Journal of Economic Theory and Econometrics, Vol. 28, No. 4, Dec. 2017, 1–21

Testing the Nested Fixed-Point Algorithm in BLP Random Coefficients Demand Estimation*

Jinhyuk Lee Kyoungwon Seo

Abstract This paper examines the numerical properties of the nested fixed-point algorithm (NFP) using Monte Carlo experiments in the estimation of Berry, Levinsohn, and Pakes's (1995) random coefficient logit demand model. We find that in speed, convergence and accuracy, nested fixed-point (NFP) approach using Newton's method performs well like a mathematical programming with equilibrium constraints (MPEC) approach adopted by Dubé, Fox, and Su (2012).

Keywords random coefficients logit demand; numerical methods; nested fixed-point algorithm; Newton's method

JEL Classification C1, L1

*Corresponding author: Kyoungwon Seo, Business School, Seoul National University, Seoul, South Korea, seo8240@snu.ac.kr. Jinhyuk Lee is at Department of Economics, Korea University, Seoul, South Korea, jinhyuklee@korea.ac.kr. Seo gratefully acknowledges the financial support of the SNU Invitation Program for Distinguished Scholar, the Institute of Finance and Banking, and the Institute of Management Research at Seoul National University. Lee's work is supported by a Korea Unversity Grant (K1613461).

Received May 22, 2017, Revised September 19, 2017, Accepted September 21, 2017

1. INTRODUCTION

The nested fixed-point approach (NFP) has been widely used for random coefficients logit demand model since Berry, Levinsohn, and Pakes (1995; BLP hereafter). To estimate structural parameters, NFP iterates two nested loops: The inner loop numerically solves a fixed-point problem given a parameter value, and the outer loop optimizes the objective function over a parameter space. Dubé, Fox, and Su (2012; DFS hereafter) argue that numerical errors from the inner loop propagate into the objective function, which leads to less convergence or to convergence to a point that is not a local minimum if a loose stopping criterion is used. As an alternative to NFP, DFS propose a mathematical programming with equilibrium constraints (MPEC) approach for the BLP estimation. There is no nested inner loop in MPEC, and thus it does not suffer from the error propagation problem and can be faster. DFS show that MPEC performs better than NFP in terms of accuracy and speed.

This paper examines the actual numerical performance of NFP in the estimation of BLP model using Monte Carlo experiments while Lee and Seo (2016; hereafter LS) theoretically derive the error bound of the estimate computed by NFP. In this paper, we test whether DFS's argument mentioned above or the implication of the theoretical results in LS can be confirmed in the quasi-real setting.

As in LS, we study two versions of NFP. The first is BLP's nested contraction mapping algorithm, which DFS also study. We refer to this algorithm as NFP/CTR. Our second version of NFP, which we refer to as NFP/NT, uses Newton's method to solve the inner fixed-point problem. In NFP/NT, we combine contraction mapping iterations and Newton's method, following Rust (1987) to ensure global convergence. The original version of Newton's method converges locally only. One way to guarantee global convergence is to start with contraction mapping iterations and then switch to Newton's method. A similar implementation to ours appears in Iskhakov et al. (2016), who estimate the single-agent dynamic discrete choice model. They show that NFP with Newton's method is as fast as MPEC.

LS show that NFP/CTR has the same order of the bound on the estimate error as MPEC if the tolerances of NFP/CTR are set to the same levels as those of MPEC. Their argument implies that on convergence, the numerical accuracy of NFP/CTR would be close to that of MPEC. They also argue that in theory, NFP/NT has good numerical properties. First, due to the quadratic convergence rate of Newton's method, only a few more iterations are necessary to reduce the inner loop error, for example, from 10^{-6} to 10^{-12}. In addition, the quadratic

convergence rate of Newton's method enables NFP/NT to have less inner loop error than NFP/CTR under the same inner loop tolerance. These properties make it easy to minimize the error propagation into the outer loop objective function, and thus the inner loop tolerance does not need to be loosened. With a tight tolerance, an optimization routine is more likely to converge to a local optimum when using NFP/NT. Second, the numerical error in the parameter estimate of NFP/NT is smaller than that of NFP/CTR. They show that the error bound of NFP/NT has an order of the square of the inner loop tolerance while NFP/CTR has an order of the inner loop tolerance.

We illustrate the actual numerical performance of NFP in our Monte Carlo experiments. We vary the level of inner loop tolerance but use a tight outer loop tolerance, 10^{-6}, to observe the proportion of convergence and the accuracy of the estimates. As in DFS, we find many cases of nonconvergence of NFP/CTR when using a loose inner loop tolerance. We also study the numerical error of the NFP/CTR estimate conditional on convergence. The experiment shows that, when NFP/CTR converges, the algorithm finds the estimates properly even under a loose inner loop tolerance. We show that NFP/NT converges more often than NFP/CTR and achieves a tighter inner loop tolerance with a few more iterations, as expected. For numerical accuracy, we find that on convergence, NFP/CTR performs similar to NFP/NT. This seems to contradict the theoretical finding in LS that NFP/NT has a smaller upper bound of the estimate error than NFP/CTR because NFP/NT has a smaller inner loop error.[1] A potential reason may be related to an outer loop error because an outer loop error becomes dominant as an inner loop error gets smaller. Thus, the different inner loop errors seem to make no noticeable difference between the estimates of NFP/NT and NFP/CTR. In the other experiment, we check the computational time of each approach varying the level of mean intercept, and the number of markets, of products and of simulation draws. As DFS argue, NFP/CTR is affected by the value of Lipschitz constant whereas NFP/NT is not. We also find that NFP is comparable to MPEC in most cases.

Our implementation of NFP/NT is not the first to use Newton's method for the inversion of the market share equations in the BLP estimation. Patel (2012) proposes Newton's method as a supplement to contraction mapping when contraction mapping does not perform efficiently. Reynaerts, Varadhan, and Nash (2012) propose Newton's method as an alternative to the contraction mapping iterations. Houde (2012) proposes combining a quasi-Newton method with the contraction mapping iterations. However, none of these studies discusses the

[1] See Corollary 2 and the corresponding equation in Subsection 3.2 in Lee and Seo (2016).

effect of the change in tolerance on the numerical error in the estimate.

We organize the rest of this paper as follows. In Section 2, we discuss a BLP model and a data-generating process, and present the estimation procedures, NFP/CTR and NFP/NT. Section 3 provides the Monte Carlo experiments. Section 4 concludes. Finally, we discuss technical details of our implementation of NFP/NT in Appendix A.

2. MODEL AND ESTIMATION

In this section, we briefly explain the random coefficient logit demand model using aggregate data and the data-generating process for the Monte Carlo experiment. Then, we discuss NFP/CTR and NFP/NT estimation procedures.

2.1. BLP MODEL AND DATA-GENERATING PROCESS

To make comparison easier, we use the same setup and data-generating process as DFS. We assume a set of independent markets, $t = 1,...,T$. For simplicity, we assume each market to have the same set of products, $j = 1,...,J$. No purchase in market t is denoted $j = 0$. In the experiment, there are 50 markets (T) and the same 25 products (J) in each market. The utility of consumer i from consuming product j in market t is

$$U_{ijt} = X_{jt}\beta_i + \xi_{jt} + \varepsilon_{ijt} \text{ and } U_{i0t} = \varepsilon_{i0t}, \tag{1}$$

where X_{jt} is a vector of observed product characteristics, β_i is a vector of consumer i's preference for observed product characteristics, ξ_{jt} is a product characteristic or a demand shock that is unobserved by the econometrician, and ε_{ijt} is an idiosyncratic shock.

For the experiment, let $X_{jt} = \{1, x_{j1}, x_{j2}, x_{j3}, p_{jt}\}$ where $\{x_{j1}, x_{j2}, x_{j3}\}$ are observed market invariant characteristics following the multivariate normal distribution with zero means and the covariance matrix:

$$\begin{bmatrix} 1 & -0.8 & 0.3 \\ -0.8 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix}.$$

The price p_{jt} is generated as follows:

$$p_{jt} = 3 + 1.5 \cdot \xi_{jt} + u_{jt} + \sum_{m=1}^{3} x_{jm}, \tag{2}$$

where ξ_{jt} follows the i.i.d. standard normal distribution $N(0,1)$ and u_{jt} follows the uniform distribution $U[0,1]$.

From the assumption that ε_{ijt} follows the Type I extreme value distribution, the probability of consumer i purchasing product j is

$$\frac{\exp\left(X_{jt}\beta_i+\xi_{jt}\right)}{1+\sum_{j'=1}^{J}\exp\left(X_{j't}\beta_i+\xi_{j't}\right)}. \tag{3}$$

Following the standard random coefficient logit demand model, we further assume that the vector of random coefficients β_i is drawn independently from the distribution $F(\beta_i;\theta)$. For the experiment, let $\beta_i=\left\{\beta_i^0,\beta_i^1,\beta_i^2,\beta_i^3,\beta_i^p\right\}$ where each is distributed independently normal with $E[\beta_i]=\{0,1.5,1.5,0.5,-3\}$ and $Var[\beta_i]=\{0.5,0.5,0.5,0.5,0.2\}$.

Under the assumptions, the market share function of product j in market t is

$$s_j(X_t,\xi_t;\theta)=\int\frac{\exp\left(X_{jt}\beta_i+\xi_{jt}\right)}{1+\sum_{j'=1}^{J}\exp\left(X_{j't}\beta_i+\xi_{j't}\right)}dF(\beta_i;\theta), \tag{4}$$

where $X_t\equiv(X_{1t}',...,X_{Jt}')'$ and $\xi_t\equiv(\xi_{1t},...,\xi_{Jt})'$. In the experiment, we approximate the integral using simulation; that is, we generate $1,000$ draws of β_i and then evaluate (4) by calculating

$$\frac{1}{1,000}\sum_{i=1}^{1,000}\frac{\exp\left(X_{jt}\beta_i+\xi_{jt}\right)}{1+\sum_{j'=1}^{J}\exp\left(X_{j't}\beta_i+\xi_{j't}\right)}. \tag{5}$$

We use the same 1,000 draws to compute market share in both the data-generating process and the estimation to prevent the simulation errors.

For simplicity, we write $s_j(\xi_t;\theta)$ instead of $s_j(X_t,\xi_t;\theta)$. We define $s(\xi_t;\theta)\equiv(s_1(\xi_t;\theta),...,s_J(\xi_t;\theta))'$ as the predicted market share functions in market t, and $S_t\equiv(S_{1t},...,S_{Jt})'$, where S_{jt} is the observed market share of product j in market t.

2.2. ESTIMATION PROCEDURES

BLP estimation procedure consists of inner loop and outer loop.

Inner loop: BLP invert the market share equations $S_t=s(\xi_t;\theta)$ for a given θ, and obtain the solution, denoted as $\xi_t(\theta)\equiv(\xi_{1t}(\theta),...,\xi_{Jt}(\theta))'$. To obtain $\xi_t(\theta)$, BLP suggest the following contraction mapping iterations:

$$\xi_{CTR}^{h}=\xi_{CTR}^{h-1}+\ln S_t-\ln s\left(\xi_{CTR}^{h-1};\theta\right),\quad h=1,2,\cdots \tag{6}$$

such that

$$\left\|\xi_{CTR}^{h+1}-\xi_{CTR}^{h}\right\| \leq L(\theta)\left\|\xi_{CTR}^{h}-\xi_{CTR}^{h-1}\right\|, \tag{7}$$

with a Lipschitz constant $L(\theta) \in [0,1)$. They show that iterative applications of the contraction mapping converge to $\xi_t(\theta)$. This is what we refer to as NFP/CTR.

As an alternative, LS adopt NFP/NT, which starts with contraction mapping iterations and switches to Newton's method to ensure global convergence. Newton's method in NFP/NT begins with an initial guess ξ_{NT}^{0} obtained by the contraction mapping, and the subsequent iterate is computed using the iteration rule:

$$\xi_{NT}^{h}=\xi_{NT}^{h-1}+\left[\nabla_{\xi} s\left(\xi_{NT}^{h-1};\theta\right)\right]^{-1}\left[S_t-s\left(\xi_{NT}^{h-1};\theta\right)\right], \quad h=1,2,\cdots \tag{8}$$

where $\nabla_{\xi} s\left(\xi_{NT}^{h-1};\theta\right)$ is the matrix of first partial derivatives of $s(\xi;\theta)$ with respect to ξ at the iterate ξ_{NT}^{h-1}. LS show that NFP/NT has the following property:

$$\left\|\xi_{NT}^{h+1}-\xi_{NT}^{h}\right\| \leq \frac{\kappa}{2\sqrt{1-2\rho}}\left\|\xi_{NT}^{h}-\xi_{NT}^{h-1}\right\|^{2}, \tag{9}$$

where $\frac{\kappa}{2\sqrt{1-2\rho}}$ is not too large. This implies that Newton's method converges quadratically to a local solution, whereas contraction mapping iterations converge linearly.

When we solve for $\xi(\theta)$ in the inner loop, the exact value of the solution is not available. Instead, we impose a stopping rule of the inner loop, which requires the change of two successive iterates to be less than a given inner loop tolerance, ε_{in}:

$$\left\|\xi^{h}-\xi^{h-1}\right\| \leq \varepsilon_{in}. \tag{10}$$

In the experiment, we vary the level of inner loop tolerance from 10^{-3} to 10^{-12} and see its effect on the convergence and numerical accuracy of the estimate.

Outer loop: To address the price endogeneity, BLP propose the generalized method of moments (GMM) estimation with the moment conditions

$$E\left[\xi_{jt} Z_{jt}\right]=0, \tag{11}$$

where Z_{jt} is a vector of instrumental variables. The sample moment is

$$g_T(\xi(\theta))=\frac{1}{T}\sum_{t=1}^{T} g(\xi_t(\theta))=\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\xi_{jt}(\theta) Z_{jt}, \tag{12}$$

where $\xi(\theta) \equiv \left(\xi_1(\theta)', ..., \xi_T(\theta)'\right)'$ and should be close to 0 when T is large. In the experiment, the 42 instruments Z_{jt} are generated as follows: $(1,1,...,1)'$, $x_{jm}, z_{jtd}, x_{jm}^2, z_{jtd}^2, x_{jm}^3, z_{jtd}^3$, $\prod_{m=1}^3 x_{jm}, \prod_{d=1}^6 z_{jtd}, z_{jtd}x_{j1}$ and $z_{jtd}x_{j2}$, where $z_{jtd} \sim \varepsilon_{jt} + \frac{1}{4}\left(e_{jt} + 1.1\sum_{m=1}^3 x_{jm}\right)$ for $d = 1,2,...,6$, $\varepsilon_{jt} \sim U[0,1]$ and $e_{jt} \sim N(0,1)$.

The BLP GMM problem is as follows:

$$\min_{\theta\in\Theta} Q(\xi(\theta)) = \min_{\theta\in\Theta} g_T(\xi(\theta))' W g_T(\xi(\theta)), \tag{13}$$

where the weight matrix $W = (Z'Z)^{-1}$ and $Z = \left\{Z_{jt}\right\}_{j,t}$ is the $(JT \times 42)$ matrix of instruments. The BLP estimator θ^* is the minimizer of the BLP GMM problem in the finite sample. In the outer loop, we use 10^{-6} for its tolerance.

3. MONTE CARLO SIMULATION

In the Monte Carlo simulation, we implement three approaches: MPEC, NFP/CTR, and NFP/NT. We generate synthetic data sets under the same setting used by DFS in section 2.

We ran all experiments on a computer with a CPU Intel Xeon processor E3 1270 v2, 16GB of RAM, Windows 7 64 bit, MATLAB R2012a, and TOMLAB 7.9 equipped with KNITRO 8.0. We use the codes that DFS provide in *Econometrica* after some modifications. We use the interior point algorithm (for the outer loop) with algorithm option ALG=1 in KNITRO 8.0. We provide NFP with the analytic Jacobian/gradient and Hessian. We use 10^{-6} for the outer loop (optimality) tolerance. We use the same starting points for the implementations of the three approaches.

For NFP/NT, we use a combination of contraction mapping and Newton iterations because Newton's method does not converge globally but contraction mapping iterations do. Our approach is similar to Rust's (1987) strategy in that the inner loop begins with the contraction mapping iterations and switches to Newton's method after the contraction mapping converges under a loose tolerance. However, the way we deal with numerical instability is different from Rust (2000).[2]

[2]Theoretical results in LS on convergence rate of numerical errors still apply to our implemention of modified Newton's method. Recall that LS assume convergence. Conditional on convergence, the last several guesses are close to the solution and by the local convergence property of Newton's method, the original Netwon steps work well on these guesses. Our modifications come into play only if the original Newton step fails to find a reasonable direction. Thus when the inner loop converges, the termination condition of the original Newton method is satisfied.

There are issues with the numerical instability of NFP/NT. For example, Newton's method does not perform well if the initial guess is not close enough to the solution, or the derivative of $s(\xi;\theta)$ with respect to ξ in (8) may be numerically ill-posed, in which case an inaccurate Newton step may be obtained. We discuss how we handle these problems in detail in Appendix A.[3]

3.1. TESTABLE STATEMENTS

In the experiments, we seek to test the statement argued by DFS and the implications of the theoretical findings in LS. We hypothesize them as follows:

DFS argue that:

(H1) (i) Numerical errors from the inner loop of NFP/CTR propagate into the objective function, (ii) which leads to less convergence.

(H2) Loose inner loop tolerances of NFP/CTR can lead to incorrect parameter estimates.

(H3) MPEC is much faster than NFP/CTR in their experiments.

The followings are the implications in LS:

(H4) If the tolerances of NFP/CTR are set to the same levels as those of MPEC, the upper bounds on the estimate errors for both approaches are of the same order. This implies that on convergence, the numerical accuracy of NFP/CTR would be close to that of MPEC.

(H5) Given the same level of inner loop tolerance, NFP/NT converges more often in the outer loop than NFP/CTR because NFP/NT has smaller actual inner loop error.

(H6) Due to the quadratic convergence rate of Newton's method, only a few more iterations are necessary for NFP/NT to reduce the inner loop error, which implies less computational time for NFP/NT.

(H7) NFP/NT obtains a smaller upper bound of the estimate error than NFP/CTR. This implies that the estimate of NFP/NT is more accurate than that of NFP/CTR under the same set of tolerances.

[3] As Iskhakov et al. (2015) discuss, if a researcher uses MPEC (e.g., KNITRO) instead of NFP, an effort to customize the programming code may be reduced or avoided.

Table 1: Monte Carlo Results Varying the Inner Loop Tolerance

Inner loop tolerance	Approach	Runs conv.	CPU time (seconds)	Major iter.	Func. eval.	Grad/Hess. eval.	CTR. iter.	NT. iter.
10^{-3}	NFP/CTR	14 %	92.9	26	114	52	530	
	NFP/NT	7 %	91.6	26	88	54	118	32
10^{-6}	NFP/CTR	16 %	47.1	11	65	24	700	
	NFP/NT	80 %	37.1	10	26	23	53	42
10^{-9}	NFP/CTR	60 %	46.9	10	31	22	991	
	NFP/NT	100 %	33.1	9	13	21	23	48
10^{-12}	NFP/CTR	99 %	45.3	9	14	21	1,147	
	NFP/NT	100 %	33.2	9	13	21	36	53

Notes: We generate 20 data sets for each level of inner loop tolerance with $E\left[\beta_i^0\right] = 2$, $T = 50$, $J = 25$ and $n_s = 1,000$. We use five starting points in the estimation for each data set. The reported means are over 100 runs. The analytic Jacobian/gradient and the Hessian are provided to NFP and MPEC. Major iter. is the number of iterations in the outer loop. Func. eval. is the number of function evaluations in the outer loop. Grad/Hess. eval. is the number of gradient and Hessian evaluations in the outer loop. CTR. iter. is the number of contraction mapping iterations in the inner loop. NT. iter. is the number of Newton iterations in the inner loop.

(H8) NFP/NT has speed advantage over NFP/CTR because NFP/NT has the quadratic convergence rate and is not affected by the level of Lipschitz constant.

3.2. RESULTS OF MONTE CARLO EXPERIMENTS

In brief, (H2), (H3) and (H7) are disproved while other hypotheses are supported. See the details in the following.

In the first experiment, we examine how the level of inner loop tolerance affects convergence and numerical accuracy. We report the results in Tables 1 and 2. This experiment is used to check convergence and accuracy of NFP.

We check the error propagation and convergence of NFP. We report the numbers of iterations in the outer loop and in the inner loop in Table 1. For both NFP/NT and NFP/CTR, as the inner loop tolerance gets looser, the numbers of iterations in the outer loops (i.e., major iterations, function evaluations, and gradient and Hessian evaluations) increase, but the numbers of iterations in the inner loop (i.e., contraction mapping iterations and Newton iterations) decrease.

This implies that a large inner loop error propagates into the outer loop and thus prevents the outer loop from converging, resulting in more iterations in the outer loop, which DFS also point out. (H1) (i) is supported.

We also report the rate of runs that converged for each method in Table 1. The third column in the table shows that NFP/CTR does not converge in many cases for the loose tolerances of 10^{-3} through 10^{-9}, as DFS also show. In particular, for 10^{-3}, only 14 runs out of 100 converged. However, for 10^{-12}, the algorithm converges in almost all runs. Therefore, (H1) (ii) is supported.

Next, we measure the numerical accuracy of NFP. For this, we need the theoretically exact estimate in the finite sample. Because the numerical error-free solution of the BLP GMM problem is not available, we assume that the MPEC estimates are the exact estimates. The optimality and feasibility tolerances for MPEC are set to a very tight level, 10^{-10}, to obtain an estimate as accurate as possible. DFS use 10^{-6} in their experiments. To simplify an order of magnitude of discrepancy, we take the logarithm (base 10) of the absolute deviations of the NFP estimates from the MPEC estimates.

Table 2 reports the mean and maximum of the log absolute deviations for each level of inner loop tolerance. Both the NFP and the MPEC estimates are the minimizers among the local minima for each of 20 data sets.[4] We find that for all data sets with convergence reported, the absolute deviations between the estimates of NFP/CTR and MPEC are, on average, less than the optimality tolerance, 10^{-6}. Furthermore, the maximum of the log deviations ranges from -5.94 to -7.92. Even for $\varepsilon_{in} = 10^{-3}$, the maximum log deviation is -5.94, which is small. According to Theorem 1 in LS, the upper bound of the deviation of NFP/CTR has an order of 10^{-3} when $\varepsilon_{in} = 10^{-3}$. We conjecture that there might exist a sharper upper bound than what Theorem 1 suggests or that actual deviations do not always achieve the upper bound. With a tight outer loop tolerance, (H2) is disproved and (H4) is supported because the estimates of NFP/CTR and MPEC are very similar.

[4]There is one case in which NFP and MPEC converge to different local minimizers, which we report as not convergent.

Table 2: Log deviation of NFP SD estimates from MPEC estimates

Inner loop tolerance	Approach	Datasets conv.	True values:	Log dev. of estimates from MPEC 0.7071	0.7071	0.7071	0.7071	0.4472	All
10^{-3}	NFP/CTR	30 %	Avg	-9.34	-7.53	-7.40	-7.86	-7.35	-7.90
			Max	-5.94	-6.76	-7.09	-6.96	-6.97	-5.94
	NFP/NT	20 %	Avg	-13.24	-8.16	-7.56	-7.89	-7.34	-8.84
			Max	-6.47	-7.38	-7.26	-7.55	-7.04	-6.47
10^{-6}	NFP/CTR	40 %	Avg	-11.29	-8.22	-8.09	-8.62	-7.93	-8.83
			Max	-7.06	-7.39	-7.58	-7.88	-7.57	-7.06
	NFP/NT	100 %	Avg	-8.46	-9.05	-8.75	-9.01	-8.57	-8.77
			Max	-6.18	-7.54	-7.96	-7.54	-7.34	-6.18
10^{-9}	NFP/CTR	85 %	Avg	-8.85	-8.72	-8.62	-9.13	-8.79	-8.82
			Max	-6.92	-7.87	-8.12	-8.10	-8.09	-6.92
	NFP/NT	100 %	Avg	-9.28	-9.39	-9.14	-9.72	-9.01	-9.31
			Max	-6.47	-7.87	-8.19	-7.81	-8.37	-6.47
10^{-12}	NFP/CTR	100 %	Avg	-9.37	-9.63	-9.27	-9.81	-9.08	-9.43
			Max	-7.92	-8.45	-8.60	-8.43	-8.38	-7.92
	NFP/NT	100 %	Avg	-9.55	-9.79	-9.32	-9.87	-9.10	-9.52
			Max	-7.20	-8.48	-8.52	-8.70	-8.37	-7.20

Notes: For each tolerance level, there are 20 data sets as described in Table 1. The local minimum with the lowest objective value is determined to be the global minimum for each data set. Absolute deviations between the estimates of NFP and MPEC are calculated, and the log with base 10 of the deviation is shown.

Next, we measure the numerical accuracy of NFP. For this, we need the theoretically exact estimate in the finite sample. Because the numerical error-free solution of the BLP GMM problem is not available, we assume that the MPEC estimates are the exact estimates. The optimality and feasibility tolerances for MPEC are set to a very tight level, 10^{-10}, to obtain an estimate as accurate as possible. DFS use 10^{-6} in their experiments. To simplify an order of magnitude of discrepancy, we take the logarithm (base 10) of the absolute deviations of the NFP estimates from the MPEC estimates.

Table 2 reports the mean and maximum of the log absolute deviations for each level of inner loop tolerance. Both the NFP and the MPEC estimates are the minimizers among the local minima for each of 20 data sets.[5] We find that for all data sets with convergence reported, the absolute deviations between the estimates of NFP/CTR and MPEC are, on average, less than the optimality tolerance, 10^{-6}. Furthermore, the maximum of the log deviations ranges from -5.94 to -7.92. Even for $\varepsilon_{in} = 10^{-3}$, the maximum log deviation is -5.94, which is small. According to Theorem 1 in LS, the upper bound of the deviation of NFP/CTR has an order of 10^{-3} when $\varepsilon_{in} = 10^{-3}$. We conjecture that there might exist a sharper upper bound than what Theorem 1 suggests or that actual deviations do not always achieve the upper bound. With a tight outer loop tolerance, (H2) is disproved and (H4) is supported because the estimates of NFP/CTR and MPEC are very similar.

Tables 1 and 2 show that a loose inner loop tolerance of NFP/CTR sacrifices convergence. However, when convergence is achieved under a tight outer loop tolerance such as 10^{-6}, the NFP/CTR deviations from MPEC are very small even at the loose inner loop tolerances. Thus, it seems that the level of inner loop tolerance significantly affects convergence rather than accuracy of the estimate. This result is consistent with the analysis of the error bound in Corollary 2 in LS that assume convergence. DFS do not investigate the combination of a tight outer loop tolerance, a loose inner loop tolerance, and the optimization routine reporting convergence simply because this never happens in their examples in Tables I and II.[6]

[5]There is one case in which NFP and MPEC converge to different local minimizers, which we report as not convergent.

[6]It is worth comparing Table 2 with Tables I and II in DFS. First, the purpose of the latter tables is not to illustrate their error bound derived in their Theorem 3, but instead to illustrate consequences of failure to find local optima of the objective function due to a loose inner loop tolerance. Our purpose of the table is to illustrate the error bound of the estimate on convergence. Second, the numbers in the tables may appear inconsistent, in particular for NFP/CTR with a loose inner loop tolerance. For example, the fraction of convergence in our table is 30% for the tolerance of 10^{-3}, while that in Tables I and II in DFS is 0. The maximum absolute deviation for a tolerance

Table 3: Monte Carlo Results Varying the Lipschitz Constant

Mean intercept	Approach	Runs conv.	CPU Time (seconds)	Major iter.	Func. eval.	Grad/Hess. eval.	CTR. iter.	NT. iter.
-2	MPEC	95 %	86.8	23	45	49	-	-
	NFP/CTR	100 %	39.6	11	16	24	302	-
	NFP/NT	100 %	39.6	11	16	24	36	50
-1	MPEC	99 %	63.8	17	26	37	-	-
	NFP/CTR	100 %	40.0	11	16	23	403	-
	NFP/NT	100 %	38.8	11	16	23	38	53
0	MPEC	98 %	61.9	16	29	35	-	-
	NFP/CTR	98 %	40.7	10	16	23	584	-
	NFP/NT	100 %	37.2	10	15	22	38	54
1	MPEC	98 %	61.7	17	28	35	-	-
	NFP/CTR	100 %	42.0	10	14	21	859	-
	NFP/NT	100 %	37.2	10	14	22	40	52
2	MPEC	97 %	58.4	16	26	34	-	-
	NFP/CTR	99 %	45.3	9	14	20	1,315	-
	NFP/NT	100 %	33.2	9	13	20	35	49
3	MPEC	99 %	50.3	14	19	29	-	-
	NFP/CTR	97 %	58.6	9	14	20	2,515	-
	NFP/NT	100 %	31.9	9	12	19	29	46
4	MPEC	98 %	55.6	15	25	32	-	-
	NFP/CTR	93 %	88.1	9	16	20	5,010	-
	NFP/NT	100 %	32.2	9	13	20	33	51

Notes: We generate 20 datasets for each level of mean intercept $E\left[\beta_i^0\right]$ with $T = 50$, $J = 25$ and $n_s = 1,000$. We use five starting points in the estimation for each dataset. The reported means are over 100 runs. The analytic Jacobian/gradient and the Hessian are provided to NFP and MPEC.

Now we compare NFP/CTR with NFP/NT. First, Table 1 shows that NFP/NT converges better than NFP/CTR under the same level of inner loop tolerance, except 10^{-3}. NFP/NT converges 80% for a tolerance of 10^{-6} and 100% for tighter tolerances. The only difference between NFP/NT and NFP/CTR in the implementations is that they use different numerical algorithm for the inner loop: Newton's method vs. contraction mappings. This implies that NFP/NT converges more often than NFP/CTR under the same inner loop tolerance because NFP/NT has smaller inner loop errors. (H5) is supported.

Second, Table 1 reports that, for the inner loop tolerances of 10^{-12}, NFP/NT and NFP/CTR have almost the same numbers of iterations in the outer loop, but there is a drastic difference in the numbers of iterations in the inner loop. For example, with $\varepsilon_{in} = 10^{-12}$, both NFP/NT and NFP/CTR have, on average, 9 major iterations in the outer loop, but NFP/NT has 89 iterations (36 contraction mapping iterations and 53 Newton's iterations) in the inner loop while NFP/CTR has $1,147$ contraction mapping iterations. The small number of iterations in the inner loop of NFP/NT is the main driver for speeding up the estimation procedure as in Iskhakov et al. (2015). Furthermore, as we tighten the inner loop tolerance from 10^{-9} to 10^{-12}, the total number of iterations in the inner loop until the outer loop converges increases by 18 (13 contraction mapping iterations and 5 Newton's iterations) for NFP/NT, while the number of contraction mapping iterations increases by 156 for NFP/CTR. Because the objective function is evaluated 13 times until convergence, NFP/NT needs only one or two more inner loop iterations for each objective function evaluation to achieve a tighter inner loop tolerance 10^{-12} from 10^{-9}. This is consistent with the quadratic convergence rate of Newton's method. (H6) is supported.

Third, Table 2 shows that the estimate errors of NFP/NT seem close to those of NFP/CTR while LS show that NFP/NT produces smaller estimate errors. A potential reason may be related to an outer loop error because an outer loop error becomes dominant as an inner loop error gets smaller. Thus, the different inner loop errors seem to make no noticeable difference between the estimates

---

of 10^{-3} is small in our table, whereas the difference (in the mean own-price elasticities) between loose (10^{-4}) and tight (10^{-14}) inner loop tolerances is large in their tables. One reason for these differences is that we supply the analytic gradient of the objective function in our experiment, and DFS do not in Table I. Because a numerical gradient generates an additional error, it explains the difference in the fractions of convergence. However, in Table II, DFS apply analytic derivatives but do not observe convergence. DFS use the pseudo-real cereal data set from Nevo (2000), which may be harder to optimize with a loose inner loop tolerance than the Monte Carlo data set that we use. Another reason for the difference is that we use the global minimum (i.e., the minimum among the local minima from the five starting points) for each data set, and they use all results from all the starting points, including the ones that did not converge.

of NFP/NT and NFP/CTR. (H7) is disproved.

In the second experiment, we alter the level of mean intercept $E\left[\beta_i^0\right]$, and also the number of markets T, of products J, and of simulation draws n_s. We examine (i) how much the speed of NFP is affected by the level of the Lipschitz constant; and (ii) how NFP performs in large datasets. We report the results in Tables 3 and 4. We replicate the experiment taken by DFS. Thus, we use 10^{-14} for the inner loop tolerance of NFP/CTR. MPEC uses 10^{-6} for both the optimality tolerance and the feasibility tolerance of its constraints. NFP/NT uses 10^{-10} for the inner loop tolerance.

We find in Table 3 that as DFS argue, the computational time of NFP/CTR is affected by the level of the Lipschitz constant. The computational time of NFP/CTR increases as the value of mean intercept increases, which increases the level of Lipschitz constant. However, the influence is much smaller in extent than what DFS find. We also find that NFP/CTR and NFP/NT are comparable in computational time to MPEC in most cases. We also report in Table 4 the Monte Carlo results for large data sets. In Table V in DFS that corresponds to our Table 4, NFP/CTR and MPEC were not even comparable – NFP/CTR was up to 42 times slower than MPEC. In our experiment, however, NFP/CTR performs comparable to MPEC and is faster in some cases due to our modifications. Thus, (H3) is disproved.

From the Table 3, NFP/NT seems unrelated to the Lipschitz constant. In all cases, NFP/NT saves the computational time compared to NFP/CTR. We find that NFP/CTR and NFP/NT perform almost identically to each other in terms of the number of major iterations and of function evaluations. However, NFP/NT dramatically reduces the number of contraction iterations for finding fixed-points while it adds a small number of Newton iterations. As a result, NFP/NT improves NFP/CTR in terms of the computational speed. We find a similar results in Table 4. Therefore, (H8) is supported.

Table 4: Monte Carlo Results Varying the Number of Markets, Products, and Simulation Draws

Markets T	Products J	Draws s	Approach	Runs conv.	CPU Time (seconds)	Major iter.	Func. eval.	Grad/Hess. eval.	CTR. iter.	NT. iter.
100	25	1000	MPEC	96 %	114	17	29	35	-	-
			NFP/CTR	84 %	154	10	23	21	3,722	-
			NFP/NT	100 %	71	10	14	21	38	54
250	25	1000	MPEC	92 %	370	21	37	45	-	-
			NFP/CTR	76 %	384	9	22	21	3,752	-
			NFP/NT	100 %	176	10	13	21	33	50
500	25	1000	MPEC	80 %	810	23	41	48	-	-
			NFP/CTR	88 %	535	9	19	21	1,732	-
			NFP/NT	100 %	359	10	14	21	30	49
100	25	3000	MPEC	88 %	535	27	53	55	-	-
			NFP/CTR	80 %	326	8	21	19	2,868	-
			NFP/NT	100 %	186	9	12	19	30	44
250	25	3000	MPEC	100 %	685	13	20	28	-	-
			NFP/CTR	92 %	1,000	10	16	21	2,821	-
			NFP/NT	100 %	513	9	13	21	36	49
25	100	1000	MPEC	100 %	116	14	18	29	-	-
			NFP/CTR	100 %	185	10	14	21	4,886	-
			NFP/NT	100 %	76	9	13	21	48	52
25	250	1000	MPEC	96 %	754	17	27	36	-	-
			NFP/CTR	100 %	1,381	9	12	19	12,695	-
			NFP/NT	100 %	356	9	13	19	59	67

Notes: We generate five datasets for each case with $E\left[\beta_i^0\right] = 2$. We use five starting points in the estimation for each dataset. The reported means are over 25 runs. The analytic Jacobian/gradient and the Hessian are provided to NFP and MPEC.

4. CONCLUSION

In this paper, we study the numerical performance of NFP/CTR and NFP/NT in BLP's random coefficients logit demand model. In an experiment with synthetic data, we demonstrate that NFP/CTR achieves reasonably accurate estimates on convergence even with a loose inner loop tolerance if a tight outer loop tolerance is applied. This implies that the error from the inner loop does not severely affect numerical accuracy of the estimate. However, a loose inner loop tolerance sacrifices convergence of NFP/CTR. As an alternative, we argue that NFP/NT can achieve a very tight inner loop tolerance rather easily. Therefore, NFP/NT performs well in numerical aspects such as convergence and accuracy. In the other experiment, we show that NFP/NT is not affected by the level of Lipschitz constant and can easily achieve a tight inner loop tolerance. Thus, NFP/NT performs well in terms of computational speed. Finally, we find that NFP/NT performs comparable to MPEC in speed, convergence and accuracy.

A. APPENDIX: ADDITIONAL TECHNIQUES FOR NEWTON'S METHOD

When we apply Newton's method for the inner fixed-point problems, computations may be unstable under some parameter values. As is known, numerically ill-posed equations may produce inaccurate solutions. We explain the techniques that we use to alleviate the numerical instability.

We introduce some notations. We omit the market index t assuming that there is only one market. We concentrate out the linear parameter as Nevo (2000) suggests. Then, the equation we need to solve in the inner loop is

$$S_j = \frac{1}{n_s}\sum_{i=1}^{n_s} \frac{\exp(\delta_j)\cdot\exp\left(\sum_{k=1}^{K} x_j^k \sigma_k v_i^k\right)}{1+\sum_{j'=1}^{J}\exp(\delta_{j'})\cdot\exp\left(\sum_{k=1}^{K} x_{j'}^k \sigma_k v_i^k\right)} \equiv s_j(\delta;\sigma), \text{ for } j=1,2,\cdots,J. \tag{A.1}$$

Here, S_j and $s_j(\delta;\sigma)$ are the observed market share and the predicted market share function of product j, respectively. The variable $X_{jt} = \left(x_{jt}^1,...,x_{jt}^K\right)$ is a K-dimensional vector containing the product characteristics, and $\beta_i = (\beta_{i1},...,\beta_{iK})$ is assumed to be of the form $\beta_{ik} = \overline{\beta}_k + \sigma_k v_{ik}$ with $v_{ik} \sim N(0,1)$. Only $\sigma = (\sigma_1,...,\sigma_K)$ is to be estimated in the concentrated GMM problem, and $\overline{\beta}_k$ will be estimated later. For each product j, $X_j\beta_i$ can be expressed by the sum of its mean utility $\delta_j = \sum_{k=1}^{K} x_j^k \overline{\beta}_k$ and the individual i's utility $\sum_{k=1}^{K} x_j^k \sigma_k v_i^k$ deviated from the mean utility. For each given σ, we need to solve for $\delta = (\delta_1,...,\delta_J)$.

1. Adjust the scale of the unknowns to be solved: Instead of solving for δ directly, we solve for $\omega = (\omega_1,...,\omega_J)$ where $\omega_j = \frac{\exp(\delta_j)}{\alpha_j}$, $j = 1,2,\cdots,J$ given some constant vector $\alpha = (\alpha_1,\cdots,\alpha_J)$. The equation is then,

$$S_j = \frac{1}{n_s}\sum_{i=1}^{n_s} \frac{\omega_j\cdot\exp\left(\sum_{k=1}^{K} x_j^k \sigma_k v_i^k\right)\cdot\alpha_j}{1+\sum_{j'=1}^{J}\omega_{j'}\cdot\exp\left(\sum_{k=1}^{K} x_{j'}^k \sigma_k v_i^k\right)\cdot\alpha_{j'}} \equiv \widetilde{s}_j(\omega;\sigma), \quad j=1,2,\cdots,J. \tag{A.2}$$

After solving for ω, it is easy to obtain δ. In our simulation, we choose $\alpha_j = S_j$ for each $j = 1,...,J$. Then, even for a very small value of S_j, such as around 10^{-9}, ω_j is not very small and we expect ω_j and $\omega_{j'}$ to be of a similar magnitude, unlike $\exp(\delta_j)$ and $\exp(\delta_{j'})$.

2. Adjust the scale of the matrix $\nabla_\omega \widetilde{s}(\cdot)$ if its inversion is ill-posed: The Newton step $\Delta\omega$ is computed as follows:

$$\Delta\omega = \left[\nabla_\omega \widetilde{s}(\cdot)\right]^{-1}\left[S - \widetilde{s}(\cdot)\right], \tag{A.3}$$

where $S = (S_1, \cdots, S_J)$, $\widetilde{s}(\cdot) = (\widetilde{s}_1(\cdot), \cdots, \widetilde{s}_J(\cdot))$ and $\nabla_\omega \widetilde{s}(\cdot)$ is the matrix of the first partial derivatives of $\widetilde{s}(\cdot)$ with respect to ω. If elements of $\nabla_\omega \widetilde{s}(\cdot)$ are badly scaled, the computation may be very inaccurate. To handle the bad scale problem, we use the identity

$$[\nabla_\omega \widetilde{s}(\cdot)]^{-1}[S - \widetilde{s}(\cdot)] = B[A\nabla_\omega \widetilde{s}(\cdot) B]^{-1} A[S - \widetilde{s}(\cdot)] \tag{A.4}$$

for any invertible $(J \times J)$ matrices A and B. We set A to be the identity matrix and B a diagonal matrix whose (j, j) element is $(\partial \widetilde{s}_j(\cdot)/\partial \omega_j)^{-1}$. The reason for adjusting the scale of the diagonal elements is that the matrix $\nabla_\omega \widetilde{s}(\cdot)$ is diagonally dominant (Lee and Seo 2015, Lemma 1). We check the condition number of $\nabla_\omega \widetilde{s}(\cdot)$ to decide whether to apply the above identity. Roughly speaking, if the condition number is 10^m, we may lose up to m digits of accuracy. In our simulation, we apply the above identity if the condition number is greater than 10^{15}.

3. Use the contraction iteration, if computing $B[\nabla_\omega \widetilde{s}(\cdot) B]^{-1}[S - \widetilde{s}(\cdot)]$ is still numerically ill-posed: Adjusting the scale of the matrix $\nabla_\omega \widetilde{s}(\cdot)$ does help obtain numerical stability but does not guarantee it. Thus, if the condition number of $\nabla_\omega \widetilde{s}(\cdot) B$ is still greater than 10^{15}, the computation may be still unreliable. In this case, we iterate contraction mappings 10 times and then try the Newton step again.

4. Adopt a line search: Even if the condition number of $\nabla_\omega \widetilde{s}(\cdot)$ is smaller than 10^{15}, the Newton step may not produce descent improvement if the current iterate is not close enough to the solution. This is a well-known problem of the original Newton iteration (see for example Nocedal and Wright 2006). A line search is one of the popular choices to solve this problem. Even though there may be more efficient line search algorithm in the literature, we believe pursuing efficiency of a line search is beyond this paper. Instead, we do the following simple line search: If the next Newton iterate $\omega^{\tau+1} = \omega^\tau + \Delta\omega \in \mathbb{R}^J$ contains 0 or a negative element, we believe that the Newton step is too large to be stable because the solution vector ω satisfies $\omega_j > 0$ for all $j = 1, ..., J$. If so, we take the next Newton step as follows:

$$\omega^\tau + a\Delta\omega \tag{A.5}$$

for some $a > 0$. We set

$$a = \tfrac{1}{2} / \max_j \frac{|\Delta\omega_j|}{|\omega_j^\tau|} \tag{A.6}$$

so that each component of $\omega^{a,\tau+1} \equiv \omega^{\tau} + a\Delta\omega$ is positive. But, since there is no guarantee that this process converges, we compute the contraction iterate, $\omega_j^{CTR,\tau+1} \equiv \omega_j^{\tau} S_j / \widetilde{s}_j(\omega^{\tau};\sigma)$ as well. Then, between these two iterates, we choose the one producing a smaller change of ω: If $\left\|\omega^{a,\tau+1} - \omega^{\tau}\right\| < \left\|\omega^{CTR,\tau+1} - \omega^{\tau}\right\|$, $\omega^{a,\tau+1}$ is chosen as the next iterate. Otherwise, $\omega^{CTR,\tau+1}$ is chosen and 10 contraction iterations are applied since the Newton step seems unstable at the current iterate.

5. Initial guess for ω in the next outer loop iteration: The value of a new parameter candidate is not expected to move very far from the current value in the outer loop, and thus it is common to use the current solution ω_σ for the current parameter σ as the initial guess for the market share equations in the next outer loop iteration. However, the solution ω_σ may not be accurately found under some parameter values. (For example, it is not possible to compute the predicted market share accurately under large parameter values such as standard deviation parameter values being 100.) Then, in the next iteration of the outer loop, it is also likely that the solution $\omega_{\sigma_{new}}$, given a new parameter value σ_{new}, is hard to compute. Therefore, the objective function may produce different values even under the same parameter values because the inaccurate solution could depend on the initial guess for the market share equation. Then, unstable objective function values may make the optimization routine move to a wrong direction and, even worse, oscillate. To avoid this problem, if the inner loop does not converge at the current outer loop iteration, we set the initial guess for the next outer loop iteration to be a predetermined value, not the inaccurate solution of the current outer loop iteration. This way, the optimization routine may get out of a bad parameter value area quickly.

REFERENCES

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

Dubé, J. P., J. T. Fox, and C. L. Su (2012): "Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation," *Econometrica*, 80(5), 2231-2267.

Houde, J. (2012): "Spatial Differentiation and Vertical Mergers in Retail Markets for Gasoline," *American Economic Review*, 102(5), 2147-2182.

Iskhakov, F., J. Lee, J. Rust, B. Schjerning, and K. Seo (2016): "Constrained Optimization Approaches to Estimation of Structural Models: Comment," *Econometrica*, 84(1), 365-370.

Lee, J. and K. Seo (2015): "A Computationally Fast Estimator for Random Coefficients Logit Demand Models Using Aggregate Data," *RAND Journal of Economics*, 46(1), 86-102.

Lee, J. and K. Seo (2016): "Revisiting the Nested Fixed-Point Algorithm in BLP Random Coefficients Demand Estimation," *Economics Letters*, 149, 67-70.

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

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

Patel, K. (2012): "How to Estimate Random Coefficients Demand Models," Essays in Industrial Organization, Doctoral dissertation, Northwestern University.

Reynaerts, J., R. Varadhan, and J. Nash (2012): "Enhancing the Convergence Properties of the BLP (1995) Contraction Mapping," VIVES Discussion Paper 35.

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

———— (2000): "Nested Fixed Point Algorithm Documentation Manual: Version 6," Department of Economics, Yale University.