High Performance Quadrature Rules: How Numerical Integration Affects a Popular Model of Product Differentiation

Benjamin S. Skrainka*
University College London
b.skrainka@ucl.ac.uk

Kenneth L. Judd
Hoover Institution
judd@hoover.stanford.edu

January 13, 2010

Abstract

The complexity of modern economic models – especially those used in applied econometrics – has increased the use of and need for efficient, robust, and accurate tools for multi-dimensional integration. First, we explain how to use modern integration techniques such as Gaussian product and monomial rules as well as the correct implementation of simple Monte Carlo methods, the de facto standard of most economists despite the superiority of monomial rules. Next, we consider how these different integration rules affect the results from the industry-standard 'BLP' model for product differentiation (Berry, Levinsohn, and Pakes, 1995). We show that simulation often leads to inaccurate market share calculations, that these errors affect the convergence of Berry's mapping – which is used to invert the market share equations for the unobserved product-market shock – and that all these problems result in multiple local maxima in the GMM objective function as well as erroneous parameter estimates. After analyzing the problems with the status quo, we provide several quadrature rules which, thanks to the power of modern computers and numerical analysis, are both more accurate and less expensive computationally. Finally, we develop custom quadrature rules which exploit the boundedness of the multinomial logit and are much more efficient for computing market share integrals in random coefficients (or mixed logit) models. These integration tools are quite general and applicable to a wide variety of economic problems.

Keywords: Numerical Integration, Monomial Rules, Gauss-Hermite Quadrature, Monte Carlo Integration, Product Differentiation, Econometrics.

*This paper would not have been possible without the help of Che-Lin Su and JP Dube for sharing their knowledge of BLP and their MATLAB code. The authors were generously supported and hosted by Ian Foster and the Computation Institute at The University of Chicago and Argonne National Laboratory during Fall 2009.

1 Introduction

Unobserved heterogeneity, incomplete information, and uncertainty are features of many modern economic models and have improved our ability to understand the complexity and variety of the real world. These features often require the computation of an integral over some – often multi-dimensional – probability distribution with some conveniet distribution in order to calculate economically relevant terms such as expected profits/utility or a likelihood. Finding the correct solution depends on computing these integrals quickly and accurately. Nevertheless, the current status quo in Economics is to use Monte Carlo methods which, although easy to program and understand, are inaccurate or even biased if not implemented correctly – i.e., when researchers do not take enough samples to approach the region where the asymptotic approximation is valid. In this paper, we explain how to obtain much more accurate numerical approximations for multidimensional integrals using monomial or product rules. To illustrate the stakes of using Monte Carlo methods incorrectly, we analyze how simulation introduces error in the point estimates in Berry, Levinsohn, and Pakes (1995)'s 'industry standard' model of product differentiation (BLP hereafter), leads to multiple local minima in the GMM objective function, and affects the convergence of Berry's mapping, which is used to invert market shares to obtain ξ_{jt}, the unobserved product-market heterogeneity (Berry, 1994). Our goal, then, is to develop better methods of multidimensional integration for economic problems and explain how to apply these tools to a wide range of problems of moderate size – i.e. $10 - 15$ dimensions. The benefits are obvious: more accurate computation at a lower computational cost.

Outline:

1. Objectives of paper
    - How to use simulation
    - How to use quadrature
    - When to use simulation or quadrature
    - How BLP can be improved with better integration methods:
        - Market shares
        - Berry's mapping
        - Multiple local optima
        - Error in Point estimates
2. Overview of integration (Cools, 2002)
3. Overview of relevant literature:
    (a) BLP Demand estimation
    (b) Numerical Integration
        i. Number-theoretic methods: Simulation

ii. Polynomial based methods: Quadrature

Integration and distributional assumptions – the focus of this paper – are just some of the numerical challenges a researcher must overcome to successfully apply BLP to a real-world problem. We mention in passing that choosing the a modern solver such as SNOPT or KNITRO, configuring the right optimization options for your problem, and correctly specifying the optimization program – especially sparseness – are also crucial numerical decisions. See Su and Judd (2008) and Dubé, Fox, and Su (2009) for a thorough discussion of the advantages of using a modern solver (i.e. not fmincon/fminunc) and formulating the optimization problem as a Mathematical Program with Equilibrium Constraints (MPECC).[1]

We begin the paper with a discussion the current state of the art for numerical integration, explaining the strengths and weaknesses of the two methods (number theoretic and polynomial-based) for computing multidimensional integrals. Next, we analyze how Monte Carlo methods provide inferior results to monomial and product rules in BLP: we start by quickly reviewing the BLP model of product differentiation and then show how simulation error ripples through the model, affecting market share calculations, convergence of Berry's mapping, and the accuracy of the point estimates. We conclude by developing custom quadrature rules which exploit the extra structure of the multinomial distribution – boundedness and slope – in the random-coefficients integral which produce even more accurate results at lower computational cost than traditional monomial rules.

2 Basic Multi-Dimensional Numerical Integration

For more almost four decades – and many rules have been known longer than that – there have been well understood rules to compute multidimensional integrals on a variety of domains accurately and efficiently (Stroud, 1971). Broadly speaking, all methods approximate an integral as a weighted sum of the integrand evaluated at a finite set of well-specified points. The art of numerical integration lies in choosing these nodes and weights so that the approximation is inexpensive to compute and has little error. Many of these rules give an exact result for all polynomials or monomials[2][[[[]]] below a certain degree. Because monomials and polynomials span the vector space of 'well-behaved' functions, any economic function which is smooth and differentiable should be easy to integrate numerically. The quality of the approximation will also depend on the

[1] Dubé, Fox, and Su (2009) also explain how sloppy loop tolerances prevent convergence of the nested fixed-point (NFX) algorithm of Rust (1987) to reliable parameter estimates.

[2] Let $\mathbb{P}^d$ be the vector space of all polynomials in d variables. Then this space is spanned by all *monomials* $x^{\mathbf{k}} \equiv \prod_{i=1}^{d} x_i^{k_i}$ where $\mathbf{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d$. I.e., a monomial is a product of x_i's raised to different powers. The *degree* of a monomial, $\deg\left(x^{\mathbf{k}}\right) \equiv \sum_i k_i$.

properites of the integrand such as smoothness, symmetry, differentiability, and thickness of the tails.

To be more explicit, consider the integral of a function $f(x)$:

$$I[f] := \int_{\Omega} w(x) f(x) dx,\ \Omega \subset \mathbb{R}^d,\ w(x) \geq 0 \, \forall x \in \Omega$$

where I have used Cools (2002)'s notation. $w(x)$ is the weight function such as 1, $\exp(-x)$, or $\exp\left(-x^2\right)$ depending on the problem. Similarly, the region of integration, Ω, is also problem dependent. For example, in a random-coefficients model $\Omega = \mathbb{R}^d$ and $w(x) = \exp\left(-x^2\right)$ after a suitable change of variables . [3]To minimize the error in computing an integral, you must choose a good approximation for $I[f]$. Following Cools, call this approximation

$$Q[f] := \sum_{j=1}^{N} w_j f(y_j),\ y_j \in \Omega,$$

where $\{w_j\}$ and $\{y_j\}$ are the quadrature nodes and weights, respectively. For example, a simple Monte Carlo rule would set $w_j = 1/N, \forall j$ and draw y_j from a suitable distribution such as $w(x)$. An ideal solution is exact, i.e. $I[f] = Q[f]$ so the approximation has no error. More likely, the approximation will not be exact. A good approximation – as well as minimizing error and the number of (expensive) function evaluations – should converge to the true value of the integral as the number of nodes goes to infinity Stroud (1971). This is increasingly important in higher dimensions.

There are two primary methods for choosing the quadrature nodes and weights to calculate integrals numerically: number theoretic methods and polynomial-based methods Cools (2002). The former refers to (quasi-) Monte Carlo (or simulation) methods whereas the later includes product rules based on the Gaussian quadrature family of methods as well as monomial rules.[4]

2.1 One Dimensional Integration: A Simple Example

To illustrate these issues, consider a simple one dimensional random coefficients multinomial logit model. An agent i chooses the alternative $j \in J$ which yields the highest utility $U_{ij} = \alpha_i (\log y_i - \log p_j) + z_j^T \beta + \epsilon_{ij}$ where ϵ_{ij} follows a Type 1 Extreme Value distribution and $\alpha_i \sim N\left(\alpha, \sigma^2\right)$ is a one

[3]To be explicit: if you are integrating over a normal densisty $\tilde{w}(u) = (2\pi |\Omega|)^{-\frac{n}{2}} \exp\left(-\frac{1}{2} u^T \Sigma^{-1} u\right)$, the change of variables exploits the Cholesky decomposition $CC^T = 2\Sigma$ so $x = C^{-1}u$ produces the form in the text. This convenience of the form becomes clear once you have a set of quadrature nodes $\{y_j\}$ and need to transform them for a specific problem. See below.

[4]Some authors (e.g. Cools, 2002) use *quadrature* to refer to one dimensional integrals and *cubature* to refer to integrals of dimension ≥ 2. We will always use quadrature to refer to any integration rule, regardless of the dimension.

dimensional random coefficient . Then conditional market shares are $s_{ij} = \frac{\exp\left[\alpha_i \left(\log y_i - \log p_j\right) + z_j^T \beta\right]}{\sum_k \exp\left[\alpha_i \left(\log y_i - \log p_k\right) + z_k^T \beta\right]}$ so the total market share of good j must be

$$\begin{aligned} s_j &= \int_{-\infty}^{\infty} s_{ij}\left(\alpha_i\right) f\left(\alpha_i\right) d\alpha_i \\ &= \int_{-\infty}^{\infty} s_{ij}\left(\alpha_i\right) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left[\alpha_i - \alpha\right]^2\right) d\alpha_i \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} s_{ij}\left(\sqrt{2}\sigma u\right) \exp\left(-u^2\right) du \\ &\approx \frac{1}{\sqrt{\pi}} \sum_k w_k s_{ij}\left(\sqrt{2}\sigma u_k\right) \end{aligned}$$

where $\{u_k\}$ and $\{w_k\}$ are the quadrature nodes and weights for Gauss-Hermite integration and I have used a simple Cholesky transformation to convert from the economic problem to the mathematical formula. We chose to use the Gauss-Hermite rule because $\exp\left(-x^2\right)$ is the weighting function for athe variance and the bounds of integration are $\pm\infty$. An added benefit is that variance σ^2 drops out from the change of variables.

2.2 Monte Carlo Integration

Monte Carlo integration involves computing the integral by taking draws from some suitable distribution and sometimes includes extra tricks to increase accuracy and speed, such as importance sampling, Halton draws, and antithetic draws (See Train (2003) for details). The alternative is to use a quadrature rule which tries to improve accuracy and efficiency through a clever choice of nodes and weights which exploits the structure of the problem. These rules are exact for all polynomials and/or monomials below a certain degree. If the integrand is well approximated by a polynomial, then a quadrature rule should perform well. Nevertheless some researchers continue to believe that simulation is the only option for high dimensional integrals. Because accuracy only increases as $\sqrt{N}$ – so the number of nodes must be increased by a factor of 100 to for each additional digit of accuracy – a more sophisticated quadrature rule will usually outperform Monte Carlo because adding well-chosen nodes should improve the integral approximation more quickly than the same number of randomly-chosen points.

In its simplest form, simulation weights weights all nodes equally by setting the weights $\omega_j = 1/N$, where $N = |J|$, and the nodes are drawn from a suitable distribution. The weight function is set to 1 because the draws come from the corresponding distribution. Consequently, simulation is easy to understand and implement and also works with functions which are not smooth and

over irregular-shaped regions, even if it does not always produce an accurate approximation of the integral. These properties make simulation one of the most popular choices for numerical integration by economists. More sophisticated methods of taking draws such as quasi-Monte Carlo methods, importance sampling, and antithetic draws are also popular and attempt to remedy the difficiencies of simulation. But, the basic problems of simulation remain: it is dirty and can produce inaccurate results, as Berry, Levinsohn, and Pakes (1995) point out: 'On the other hand, we are concerned about the variance due to simulation error. Section 6 develops variance reduction techniques that enable us to use relatively efficient simulation techniques for our problem. Even so, we found that with a reasonable number of simulation draws the contribution of the simulation error to the variance in our estimates (V3) is not negligible.' But, many researchers still fail to employ any variance reduction techniques.

This discussion follows Train (2003). I focus on the Method of Simulated Moments (MSM), which is the basis for estimating BLP-style models – except for those who now use MPEC. The original BLP paper uses an importance sampling algorithm. In general, the MSM estimator is the $\hat{\theta}_{MSM}$ which solves

$$\check{g}_n(\theta) = \sum_n \sum_j \left(d_j - \check{P}_{nj}(\theta)\right) z_{nj} \quad = \quad 0$$

where $d_j = \{0, 1\}$ depending on whether alternative j is chosen, z_{nj} are the instruments for agent n and choice j, and $\check{P}_{nj}(\theta)$ is the empirical probability of n choosing j. Quantities with a check $\check{x}$ refer to the approximation of x calculated using simulation. E.g., $\check{P}_{nj}(\theta)$ is the approximate value for $P_{nj}(\theta)$ calculated using simulation to compute the relevant integrals.

In the case of BLP, we don't observe individuals' decisions. Instead, the econometrician observes only market shares, some product characteristics, and some market characteristics. Consequently, the GMM moment conditions become

$$\begin{aligned} g(\xi(\theta)) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\xi_{jt}(\theta)\cdot h(z_{jt}, x_{jt}) \\ &= = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}s_j^{-1}(S_t;\theta)\cdot h(z_{jt}, x_{jt}) \end{aligned}$$

where s_j are calculated market shares, S_t are a vector of observed market shares for all products j in a market t, $h(\cdot)$ is a function which gives moment conditions. Then the GMM optimization program is

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

Clearly, this is a nasty piece of work because the market shares $\check{s}(S;\theta)$ must be simulated and then inverted to obtain an estimate of $\hat{\xi}_{jt}$. Any errors in

computing integrals, estimating $\hat{\theta}$, or inverting the market shares will compound, compromising the accuracy of the value computed for $\hat{\xi}_{jt}$

Consistency of Simulated Estimates A key issue which is overlooked by many researchers is choosing a sample size which is large enough to ensure consistency. Train (Section 10.5) shows that an estimator's error consists of three parts when using simulation :

$$\check{g}_n(\theta) = A + B + C$$

A traditional finite sample error, $g_n(\theta^*) \xrightarrow{d} N\left(0, \sigma^2\right)$

B simulation bias, $\mathbb{E}_r \check{g}_n(\theta^*) - g_n(\theta^*)$, $\sqrt{N}B = \dfrac{\sqrt{N}}{R}\mathcal{Z}$ for MSL so need $R > \sqrt{N}$ as $N \to \infty$

C simulation noise, $\check{g}_n(\theta^*) - \mathbb{E}_r \check{g}_n(\theta^*)$, $\to 0$ as $NR \to \infty$ because $C \overset{a}{\sim} N(0, S/NR)$ where S is the variance of the simulation noise for one draw.

$\mathbb{E}_r$ is 'the expectation of the simulated value over the draws used in the simulation.' N is the number of observations, R the number of draws.

Thus, the key points are:

- **A** is just the traditional estimator which is asymptotically normal.
- Simulation bias is 0 if $\check{g}_n$ is an unbiased estimator of g_n. This is not the case for Maximum Simulated Likelihood because log is a non-linear transformation. Given Train's formula above, **B** will not vanish for fixed R unless $\mathcal{Z} = 0$ which is the case only for MSM. Consequently, MSM is unbiased but MSL will only be unbiased if $R \to \infty$ faster than $\sqrt{N}$, i.e. $\dfrac{\sqrt{N}}{R} \to 0$ as $N \to \infty$. If this condition is not satisfied inference is impossible for MSL because there is no limiting distribution when R doesn't rise faster than $\sqrt{N}$! Many researchers neglect this point and think that large N alone is sufficient to guarantee an unbaised estimator.
- Simulation noise vanishes as $NR \to \infty$ so **C** vanishes if R rises with N.
- Summary:
  - MSL:
    - inconsistent for fixed R
    - R increases more slowly than $\sqrt{N}$ then estimator is consistent but not asymptotically normal
    - R increases more slowly than $\sqrt{N}$ then estimator is consistent, asymptotically normal, and efficient
  - MSM:

- properties of estimator depend on properties of instruments
- instruments simulated without bias and independent of residual $\Rightarrow$ unbiased and consistent, i.e. like normal GMM
- instruments simulated with bias and are not ideal $\Rightarrow$ like MSL except not asymptotically efficient
- ideal instruments $\Rightarrow$ efficient if R rises with N; fixed R then consistent and asymptotically normal but not efficient as ML

Consequently, it is crucial to increase R at a fast enough rate to ensure consistency, unbiased estimation, and, where possible, efficiency. Another key research question is are the instruments used actually ideal and simulated without bias. Does anyone check this? E.g., for the 'Hausman instruments' which are used in many of these papers?

2.3 Multi-dimensional Quadrature

We compare simulation to two multi-dimensional quadratures rules: Gaussian-Hermite product rules and a multi-dimensional monomial rule. Both rules work by using clever choices of nodes and weights which will exactly integrate all polynomials or monomials less than some chosen degree. The higher the degree, the more accurate the approximation of the integral at the cost of evaluating the function at more nodes. The actual choice of nodes and weights depends on the weighting function in the integral. For smooth functions which are well approximated by polynomials, a good quadrature rule should outperform simulation.

Gauss-Hermite Because the random coefficients are assumed to be normally distributed, the correct weight function to use is $w(x) = e^{-x^2}$. Obviously, a Gauss-Hermite rule is best for this kernel. We consider product rules for five dimensions, the dimensionality of the shock in JP Dubé's BLP code, with 3, 4, 5, 7, and 9 nodes in each dimension. Consequently, we have to evaluate the function at N^5 points, which quickly becomes much larger than $10,000$, a common upper limit on the number of simulated draws.

Monomial Rules For higher dimension, one can further cutdown on the number of points by using a monomial rule (Stroud, 1971). We use Stroud's monomial Rule 11-1 which is accurate for all 11 degree polynomials in five dimensions using only 983 nodes. This is a significant improvement on Gauss-Hermite product rules. For the BLP integral, the computed market shares, s_{jt}, were essentially identical.

XXX Add tables and formulas for Gauss-Hermite and Stroud monomial rules

3 The Basics of BLP

The BLP model has become one of the most popular empirical models of product differentiation because fits empircal data well, using a flexible form with both random coeffients and unobserved product-market characteristics, and also overcomes many problems with discrete choice models such as unrealistic substitution patterns and independence of irrelevant alternatives (IIA). Nevo (2000) provides a detailed and accessible explanation of the model. Here, I summarize the relevant features before examining the numerical issues.

BLP is a special case of the mixed logit which uses a random utility model with indirect utility

$$U_{ijt} = V_{ijt} + \epsilon_{ijt}$$

and

$$V_{ijt} = \alpha_i (y_i - p_{jt}) + x'_{jt}\beta_i + \xi_{jt}$$

where i is the agent, $j \in J$ the product, and $t \in T$ the market. ξ_{jt} is the unobserved to the econometrician product-market shock – i.e. unobserved product characteristics. ϵ_{ijt} is an IID, Type I Extreme value shock. Note: if α_i is not a function of y_i (through the D_i term – see below) then there are no income effects because the $\alpha_i y_i$ term cancels when the indirect utilities for each alternative are compared. In practice, y_i and p_{jt} are often the logarithm of the respective quantities. This ensures that the utility is homogeneous of degree zero.

For tractability, researchers assume that the coefficients $\theta_i = (\alpha_i, \beta_i)$ have a normal distribution. Often demographics are added to the equations for θ:

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} = \begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix} + \Pi D_i + \Sigma \nu_i$$

where $\bar{\alpha}$ and $\bar{\beta}$ are the mean value of the coefficients for all agents. Demographics $D_i \sim P_D$ and $\nu_i \sim P_\nu$. These distributions are parametric. Π determines how demographics affect tastes. Consequently, the market shares are the expectation of the regular MNL choice probabilities with respect to θ.

We also need to model the outside option (option $j = 0$ by convention). Because it is not observed, we typically use

$$V_{i0t} = \alpha_i y_i + \epsilon_{i0t}$$

for the indirect utility and ϵ_{i0t} is Type I Extreme Value.

Nevo and BLP then group the coeffients into a mean utility

$$\delta_{jt} = x_{jt}\bar{\beta} - \bar{\alpha} p_{jt} + \xi_{jt}$$

and an individual specific preference shock

$$\mu_{ijt} \quad = \quad [-p_{jt} \; x_{jt}] \left(\Pi D_i + \Sigma \nu_i\right)$$

where $\mu_{ijt} + \epsilon_{ijt}$ is the mean zero, composite shock which captures individual heterogeneity.

There are a couple things to note about this specification:

- $\alpha_i < 0$ implies a positive price coefficient, violating consumer theory. This problem occurs for a non-zero measure of the population. One appeal of simulation is that you can find a draw which doesn't have any positive price coefficients. Nevo has a footnote about discarding these draws. A better option is to assume that α_i is log-normally distributed. I.e., here is a case of 'sample fishing.'

- The Type I Extreme value distributional assumption + IID provides the closed form, multinomial logit solution for the conditional market shares, i.e. the probability of an agent choosing an option, conditional on the agent's type. A more general distribution would appear to require the computation of the integral for the conditional choice probability, aka market share:

$$s_{ijt} \quad = \quad \int_{-\infty}^{\infty} \left\{ \prod_{k \neq j} \int_{-\infty}^{\epsilon_{ijt} + V_{ijt}(\theta) - V_{ikt}(\theta)} f\left(\epsilon_{ikt}\right) d\epsilon_{ikt} \right\} f\left(\epsilon_{ijt}\right) d\epsilon_{ijt}$$

  where we have still assumed IID. However, McFadden and Train (2000) show that any preferences can be modeled using a suitable mixing distribution with the multinomial logit. Consequently, there is no loss in generality from using the multinomial logit. The common specification of a normal mixing distribution could cause problems from misspecification. An area for future research is to study the impact of a more flexible mixing distribution. Then the market share is

$$s_{jt} \quad = \quad \int_{\Theta} \left[\int_{-\infty}^{\infty} \left\{ \prod_{k \neq j} \int_{-\infty}^{\epsilon_{ijt} + V_{ijt}(\theta) - V_{ikt}(\theta)} f\left(\epsilon_{ikt}\right) d\epsilon_{ikt} \right\} f\left(\epsilon_{ijt}\right) d\epsilon_{ijt} \right] g\left(\theta\right) d\theta$$

  after integrating over all possible preference types which are distributed as $\theta \sim g\left(\theta\right)$.

- Without IID, we would need to replace the product of integrals over the appropriate joint distribution conditional on ϵ_{ijt}. The unconditional mar-

ket share with IID is

$$
\begin{aligned}
s_{jt} &= \int_{\Theta} s_{ijt}(\theta_i)\, g(\theta_i)\, d\theta_i \\
&= \int_{\Theta} \left[\int_{-\infty}^{\infty} \left\{ \prod_{j \neq i} \int_{-\infty}^{\epsilon_{ijt}+V_{ijt}(\theta_i)-V_{ikt}(\theta_i)} f(\epsilon_{-ikt} \,|\epsilon_{ijt}\,)\, d\epsilon_{ikt} \right\} f(\epsilon_{ijt})\, d\epsilon_{ijt} \right] g(\theta_i)\; d\theta_i .
\end{aligned}
$$

- The integrals are computed by simulation:
  - Nevo draws demographics D_i for a limited number of households from the CPS.
  - Nevo draws ν_i from a Normal
  - A point of concern is that if researchers too few draws, R, for asymptotic consistency (See below).
- The model is estimated using a nested GMM algorithm:
  1. Given parameters $\hat{\theta}^n = (\alpha, \beta, \Pi, \Sigma)$, obtain ξ_{jt} by
     (a) Solve for δ_{jt} from market shares using a contraction mapping Berry, Levinsohn, and Pakes (1995)
     (b) Invert δ_{jt} to obtain the latest estimate for the unobserved product heterogeneity, $\hat{\xi}^n_{jt}$
  2. Use $\hat{\xi}^n_{jt}$ to form GMM moment conditions and re-estimate parameters $\hat{\theta}^n$
  3. Repeat until a stopping criterion is satisfied.

4 Simulation vs. Quadrature Rule

To compare Monte Carlo integration with quadrature and monomial rules, I use these different numerical techniques to compute key quantities in the BLP model such as market share integrals, the unobserved heterogeniety ξ_{jt}, the information matrix, the standard errors, and the GMM objective function. In addition, I compare how the different methods affect the parameter estimates (optima) produced by a state of the art solver (KNITRO).

4.1 Computation of Market Shares

To compare the accuracy of different integration approaches, I compute the market share integrals for $T = 50$ markets and $J = 25$ products for the BLP model. I use the code from Dubé, Fox, and Su (2009)[5] to generate a dataset

[5] The code was downloaded from `http://faculty.chicagobooth.edu/jean-pierre.dube/research/`, Fall 2009.

Table 1: Comparison of Integration Rules

Rule	**Type**	N_{nodes}	**Max Abs Error**	**Min Abs Error**
Simulation	Simple Random Draw	100	7.551948439036227e-02	1.527297314339364e-12
		1,000	4.708060066243702e-02	3.335032180409949e-11
		10,000	3.896166560785574e-02	1.182201694457840e-10
Gauss-Hermite	Product Rule	$3^5 = 243$	1.072985611572319e-03	4.533583487614291e-12
		$4^5 = 1,024$	2.517427160893537e-04	1.758592200356603e-12
		$5^5 = 3,125$	5.644382503433576e-05	1.311538115999068e-13
		$7^5 = 16,807$	0	0
		$9^5 = 59,049$	–	–
Stroud Rule 11-1	Left Column	983	8.121581065728689e-05	6.107747097034455e-15
	Right Column	983	1.379071789571751e-04	1.551264895804618e-14

All values are computed at $\hat{\theta}_{MPEC}$ based on $R = 100$ draws. The 9^5 product rule cannot be evaluated because MATLAB runs out of memory to calculate all $T \times J$ market shares. Errors are relative to the 7^5 Gauss-Hermite product rule. Values for simulation are max/min for 5 trials.

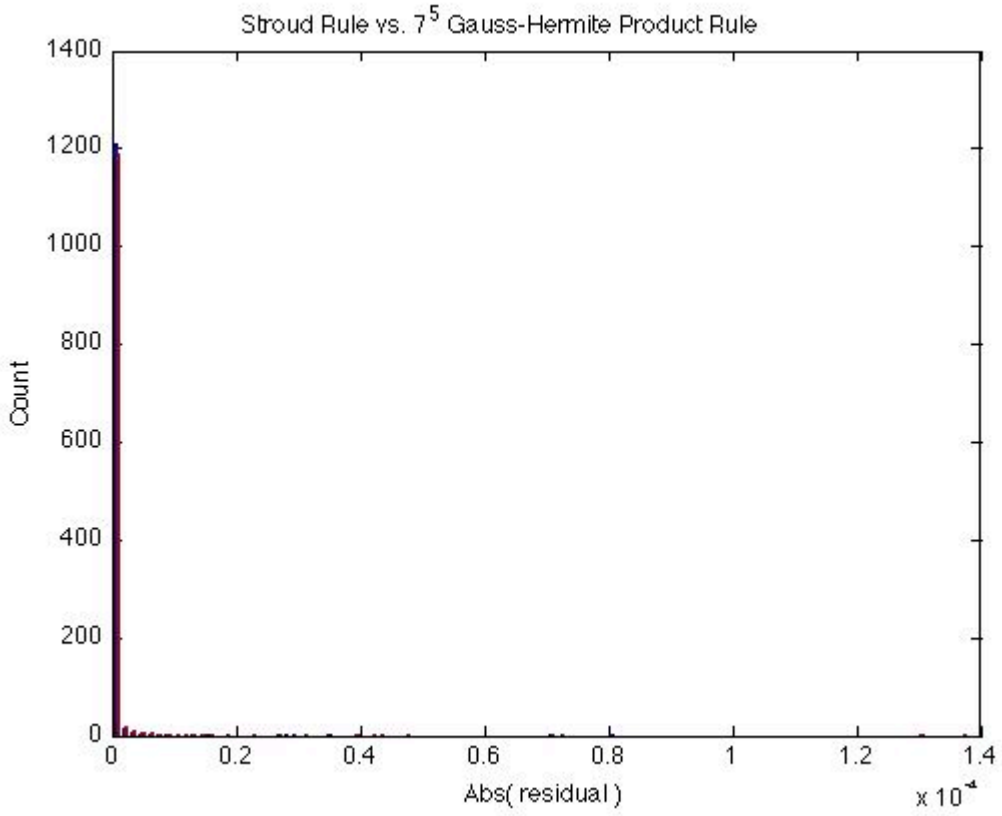
using $R = 100$ draws from a Normal distribution. I then compute the market shares, s_{jt}, for this data using simulation, Gauss-Hermite product rules, and Stroud's monomial rule 11-1. For all cases, I compute the market shares at the point estimate, $\hat{\theta}_{MPEC}$, generated by the MPEC estimation code provided in Dubé, Fox, and Su (2009). These results are tabulated below in Table 4.1.

The striking thing when looking at a histogram of the Stroud vs. Gauss-Hermite product rule residuals, is that the Stroud rule is only off at a few (˜10) points. See figure 1. The same histogram for the simulated market shares shows a similar distribution, though the tail is a little fatter and the order of magnitude is a factor of 100 worse for 10^4 draws! Also note that even increasing the number of simulation draws from 100 to $10,000$ does little to improve the accuracy of the integral. In fact, it is less than the expected factor of $\sqrt{N}$ ($\sqrt{100} = 10$). However a crucial factor driving this result is that for any market, most of the products have very small market share. Consequently, only a few products determine the parameter values and estimating these market shares correctly, then, is crucial. The larger market shares are also those with larger variances across multiple simulation runs: the market shares which have small variances do so because the market shares are essentially zero. Furthermore, this effect becomes starker with more simulation draws. See table 2. Another issue is that the parameter value used to compute the shares will affect which combinations of product and market produce the largest shares. Simple tests show that 10% or more of shares could move into or out of the top decile. Consequently, simulation will also make shifts in parameters more knife-edged at the extensive margin, further complicating the task of the solver.

Examination of a plot of s_i vs. $\text{var}[\hat{s}_i]$ where $\hat{s}_i$ is the simulated version of s_i shows:

- $\hat{s}_i$ is consistently smaller or larger than the value of s_i computed via Stroud

Figure 1: Stroud Rule 11-1 vs. Gauss-Hermite Product Rule



rule 11-1.

- larger shares tend to have larger variances
- Most shares and variances are extremely close to 0. When I look at a histogram of market shares, 88.3% of shares are less than 1% and 93.3% are less than 5%.

XXX Use a Log scale here.
XXX Only need a couple digits here....

In the course of computing these market shares, I noticed that the manner of computing the market share integral affects the point estimates as well as the starting value. Running with the default configuration provided in the code, there are five random starts of which three agree, one doesn't converge in 100 iterations (verify), and the fifth is significantly better than the other four – i.e., it has a much lower value for the GMM objective function. When I compare these estimates to those obtained with either simulation with more draws (larger

Table 2: Impact of Market Share on Variance

	$R = 100$	$R = 1000$	$R = 10,000$
min Var $[s_i < 0.0095]$	1.030823534978537e-20		4.516622606490837e-10
max Var $[s_i < 0.0095]$	9.227279021096175e-06		9.477527876912232e-03
min Var $[s_i > 0.0095]$	7.043915397070283e-07		9.799958561130986e-03
max Var $[s_i > 0.0095]$	1.114497255204688e-03		9.351920115545796e-01

For 20 simulations with R draws.

R) or a Gauss-Hermite product, the point estimates usually have statistically significant differences for most components of $\hat{\theta}$. To determine significance, I use the standard errors calculated from the GMM variance for θ based on the familiar sandwich product estimator.

- **XXX how many are deviations are statistically significant? I.e., how many deviations are important?**
- **Check at solution, using high-quality quadrature, find Jacobian of equation:**
  1. Is the system well conditioned? i.e. condition number is less than 10^8.
  2. What is conditioning at parameter estimates?
     - Good quadrature rule (i.e. monomial rule)
     - Monte Carlo: use multiple draws

Negative Market Shares Stroud Rule 11-1 Right column produces several negative market shares which are basically zero:

-5.490794199853441e-09
-8.128211836318200e-10
-2.548912005938970e-12
-5.235446754952933e-10
-5.741560408205868e-12
-2.082523606903214e-10

Figure 2 clearly shows the simulation error in the computation of market shares s_{jt}. The green points represent market shares which were calculated via simulation whereas the red and blue triangles used Stroud rule 11-1. As a check, I plot the 9^5 Gauss-Hermite product rule, shown as a magenta pentagon. The majority of simulated share values are either considerably below or above those for Stroud's monomial rule 11-1 and the Gauss-Hermite product rule with 9^5 nodes, both of which produce the same market share values to many decimal places. The propagation of error from the Gaussian draws to simulate the integral cause this problem because errors in the draws for ν affect the values for the random coefficients which, in turn, affect the agent-specific utility, μ_{ijt}. Errors in computing μ_{ijt} are further distorted by the multinomial logit

transformation which can be flat, concave, or convex depending on parameter values. From Jensen's inequality we know that the expectation of a concave (convex) function is more (less) than the function of the expectation. Consequently, simulation error percolates through the multinomial logit form for the conditional shares to produce either positive or negative error. Two facts support this: (1) the mean of μ_{ijt} is large for simulation – $|\text{mean}(\mu_{ijt})| \geq 10^{-3}$ – compared to the mean with the monomial rule, $|\text{mean}(\mu_{ijt})| \leq 10^{-17}$, even with $N_{Draws} = 10,000$ draws for the Monte Carlo integral over the random coefficients (i.e., ν); and (2) the correlation coefficient of μ_{ijt} and the simulation error, $e_{jt} = s_{jt}^{MC} - s_{jt}^{Stroud}$, is about -0.2 conditional on $|e_{jt}| > 10^{-4}$ (This result, of course, depends on the draws.).[6]

In addition, simulation error explains why the solver and Berry's mapping converge more frequently – even if not always to the same value – for Monte Carlo methods than monomial or product rules (See below for further discussion of these issues.). The optimizer adjusts parameters so that the spectrum of the mapping is less singular and has local basins of attraction. The different sizes of these basins affect how often solver finds them when searching for a local minimum of the GMM objective function. I also found that SNOPT 7 could often find an optimum when KNITRO would not converge. That SNOPT 7 was recently upgraded to handle rank deficient systems further supports to these claims.

4.2 Computation of Product-Market Shock ξ_{jt}

To compare how integration methods affect the computed value of the ξ_{jt}, I computed the value of the shock using simulation, a product rule, and a monomial rule. My method was the following: take the parameter estimates from the NFX algorithm for a variety of starting values, compute the market shares s_{jt} with each integration rule, and then invert the market shares to recover the value of ξ_{jt} using BLP's contraction mapping with a tight tolerance of 10^{-14} (This tolerance is what Dubé, Fox, and Su (2009) call the 'inner loop tolerance') to ensure accurate results. Two further issues of concern about the contraction mapping are that the approximate mapping operator is not a contraction [7] and that numerical errors in the computation of the market shares could lead to non-convergent cycles.

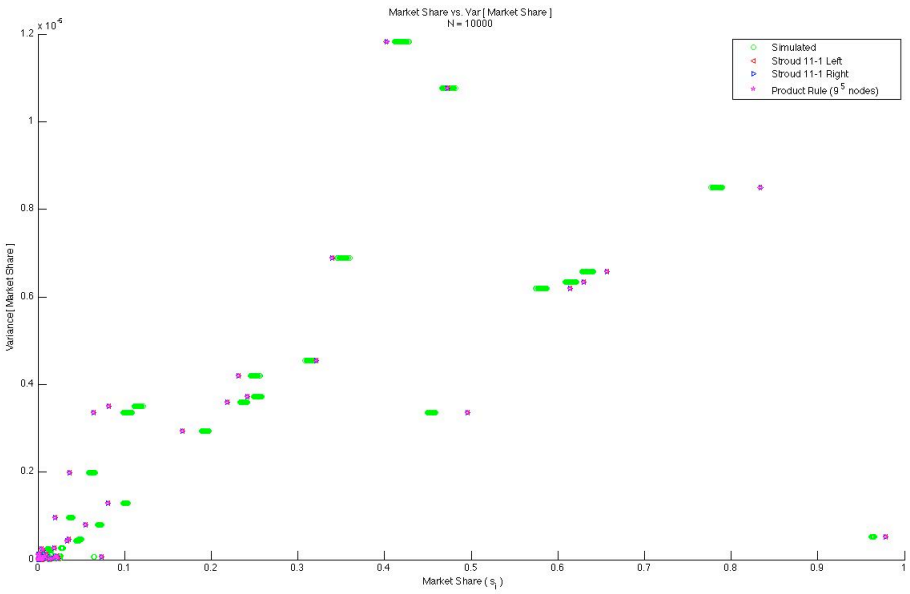
XXX Verify behavior of contraction mapping for various Monte Carlo draws

XXX Verify eigenvalues of contraction mapping – CSD – at the solution? How slow is contraction? Does sample fishing give you stability when the contraction isn't.

[6] Here, $\text{mean}(\mu_{ijt}) \equiv \frac{1}{N_{Draws}} \sum_i \mu_{ijt}$.

[7] Gandhi (2008) does show that the true mapping is invertible under some axioms. However, invertibility does not imply contraction.

Figure 2: Simulation Error: s_{jt} vs. var $[\hat{s}_{jt}]$



XXX How dows simulation affect Jacobian for ξ_{jt} or δ_{jt} in contraction mapping − use $N = 1,000$ ... do a full range.... stability implies that all eigenvalues are

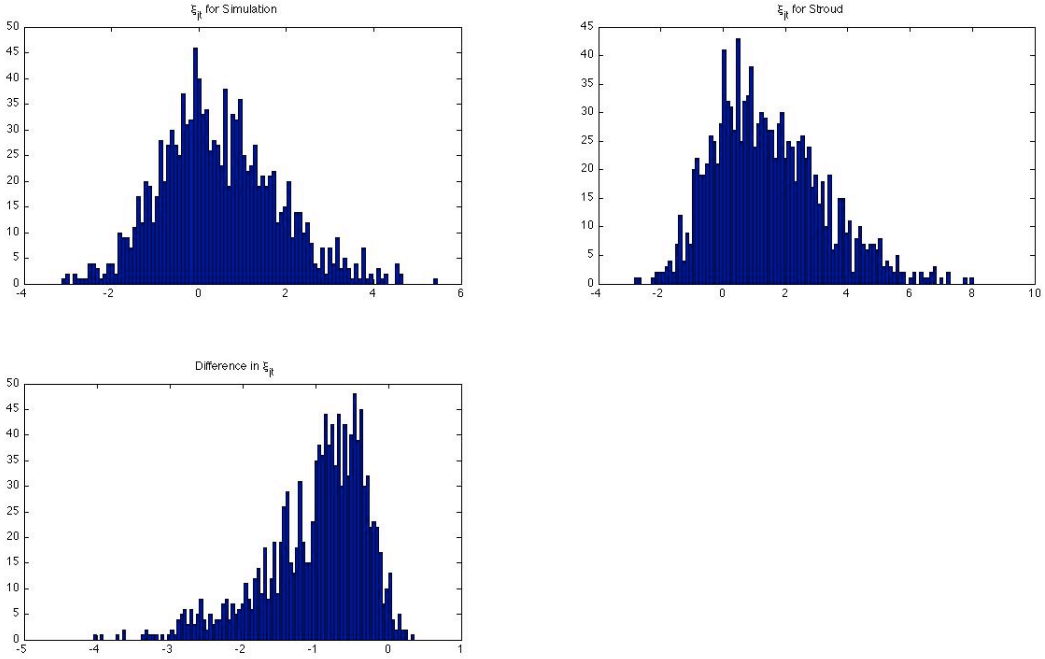
This experiment showed that simulation produces a very different result for the product-market shock from the product and monomial rules. Examining the histograms of ξ_{jt}, the mean and variance are much larger under simulation. In addition, there is more skewness. See Figure 3.

4.3 Impact of Simulation on Optimization

An important open question is to quantify how the integration method affects the solver's ability to find a (global) maximum, i.e. to find the correct parameter estimates. Simulation is dirty which means that as the parameter moves through parameter space, there are discrete jumps at the extensive margin whereas with a quadrature rule the distribution of consumers is smooth. Consequently the simulated objective function is lumpy like a step function unlike the quadrature rule objective function which is smooth. Furthermore, as mentioned above, the lumpiness of simulation means which are the large market-shares may shift in a knife-edged fashion as parameter values changed, causing local optima and increasing the difficulty of finding a global maximum.

Dubé, Fox, and Su (2009) side-step the issue of convergence of the solver to a some extent by using the same draws which generated the data to compute

Figure 3: Product Market Shocks



the market shares via simulation when estimating their BLP model. Clearly, in a real world problem these shocks would not be observed by the econometrician. [8] When I redraw the these shocks, some of the starting values fail to converge. A quadrature rule mitigates these issues. **Need more data on this XXX!**

With Seed A, 4/5 starts converge and three of these get the same answer. The other is higher. With Seed B, instead of Seed A, 3/5 starts converge. 2 have save value for GMM objective. The other is higher. Note: Seed B uses 5,000 draws to calculate integrals vs. 1,000 for Seed A and T = 20, prods = 15 vs. T = 50, prods = 25.

4.4 Problems with Berry's Mapping

In theory, Berry's mapping is a contraction Berry (1994), but in practice, when integrals must be approximated, it is unclear whether or not the mapping is still a contraction. My experiments show that more accurate approximation of the integrals with monomial rules causes the mapping not to converge whereas it usually converges for a Monte Carlo rule. This does not mean that monomial rules are bad, rather that the messyness of simulation facilities convergence.

Decreasing the number of markets and products:

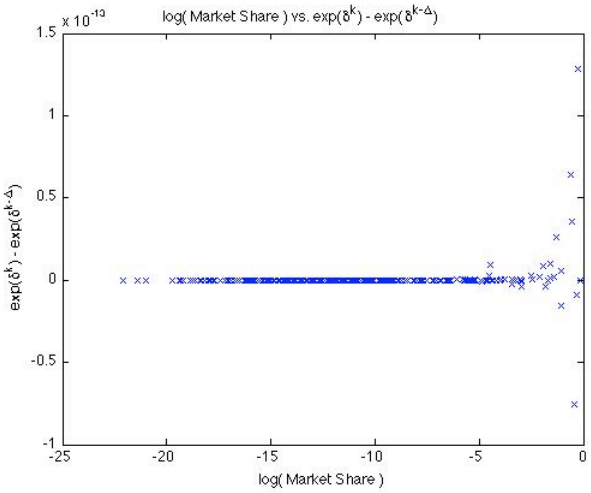
- Trouble with convergence of contraction mapping with Stroud Monomial rules
- Jacobian has huge condition number: 19 (full matrix) or 16 (sparse). Computing eigenvalue for sparse matrix produces a warning that the matrix is close to singular or badly scaled
- T = 20, prods = 15
- Decreasing $T * prods$ makes contraction mapping less stable, i.e. closer to singular
- Convergence problems with MPEC or Berry's mapping under monomial rule are not because the monomial rule is bad but because under the more accurate integration approximation the mapping is closer to singular
- The spectrum of eigenvalues for the mapping:
  - histogram of $\log_{10}(\|\lambda\|)$ shows that most eigenvalues are very small; only a few are close to 1; λ_{max} determines rate of convergence and is ~ 0.95
  - Plot quintiles/quartiles
  - How affected by number of simulation draws? Should get worse with more draws....
  - 6 of 100 mappings fail to converge with $T = 20$, $prods = 15$, and $N_{Draws} = 1,000$ draws for Monte Carlo integration.

Table 3: Statistics on Iterations Until Convergence

Statistic	Value
mean	1133.70
median	1139
variance	2909.11
N	94

Note: 6 of 100 sets of draws failed to converge using the ℓ^{∞} norm with an absolute tolerance of 10^{-14}. 20 markets, 15 products, $N_{Draws} = 1,000$ draws for Monte Carlo integration.

Figure 4: Market Share vs. Mapping Residual



The conventional wisdom is to iterate until $\left\|\exp\left[\delta^{n}\right] - \exp\left[\delta^{n-1}\right]\right\| < \epsilon_{Inner}$ where $\|\cdot\|$ is a suitable norm such as ℓ_{∞}, ϵ_{Inner} is the inner loop stopping criterion, and δ is the mean utility (and a function of ξ_{jt}). A plot of market share vs. this residual shows that convergence problems come from the largest market shares (Figure 4). In addition, heuristic tests to compute β, the rate of contraction for the mapping show that β is always close to 1 and often exceeds it. See Judd (1998) for a discussion of approximating β as $\left[\left(\left\|\exp\left(\delta^{k+2\Delta}\right) - \exp\left(\delta^{k+\Delta}\right)\right\| / \left\|\exp\left(\delta^{k+\Delta}\right) - \exp\left(\delta^{k}\right)\right\|\right)\right]^{1/\Delta}$ to examine the convergence properties of a mapping. The exponent on δ is the number of interation and $\Delta > 0$ is some additional number of steps so that the contraction property of the mapping is compared over a larger range.

[8] When we discussed the issue with J.P Dubé, he argued that he could treat the shocks ξ_{jt} as 'data'.

4.5 Negative Price Coefficients

Compute what fraction of price coefficients are negative. How does this affect things? Rerun with a lognormal distribution.

5 Custom Quadrature Rules

Traditional Gauss product rules and monomial rules are designed for general integration of functions for which the Riemann–Stieltjes integral exists. These rules fail to exploit all of the structure in the random-coefficients integral because the integral consists of two parts: the conditional market share for a given type – such as a multinomial distribution – and the distribution of the coefficients, usually assumed to be normal. Consequently a custom rule which exploits slope information or that the conditional market share must be bounded between 0 and 1 should be even more efficient and accurate than other quadrature rules.

Two approaches are possible to exploit this structure: (1) Gauss-Turan quadrature, which uses slope information, and (2) custom monomial rules.

XXX Insert Custom Monomial Rules.

6 Conclusion

With modern computing resources, there is no excuse not to use a proper quadrature rule for integration because it is both faster and more accurate.

---

7 Notes on a Degenerate Case

Consider the most simple model:

$$u_{ijt} = \xi_{jt} + \epsilon_{ijt}$$

so market shares are

$$s_{jt} = \frac{\exp\left(\xi_{jt}\right)}{\sum_k \exp\left(\xi_{kt}\right)}$$

and the Berry map is

$$\xi_i^{t+1} = S_i^{obs} + \log\left(\sum_k \exp\left(\xi_k^t\right)\right).$$

This formula depends on having no random coefficients so that the necessary terms cancel out.

If there are three goods and each has the same share of $1/3$ then the Jacobian of the above map

$$\nabla T_i \propto (1, 1, 1)$$

which means the Jacobian is singular with $n-1$ eigenvalues equal to 0. However, if you rewrite this as a Fixed Point equation/ a set of nonlinear equations is not singular because the Jacobian is $\nabla T_i - I$

Two degenerate cases:

- Infinite amount of data. What happens to Berry map for logit case as data becomes more random
- Random coefficient. Infinite amount of data. Know true parameters. Use infinitely accurate quadrature rule. As sample size $\downarrow 0$, $\uparrow$ sample error.

One implication is that if you have many market shares which are nearly the same (i.e. small) , then the Jacobian is small.

Key Issues/Themes:

- MC integration creates error in integrals. Propogation of errors $\Rightarrow$ solution is very sensitive to sample size & quadrature rule. Need special rules for bounded functions.
- If MPEC is a problem with a good quadrature rule and only works with SNOPT, this a separate paper. Must explain ??? perturbation
- Berry map blows up
- Asymptotic analysis using perturbation theory
- Later, attack CCP & Hotz-Miller, Dynamic BLP, Seim

Things to to on first paper:

- Sensitivity Testing:
  - Get point estimates from NFP.
  - Perturb solution, i.e. small perturbation of everything (including ξ_{jt}). Perturb anything which is a variable/unknown.
  - Does it still satisfy inner stoping rule? Outer rule? What is the set of points which satisfies the stopping rule?
  - Is this large relative to standard errors? $\Rightarrow$ results are garbage
  - We have 'numerical equivalence' analogous to 'observational equivalence'

Task for Tibi:

- Objectives:
  - Compute integrals for many different draws (data, integrals)

  - Compute Berry's mapping for many different draws:
    * Characterize Jacobian of mapping (eigenvalues, rate of convergence)
  - Understand propagation of error from miscalculation of integrals

- Design:
  - Input:
    * Random seeds
    * Points to evaluate market share integrals
    * Points to evaluate Berry's mapping
  - Output:
    * Market shares
    * Plots
    * Statistics:
      · Convergence of Mapping (eigenvalues, their distribution, etc.)
      · Nature of error/how it propagates

References

Berry, S. (1994): "Estimating Discrete-Choice Models of Product Differentiation," *The RAND Journal of Economics*, pp. 242–262.

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

Cools, R. (2002): "Advances in Multidimensional Integration," *Journal of Computational and Applied Mathematics*, 149(1), 1–12.

Dubé, J.-P. H., J. T. Fox, and C.-L. Su (2009): "Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation," *SSRN eLibrary*.

Gandhi, A. (2008): "Inverting Demand Systems," .

Judd, K. (1998): *Numerical Methods in Economics*. The MIT Press.

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

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

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

Stroud, A. (1971): *Approximate Calculation of Multiple Integrals*. Prentice Hall.

Su, C.-L., and K. L. Judd (2008): "Constrained Optimization Approaches to Estimation of Structural Models," *SSRN eLibrary*.

Train, K. E. (2003): *Discrete Choice Methods with Simulation*. Cambridge University Press.