2001) Demand for Di□erentiated Durable Products: The Case of the U.S. Computer Printer Market. Working paper.

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

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

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

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

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

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

[38] Petrin, A. and K. Train (2006), Control Function Corrections for Omitted Attributes in Di□erentiated Products Markets. Working paper.]

[39] Schiraldi, Pasquale (2008). "Automobile Replacement: a Dynamic Structural Approach." Working paper.

[40] Su, C.-L. and K. L. Judd (2007), Constrained Optimization Approaches to Estimation of Structural models. Working paper, CMS-EMS, Kellogg School of Management.

[41] van der Vaart, A. W. (2000). *Asymptotic Statistics*. Cambridge.

[42] Villas-Boas, J. Miguel and Russell S. Winer (1999). "Endogeneity in Brand Choice Models." *Management Science,* 45:1324-1338.

[43] Villas-Boas, J. Miguel and Ying Zhao (2005). "Retailer, Manufacturers, and Individual Consumers: Modeling the Supply Side in the Ketchup Marketplace." *Journal of Marketing Research,* 42, 83-95.

A Proofs

A.1 Proof of Theorem 3

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

$$= \frac{@Q(\ )}{@} \Big|_{=(\ ;0)} \Big[Q(\ (\ ;\ _{in})) \quad Q(\ (\ ;0)) \Big]_0 (\ (\ ;\ _{in}) \quad (\ ;0)) + O\ k\ (\ ;\ _{in}) \quad (\ ;0) k$$

then the regular triangle inequality, and then finally a Taylor series expansion, we have

$$
\begin{aligned}
& \nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} \\
& \nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} \\
& \nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} \\
= \quad & \nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\tilde{\theta};0)} + \nabla Q(\cdot)_{(\tilde{\theta};0)} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} \\
& \nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\tilde{\theta};0)} \\
& + \nabla Q(\cdot)_{(\tilde{\theta};0)} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} \\
& O\left(\frac{L(\hat{\theta})}{1 - L(\hat{\theta})}\varepsilon_{in}\right) + \nabla^2 Q(\cdot)_{(\hat{\theta};0)}\left(\tilde{\theta} - \hat{\theta}\right) + O\left(\tilde{\theta} - \hat{\theta}\right)^2 .
\end{aligned}
$$

As we have assumed $\nabla^2 Q(\cdot)_{(\hat{\theta};0)}\left(\tilde{\theta} - \hat{\theta}\right)$ is bounded, we obtain

$$
\begin{aligned}
\nabla Q(\cdot)_{(\tilde{\theta};\,\varepsilon_{in})} \quad \nabla Q(\cdot)_{(\hat{\theta};0)} + O\left(\frac{L(\hat{\theta})}{1 - L(\hat{\theta})}\varepsilon_{in}\right) + O\left(\tilde{\theta} - \hat{\theta}\right)^2 \\
= O\left(\frac{L(\hat{\theta})}{1 - L(\hat{\theta})}\varepsilon_{in}\right) + O\left(\tilde{\theta} - \hat{\theta}\right)^2 .
\end{aligned}
$$

Hence, the norm of the numerically inaccurate gradient, evaluated at an arbitrary point $\tilde{\theta}$, is bounded above by a term on the order of $\frac{L(\hat{\theta})}{1 - L(\hat{\theta})}\varepsilon_{in}$ and a term involving the arbitrary point $\tilde{\theta}$ and the GMM estimator with error, $\hat{\theta}$. The term $O\left(\frac{L(\hat{\theta})}{1 - L(\hat{\theta})}\right)$ indicates that the numerical error in the gradient is linearly increasing in ε_{in} (decreasing ε_{in} decreases the numerical error in the gradient).

A.3 Proof of Theorem 5

Rearranging the equality involving $Q(\hat{\theta}; \varepsilon_{in}) - Q(\theta(\hat{\theta}; 0))$ to focus on the $O(\hat{\theta}^2)$ term, we have

$$\begin{aligned} O(\hat{\theta}^2) = {} & Q(\hat{\theta}; \varepsilon_{in}) - Q(\theta(\hat{\theta}; 0)) - \left[\nabla_{\theta} Q(\theta)|_{\theta=\theta(\hat{\theta};0)}\right]' \left(\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\right) \\ & - O\left(\|\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\|^2\right) \\ = {} & Q(\hat{\theta}; \varepsilon_{in}) - Q(\theta(\hat{\theta}; 0)) + \nabla_{\theta} Q(\theta)|_{\theta=\theta(\hat{\theta};0)}' \left(\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\right) \\ & - O\left(\|\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\|^2\right). \end{aligned}$$

To use this bound numerically, assume that $\nabla_{\theta} Q(\theta)|_{\theta=\theta(\hat{\theta};0)}$ is bounded and that

$$O\left(\|\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\|^2\right) \gg Q(\hat{\theta}; \varepsilon_{in}) - Q(\theta(\hat{\theta}; 0)) + \nabla_{\theta} Q(\theta)|_{\theta=\theta(\hat{\theta};0)}' \left(\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\right).$$

This allows us to focus on the numerical error from the NFP algorithm's inner loop and the bias in objective values. We also know from the choice of the contraction mapping inner loop tolerance that $\|\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\| \leq \frac{L(\hat{\theta})}{1 - L(\hat{\theta})} \varepsilon_{in}$. Therefore, $\|\theta(\hat{\theta}; \varepsilon_{in}) - \theta(\hat{\theta};0)\|$ is also bounded. Hence, we obtain

$$O(\hat{\theta}^2) \gg Q(\hat{\theta}; \varepsilon_{in}) - Q(\theta(\hat{\theta}; 0)) + O\left(\frac{L(\hat{\theta})}{1 - L(\hat{\theta})} \varepsilon_{in}\right).$$

A.4 Proof of Theorem 6

The NFP method (4) solves the following unconstrained problem

$$\min_{\theta} Q(\theta(\theta)). \tag{18}$$

The first-order condition of (18) is

$$\frac{\partial Q(\theta(\theta))}{\partial \theta} = \frac{d\theta'}{d\theta} \frac{\partial Q}{\partial \theta} = 0. \tag{19}$$

The constrained optimization formulation of (18) is

$$\begin{aligned} \min_{(\theta;\beta)} \quad & Q(\theta) \\ \text{s.t.} \quad & s(\theta; \beta) = S. \end{aligned} \tag{20}$$

The Lagrangian for (20) is $L(\ ;\ ;\ ) = Q(\ )\quad {}^{T}(S\quad s(\ ;\ ))$, where is the vector of Lagrange multipliers. The first-order conditions of (20) are

$$\begin{aligned}
\frac{@L(\ ;\ ;\ )}{@} &= \frac{ds(\ ;\ )}{d}^{0} = 0 \\
\frac{@L(\ ;\ ;\ )}{@} &= \frac{@Q}{@}\quad \frac{ds(\ ;\ )}{d}^{0} = 0 \\
\frac{@L(\ ;\ ;\ )}{@} &= S\quad s(\ ;\ ) = 0:
\end{aligned} \tag{21}$$

$$\begin{aligned} \min_{;} \quad & g(\ )^{0} W g(\ ) \\ \text{subject to} \quad & s(\ ;\ ) = S \end{aligned} : \tag{23}$$

$$\text{where } g(\ ) = \frac{1}{T} \bar{X}_{t=1} {}^{0}_{t} z_t$$

Gradients for MPEC

$$\frac{@s_j(\ _t;\ )}{@\ \bar{}_k} = \int T_j(\ _t;\ ;\ )(x_{j;k;t} - \sum_i T_i(\ _t;\ ;\ )x_{k;i;t}) dF(\ )$$

$$\frac{@s_j(\ _t;\ )}{@\ \bar{}} = \int T_j(\ _t;\ ;\ )(p_{j;k;t} - \sum_i T_i(\ _t;\ ;\ )p_{k;i;t}) dF(\ )$$

$$\frac{@s_j(\ _t;\ )}{@\ _k} = \int T_j(\ _t;\ ;\ )(x_{j;k;t} - \sum_i T_i(\ _t;\ ;\ )x_{k;i;t})\ _k dF(\ )$$

$$\frac{@s_j(\ _t;\ )}{@} = \int T_j(\ _t;\ ;\ )(p_{j;k;t} - \sum_i T_i(\ _t;\ ;\ )p_{k;i;t})\ _{K+1} dF(\ )$$

$$\frac{@s_j(\ _t;\ )}{@\ _{j;t}} = \int T_j(\ _t;\ ;\ )(1 - T_j(\ _t;\ ;\ ))dF(\ )$$

$$\frac{@s_j(\ _t;\ )}{@\ _{i;t}} = \int -T_j(\ _t;\ ;\ ))T_i(\ _t;\ ;\ ))dF(\ )$$

$$\frac{@g(\ )^{0} W g(\ )}{@} = 2g(\ )^{0} W \frac{@g(\ )}{@}$$

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

$$R(p;\ ) = {}^{r^0}(p) \quad :::$$
$$\int \log \exp {}^{r^0}(p +) + \sum_j \exp {}^{r}_{j} {}^{r}(p^0{}_j +) + {}_j \quad dF_{\ ;}(\ ; \ ) \Bigg)^{!} : \tag{24}$$

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, $\ ;$ that set the following expression to zero:

$$X^0 R(p;\ ) = 0$$

and computing the lower block of the Jacobian as

$$J_{t;\ !\ S} = \left[\frac{@G}{@\ _t}\right]^{1}\left[\frac{@G}{@S_t}\right] ; = \left[\frac{@s}{@\ _t}\right]^{1}$$

where the $(j;k)$ element of $\frac{@s_{j;t}}{@\ _{k;t}}$ is

$$\frac{@S_{j;t}}{@\ _{k;t}} = \begin{cases} \sum_r {}_{r;t} s_j (p_t;\ _t;\ ^r)(1 \quad s_j (p_t;\ _t;\ ^r)) & ; \text{ if } j = k \\ \sum_r {}_{r;t} s_j (p;\ _t;\ ^r)\, s_k (p;\ _t;\ ^r) & ; \text{ otherwise:} \end{cases}$$