High Performance Quadrature Rules

How Numerical Integration Affects a Popular Model of Product Differentiation

Benjamin S. Skrainka (UCL)
Kenneth L. Judd (Hoover)

March 24, 2010

The Big Picture

Accurate and efficient numerical approximation of multi-dimensional integrals is crucial for modern economic research:

- Unobserved heterogeneity
- Uncertainty
- Incomplete information (expectations)
- Approximation of densities, functions, etc. using basis functions: e.g., sieve estimation

Need to approximate integrals quickly and accurately!

Research Objectives

Our paper shows that there are superior methods for multi-dimensional numerical integration than Monte Carlo methods:

- Polynomial-based methods are both more accurate and more efficient
- Using MC methods incorrectly compromises numerical results and masks identification problems
- MC causes many problems in Berry, Levinsohn, & Pakes (1995, 2004)-style models, including inaccurate market share values, multiple solutions, and non-convergence of the solver and Berry's mapping

Results currently based on five simulated data sets.

A Bit of Literature

Some integration literature:

- Stroud (1971)
- Genz (1993)
- Cools (1997, 2002, 2003)
- Judd (1998)
- Heiss & Winschel (2008)

Some discrete choice literature:

- Berry, Levinsohn, & Pakes (1995, 2004); Nevo (2000a, 2000b, 2001)
- McFadden & Train (2000)
- Train (2009)

Just a Bit More Literature

There is a growing, but young, literature on identification:

- Chiou & Walker (2007); Walker (2002); Walker, Ben-Akiva, & Bolduc (2004)
- Berry & Haile (2009)

Numerical literature focuses on optimization:

- Dubé, Fox, & Su (2009); Su & Judd (2009)

Roadmap

The plan for this talk is:

1. Review quadrature methods for multi-dimensional numerical integration
2. Example: mixed logit
3. BLP model
4. Results
5. Future Research
6. Conclusions

Numerical Integration

Integration Basics

Want to approximate some (multidimensional) integral

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

as

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

- The crucial issue is how to choose the nodes and weights, (w_j, y_j)
- Ideally, a rule should have $\lim_{R\to\infty} Q^R[f] = I[f]$

Example: Mixed Logit

Conditional shares with linear utility & Type I Extreme value:

$$s_{ij}(\alpha_i) = \frac{\exp\left[-\alpha_i \log p_j + z_j^T \beta\right]}{\sum_k \exp\left[-\alpha_i \log p_k + z_k^T \beta\right]}$$

Computed market shares are then:

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

Note: mixed logit $\leftrightarrow$ random coefficients

Monte Carlo Methods

There are several MC tools – all based on number theory:

- **p**seudo-**MC**:
  - Draw nodes, y_i, from weight function $w(x)$
  - Set weights to $1/R$
  - Variance reduction: importance & antithetic sampling
  - Converges as $R^{-d/2}$
  - Warning: must increase draws 100x for each extra decimal point of accuracy!
- **q**uasi-**MC**:
  - Use number theory to choose nodes which have better properties, e.g. equidistribution, low discrepancy
  - Set weights to $1/R$
  - Halton draws, Niederreiter sequences, etc.
  - Converges as $R^{-d/2}$
- For high dimensions, MC is the only option....

Warning!

Beware of 'random numbers':

- 'Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.' – von Neumann
- Computer generates *pseudo*-random numbers, which are not truly random!
- Make sure your proof works with $Q[f]$ and not just for $I[f]$!
- Different quasi-MC algorithms work better for different kinds of problems so it pays to experiment

One-dimensional Polynomial-based Methods

Domain and weight function determine which nodes are clever – based on ideal theory:

- In one dimension, use Gaussian rule:
  - With R nodes, the rule is *exact* for all polynomials of degree $2R-1$ or less!!!
  - Nodes are zeros of some basis function
  - Different weights, w_i, for each node, y_i
- Common cases:

Rule	$w(x)$	Ω
Gauss-Hermite	$\exp\left(-x^2\right)$	$\mathbb{R}$
Gauss-Chebyshev	$\left(1-x^2\right)^{-1/2}$	$[-1,1]$
Gauss-Legendre	1	$[-1,1]$
Gauss-Laguerre	$\exp(-x)$	$[0,\infty]$

- The best method for one dimension unless your function is ugly
- $\lim_{R\to\infty} Q^R[f] = I[f] \;\; \forall f \in \mathcal{C}^1$

Example: Gauss-Hermite

For five nodes we have:

y_k	w_k
2.02018287045608 6e+00	1.99532420590459 1e-02
9.58572464613818 5e-01	3.93619323152241 0e-01
0	9.45308720482941 7e-01
-9.58572464613818 5e-01	3.93619323152241 0e-01
-2.02018287045608 6e+00	1.99532420590459 1e-02

- These nodes and weights will exactly integrate any polynomial of degree $2 \times 5 - 1 = 9$ or less!
- The moral: use a Gaussian rule if your function is well approximated by a polynomial.

Multidimensional Polynomial-based Methods

Gaussian Product rules:

- Form Kronecker product of all combinations of one dimensional nodes and weights
- I.e. all points on a lattice

Sparse Grids:

- Choose nodes on lattice more symmetrically
- Easier to compute than monomial rules
- ... but uses more nodes and weights

Monomial Rules:

- Uses absolute minimum of nodes
- Look up data in a table
- Monomial: $x^{\mathbf{p}} \equiv \prod_j x^{p_j}$
  - where $\mathbf{p} = (p_1, p_2, \ldots, p_J)$
  - Degree is $\sum_j p_j$
- Exact for all monomials less than or equal to some degree

Sparse Grids

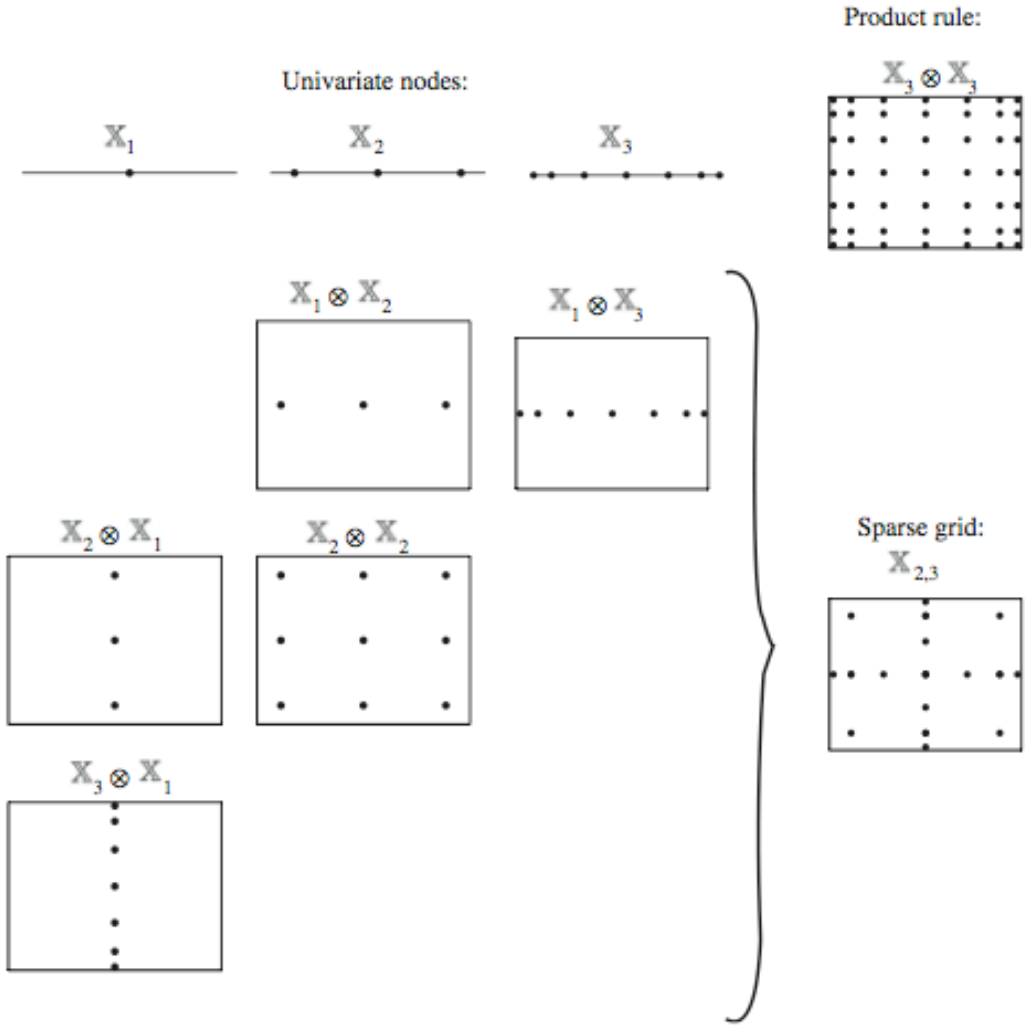


Fig. 1. Construction of the sparse grid in two dimensions.

Figure from Heiss & Winschel (2008)

Polynomial-based vs. MC Rules

The advantages of polynomial-based quadrature rules:

- Integrates any monomial below a certain degree exactly
- Integrates anything which can be approximated with monomials well
- More efficient
- More accurate

The advantages of MC rules:

- Easier to implement – but not by that much
- Only option for high dimensions
- Handle weird domains
- May be better for some irregularities (jumps, kinks)

But, neither method handles highly irregular functions – e.g. lack of smoothness, extreme curvature

The BLP Model

BLP is a mixed logit with unobserved product-market heterogeneity for studying product differentiation:

- Individuals' random coefficients capture horizontal differences in taste
- Product-market shock, ξ_{jt}, captures vertical differences in quality
- Traditional estimation strategy uses nested-fixed point (Rust, 1987):
  - Outer loop uses $\hat{\xi}_{jt}^{n}$ to compute parameter estimates, $\hat{\theta}^{n}$, using GMM
  - Inner loop uses current parameter estimates $\hat{\theta}^{n}$ to estimate $\hat{\xi}_{jt}^{n+1}$ by equating observed (S_{jt}) and calculated market share integrals (s_{jt}): $S_{jt} = s_{jt}\left(\delta\left(\xi;\theta_1\right);\theta_2\right)$
  - Market share integrals computed via pMC (inaccurate)
  - (Best practice is to use MPEC (Su & Judd, 2009; Dubé, Fox, & Su, 2000))

The Outer Loop

The outer loop estimates the parameters via GMM:

$$\hat{\theta} = \arg\max_{\theta} \left(Z'\xi\right)' W \left(Z'\xi\right)$$

- Z instruments (e.g. Hausman, Pakes, or cost shifters)
- ξ unobserved product-market heterogeneity
- W Weighting matrix

The Inner Loop

The inner loop inverts the market share equations

$$S_{jt} = s_{jt}\left(\delta\left(\xi, p, x; \theta_1\right); \theta_2\right)$$

to obtain the mean utility δ_{jt} and, thus, ξ_{jt} which is needed for the outer loop

- Usual procedure is to use Berry's mapping, aka 'The Contraction Mapping'
- Shown to converge only for the case of exact integrals (Berry, 1994)

$$\exp\left(\delta_{jt}^{n+1}\right) = \exp\left(\delta_{jt}^{n}\right) \times S_{jt} / s_{jt}\left(\delta_{jt}^{n}; \theta_2\right)$$

Remark: The mapping is written this way for performance reasons – log is more expensive than exp

Remark: For simple logit, $\delta_{jt} = \log\left(s_{jt}\right) - \log\left(s_{0t}\right)$

Specification

Utility:

$$\begin{aligned} u_{ijt} &= V_{ijt} + \epsilon_{ijt} \\ V_{ijt} &= \alpha_i \left(y_i - p_{jt} \right) + x_{jt}' \beta_i + \xi_{jt} \end{aligned}$$

Distributions:

$$\begin{aligned} \epsilon_{ijt} &\sim \text{Type I Extreme Value} \\ \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} &= \begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix} + \Sigma \nu_i \\ \nu_i &\sim \text{N}(0, 1) \end{aligned}$$

By convention, $\theta = (\theta_1, \theta_2)$ where $\theta_1 = \left(\bar{\alpha}, \bar{\beta} \right)'$ and $\theta_2 = \text{vec}\left(\Sigma \right)$

Repackaging

It is customary to repackage the utility into constant and stochastic parts:

- Mean utility: $\delta_{jt}\left(\xi_{jt};\theta_1\right) = x_{jt}\bar{\beta} - \bar{\alpha}p_{jt} + \xi_{jt}$
- Preference shock: $\mu_{ijt}\left(\nu_i;\Sigma\right) = \left[-p_{jt}\ x_{jt}\right]\left(\Sigma\nu_i\right)$

Goal of estimation is to recover δ_{jt} to obtain ξ_{jt} for GMM outer loop. Once you have estimated the mean utility, e.g. with MLE, you can even regress $\hat{\delta}_{jt}$ on the covariates to estimate $\left(\bar{\alpha},\bar{\beta}\right)'$...

Calculated Share Integrals

A central part of BLP is the computation of the market share integrals:

- Conditional market shares:

$$s_{jt}\left(\delta\left(\xi;\theta_1\right)|\theta_2\right) = \frac{\exp\left[\delta_{jt}+\mu_{ijt}\left(\nu\right)\right]}{\sum\exp\left[\delta_{kt}+\mu_{ikt}\left(\nu\right)\right]}$$

- Unconditional market shares:

$$s_{jt}\left(\delta\left(\xi;\theta_1\right);\theta_2\right) = \int_{\mathbb{R}^{K+1}} \frac{\exp\left[\delta_{jt}+\mu_{ijt}\left(\nu\right)\right]}{\sum\exp\left[\delta_{kt}+\mu_{ikt}\left(\nu\right)\right]}\phi\left(\nu\right)d\nu$$

Assumptions

The model is depends on structural assumptions:

- Specification of indirect utility
- Distributional assumptions
- Identification
- No income effects
- Static
- Purchase at most one unit of good
- Share of outside good

Results

We investigate the performance of BLP using MC data:

- Currently five simulated data sets
- Simulate typical BLP data setup:
  - Instruments: Hausman, Pakes, & supply-side cost-shifters
  - Endogenous price
  - No demographics
  - Diagonal Σ ('industry standard' assumption – facilitates identification)
  - Only $R = 100$ draws for share integrals to generate some 'real-world' noise
- Code based on Dubé, Fox, & Su (2009)
- All pseudo-random numbers created with MATLAB^{TM}'s `rand` and `randn`
- Sparse grids generated using code from `www.sparse-grids.de`

Computed Market Shares

We computed the 5-dimensional, market share integrals, s_{jt}:

- Evaluated at parameter estimates $\hat{\theta}^{MPEC}$ and at nine points drawn from $\mathrm{N}\left(\hat{\theta}^{MPEC}, 1/4 \left\|\hat{\theta}^{MPEC}\right\|\right)$
- Parameter estimates computed with MPEC and KNITRO or SNOPT
- $N = 100$ simulations for pMC and qMC
  - $R = \{100, 1000, 10000\}$ draws for pMC
  - $R = \{100, 1000, 10000\}$ with $10,000$ burn in for a Niederreiter qMC rule

Compared with polynomial-based rules:

- GH product rule with $3^5, 5^5, 7^5, 9^5$ nodes
- Sparse grids with Konrad-Patterson rule: exact for degree 11 monomials with 993 nodes.
- Stroud monomial rule 11-1: exact for all degree 11 monomials with just 983 nodes!

Share Integral Approximation

Recall the original share integral:

$$s_{jt}\left(\delta\left(\xi;\theta_1\right);\theta_2\right) = \int_{\mathbb{R}^{K+1}} \frac{\exp\left[\delta_{jt}+\mu_{ijt}\left(\nu\right)\right]}{\sum\exp\left[\delta_{kt}+\mu_{ikt}\left(\nu\right)\right]}\phi\left(\nu\right)d\nu$$

We approximate it as:

$$s_{jt}\left(\delta\left(\xi;\theta_1\right);\theta_2\right) = \sum_{m\in\mathcal{N}} w_m \frac{\exp\left[\delta_{jt}+\mu_{ijt}\left(y_m\right)\right]}{\sum\exp\left[\delta_{kt}+\mu_{ikt}\left(y_m\right)\right]}$$

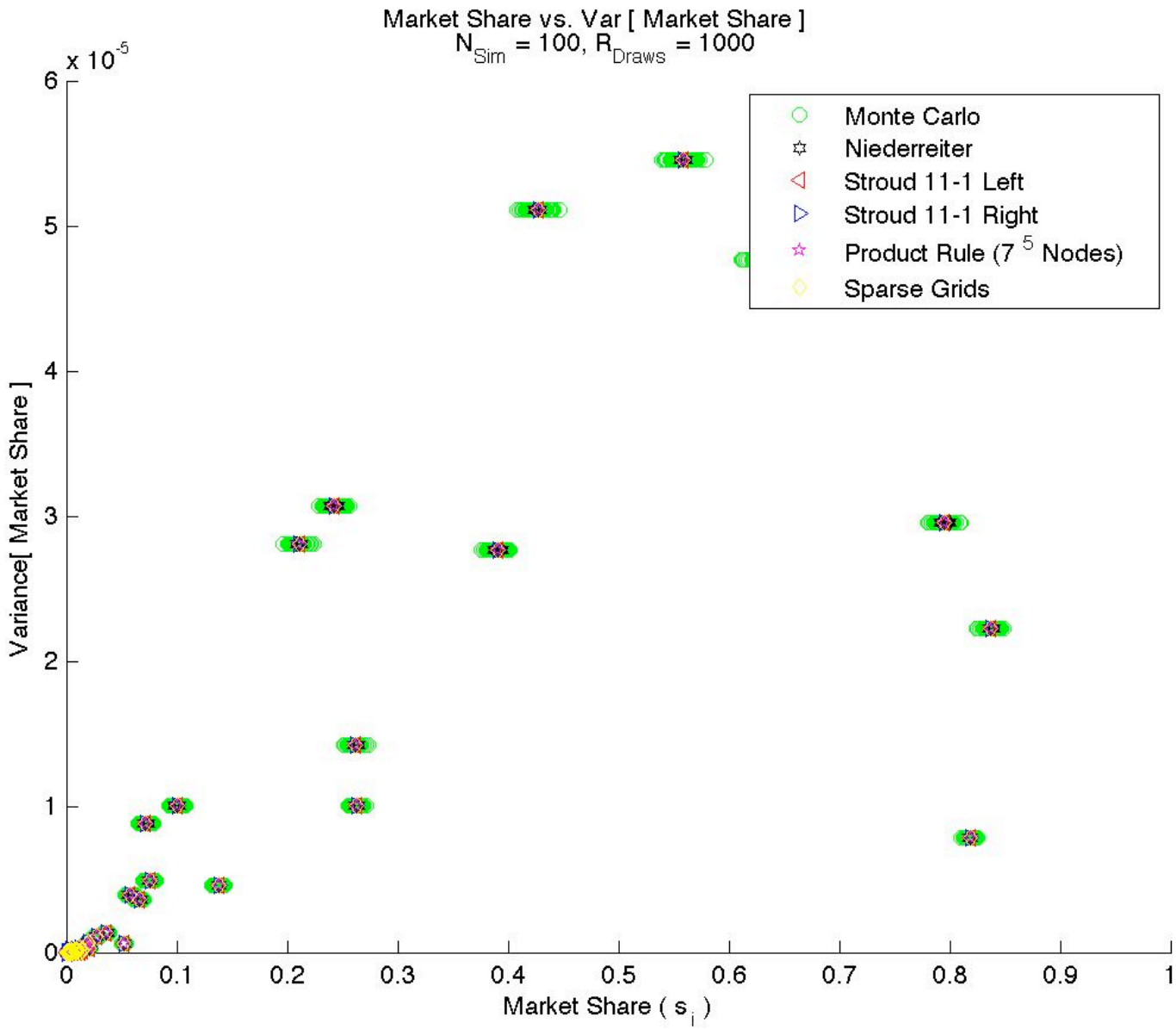
Remark: The kernel $\phi\left(\cdot\right)$ disappears because either we take pseudo-random draws from $\mathrm{N}\left(0,1\right)$, transform qMC numbers via $n=\Phi^{-1}\left(u\right)$, or use a polynomial rule with $w\left(x\right)=\exp\left(-x^2\right)$

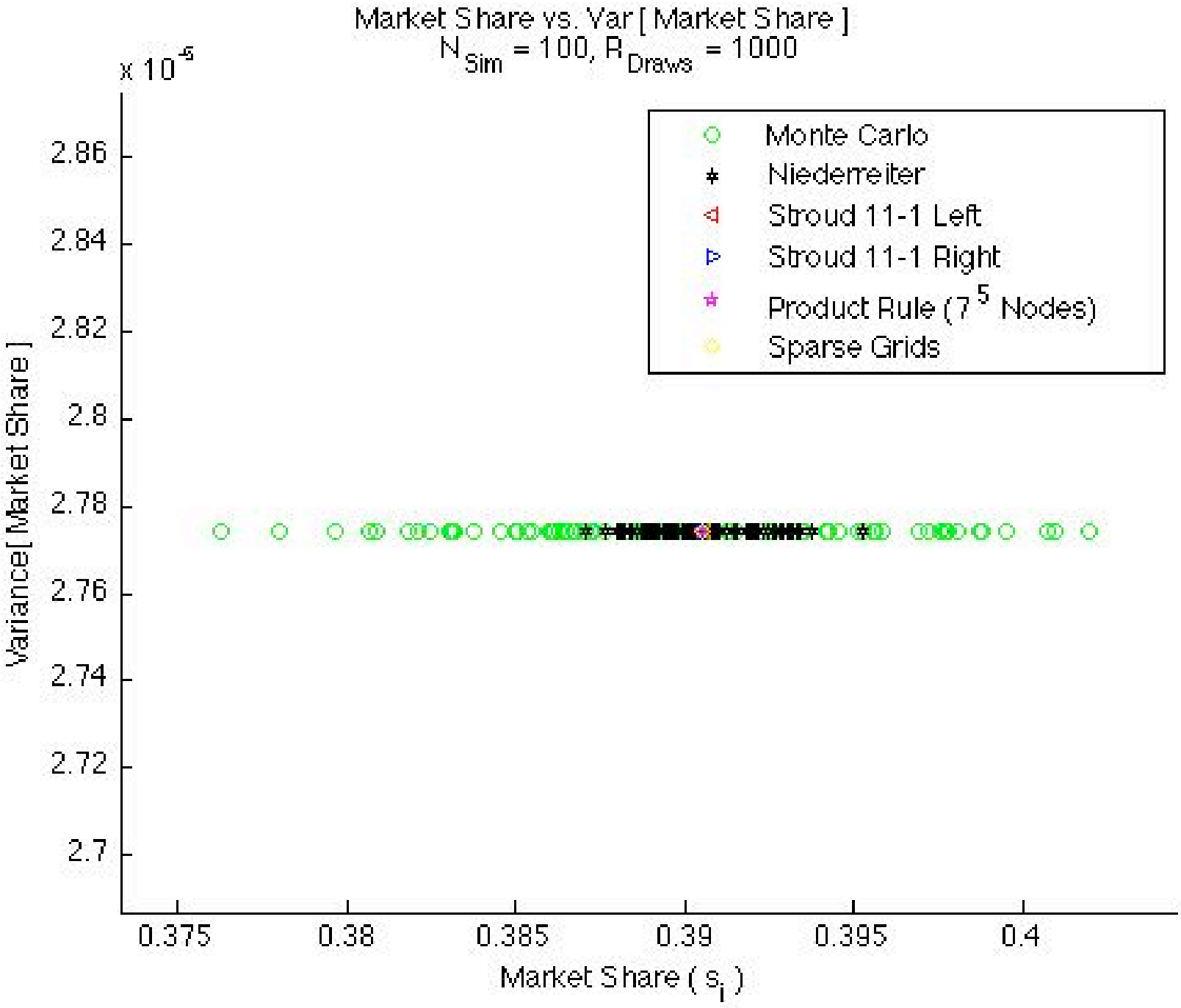
Remark: A Cholesky decomposition of the variance is necessary to transform the nodes

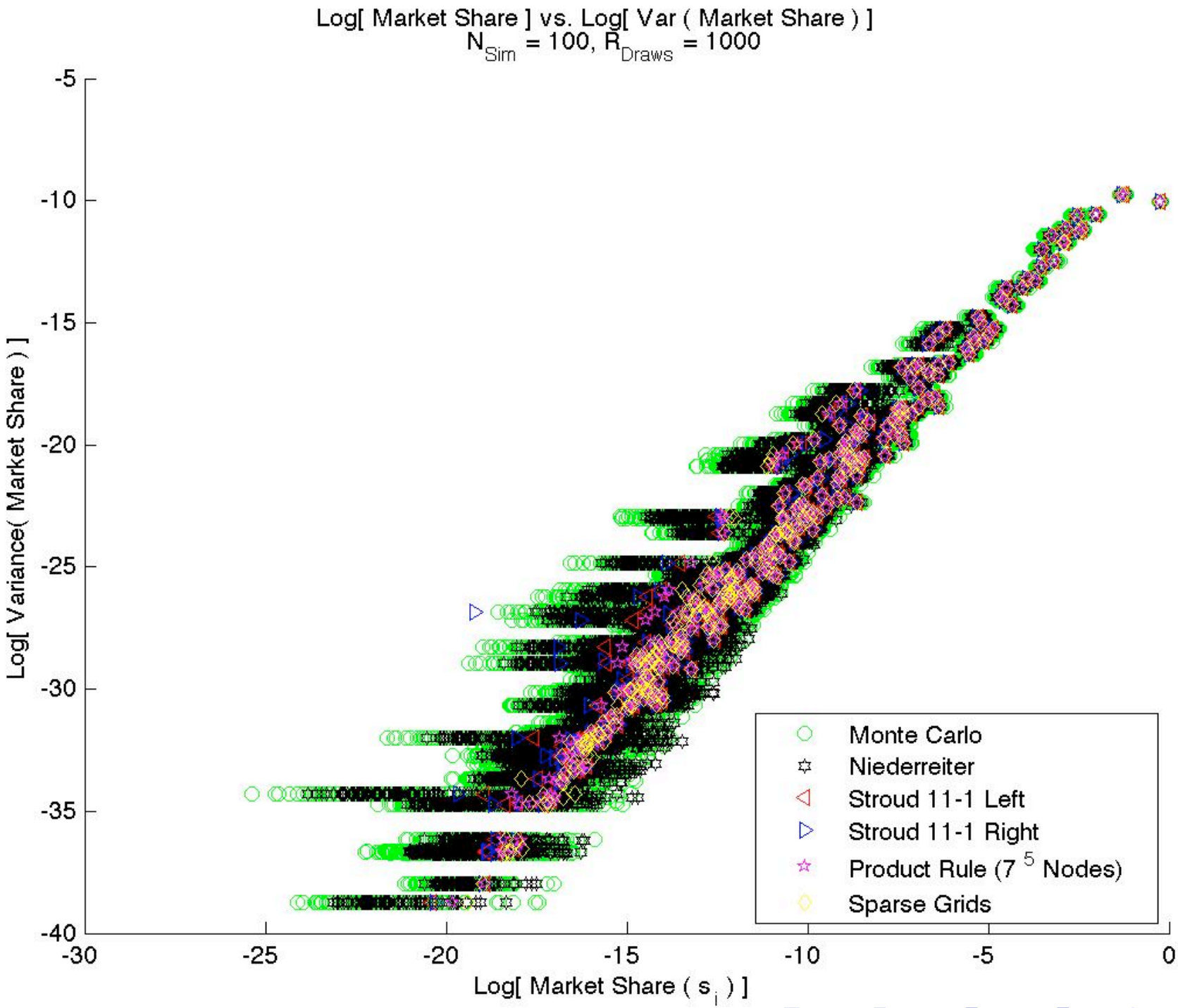
Share Results

Polynomial rules clearly superior to pMC:

- Polynomials approximate logit well
- All polynomial-based rules clustered in center of MC cloud, usually at exactly the same point
- Polynomial-based rules close to mean of pMC simulations, as expected, because pMC is unbiased.
- Monomial rule and sparse grids much more efficient in terms of points than GH product rule or MC.
- qMC has significantly less variance that pMC


Market Share vs. Var [Market Share]
N_Sim = 100, R_Draws = 1000
x 10^-5
Variance[Market Share]
Market Share (s_i)
Monte Carlo
Niederreiter
Stroud 11-1 Left
Stroud 11-1 Right
Product Rule (7^5 Nodes)
Sparse Grids
0
1
2
3
4
5
6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Market Share vs. Var [Market Share]
N Sim = 100, R Draws = 1000
x 10^-5
2.86
2.84
2.82
2.8
2.78
2.76
2.74
2.72
2.7
Variance[Market Share]
0.375
0.38
0.385
0.39
0.395
0.4
Market Share (s_i)
Monte Carlo
Niederreiter
Stroud 11-1 Left
Stroud 11-1 Right
Product Rule (7^5 Nodes)
Sparse Grids


Log[Market Share] vs. Log[Var (Market Share)]
N_Sim = 100, R_Draws = 1000
Log[Variance(Market Share)]
Log[Market Share (s_i)]
-5
-10
-15
-20
-25
-30
-35
-40
-30
-25
-20
-15
-10
-5
0
Monte Carlo
Niederreiter
Stroud 11-1 Left
Stroud 11-1 Right
Product Rule (7 5 Nodes)
Sparse Grids

Market Shares in Numbers

Rule	N_{nodes}	Max Abs Error	Min Abs Error
pMC	100	2.02858e-02	3.52172e-11
	1,000	7.88069e-03	2.55065e-11
	10,000	6.77537e-04	2.60308e-12
qMC	100	1.14975e-02	3.91531e-11
	1,000	3.55180e-03	2.72549e-11
	10,000	5.98465e-04	1.19251e-12
Product Rule	$3^5 = 243$	5.37642e-04	4.48538e-11
	$4^5 = 1,024$	3.90495e-05	2.69840e-11
	$5^5 = 3,125$	6.91544e-06	9.02103e-12
	$7^5 = 16,807$	0	0
	$9^5 = 59,049$	1.26292e-06	2.19342e-13
Stroud 11-1	983	1.03309e-04	3.27133e-13
	983	1.60621e-04	4.88308e-14
Sparse	993	4.53294e-05	4.56641e-14

* Errors relative to GH product rule with 7^5 nodes.

Asymptotic Comparison

Rule	N_{nodes}	Max Abs Error	Min Abs Error
pMC	100	6.76374e-02	2.75864e-13
	1,000	2.20766e-02	8.83621e-14
	10,000	8.58756e-03	5.71953e-14
qMC	100	2.75487e-02	2.23273e-13
	1,000	5.17365e-03	5.58996e-13
	10,000	9.13809e-04	2.00144e-14

- Comparison is now made over $N = 100$ simulations
- Increased probability of extremely large and small deviations from the Gauss-Hermite product rule with 7^5 nodes

Sensitivity Analysis

Initial tests show computed share values are very sensitive to $\hat{\theta}$:

- Most shares are very small (< 0.01)
- Only a few products in each market have large share values and appear to determine parameter estimates
- Larger shares have larger variances with MC rules because small shares are essentially 0
- Simple tests show that $> 10\%$ shares can move into/out of top decile of computed share values for small changes in $\hat{\theta}$
- MC exacerbates this sensitivity

Solver Convergence

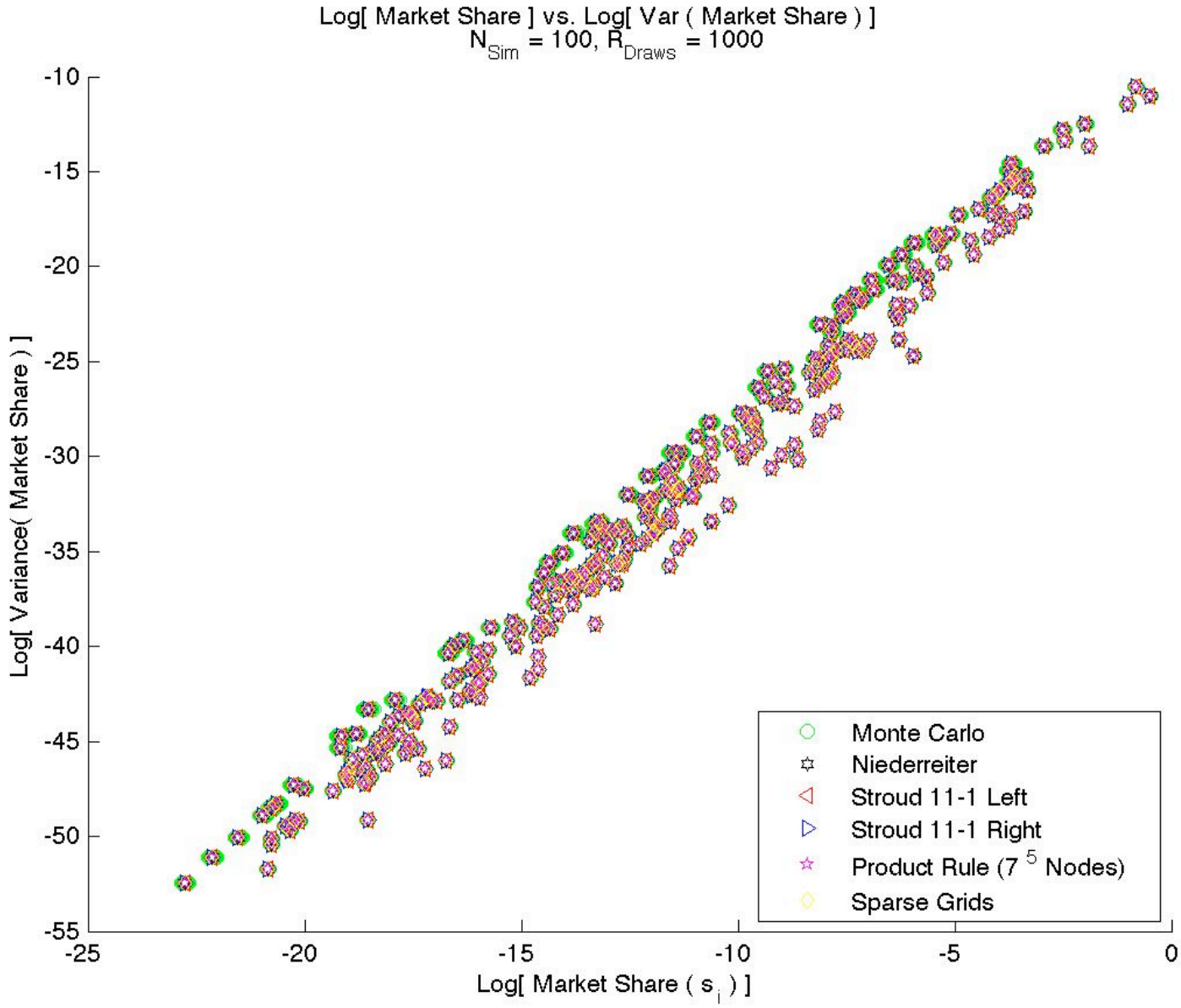
We compute point estimates:

- Five starting values for a variety of quadrature rules and draws
- Five MC data sets
- Used both KNITRO and SNOPT without significant change in results
- 'non-convergence' means:
  - Exceeded iteration limit of 100
  - Solver aborts with an error message, e.g. the problem is unbounded or it cannot make progress, such as all constraints cannot be satisfied
- With MPEC, quadrature rules only affect result via constraint that observed market shares equal calculated market shares

Solver Convergence

We find that the solver fails to converge for better approximations of the integral:

- Polynomial-based rules never converge
- pMC and qMC rules produce different point estimates
- Different starting values produce different point estimates and values of GMM objective function
- pMC and qMC do not converge for all starting values
- qMC produces better solutions than pMC:
  - Different solutions with qMC very similar, unlike pMC
  - Usually, requires fewer iterations to converge
  - At least 3x faster and often 50x faster
  - Produces different point estimates and converges more often than pMC
- No rules converge for the data set 0005 which has abnormally low variance vis-a-vis other MC data sets


Log[Market Share] vs. Log[Var (Market Share)]
N_Sim = 100, R_Draws = 1000
Log[Variance(Market Share)]
Log[Market Share (s_i)]
-10
-15
-20
-25
-30
-35
-40
-45
-50
-55
-25
-20
-15
-10
-5
0
Monte Carlo
Niederreiter
Stroud 11-1 Left
Stroud 11-1 Right
Product Rule (7 5 Nodes)
Sparse Grids

Possible Causes of Non-convergence

Remark: Non-convergence corroborates the story of an identification problem

Remark: Walker (2002) shows that taking too few draws will mask an identification problem in mixed logit models

$\Rightarrow$ Niederreiter MC rule was clearly superior to pMC

- That SNOPT 7 is often superior to KNITRO supports this story be SNOPT 7 was recently upgraded to handle rank deficient systems better

Berry's Mapping

Berry's (1994) 'contraction' mapping drives estimation procedure:

- Used to invert calculated market shares $S_{jt} = s_{jt}\left(\delta\left(\xi;\theta_1\right);\theta_2\right)$ to obtain ξ_{jt}
- A contraction in theory (Berry, 1994), but does Berry's theorem hold for MC approximation?
- BLP use importance sampling but few other authors employ variance reduction or qMC
  - I met someone at the scanner data conference using Halton draws, but he was only taking 30 draws....
  - Conlon (2009) uses sparse grids – the only non-pMC BLP paper I know of
- Expensive to compute even with Nevo's transform
- Stopping criterion: $\left\|\exp\left[\delta^{n+1}\right] - \exp\left[\delta^{n}\right]\right\| < \epsilon_{Inner}$

Convergence of Berry's Mapping

We find:

- Berry's mapping converges slowly at best, typically requiring $\sim 1,000$ iterations (not Gaussian!)
- Approximation of integral affects results:
  - No convergence for any of the polynomial rules
  - Convergence for pMC and qMC requires more iterations as number of draws, R, increases
- Convergence statistics number of iterations until convergence ($N = 100$, $T = 20$, $J = 15$, and $R = 1,000$ draws for pMC with ℓ^∞ norm and absolute tolerance of 10^{-14})

Statistic	**Value**
mean	1133.70
median	1139
variance	2909.11
N	94

- Note: 6 of 100 simulations failed to converge!

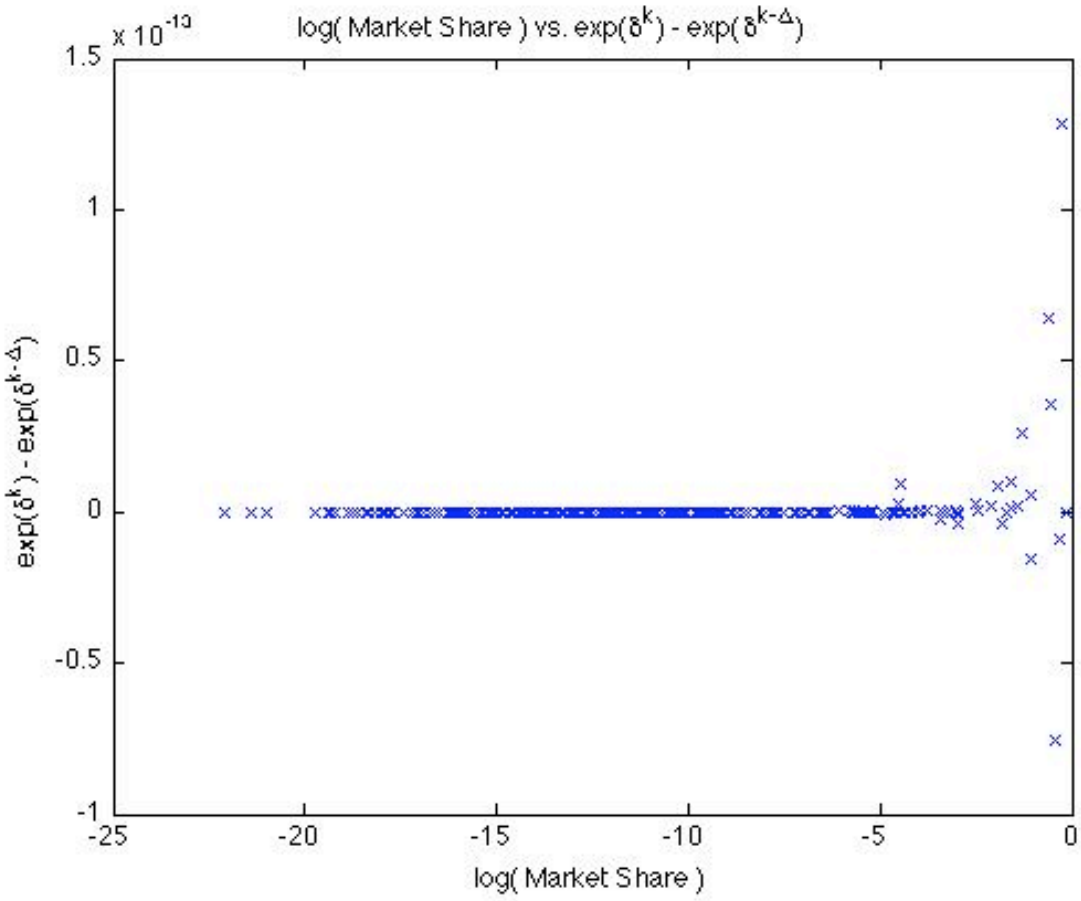
Properties of the Mapping

We also investigate the properties of Berry's mapping:

- Jacobian of mapping is nearly singular:
  - Condition number ~ 19
  - A few eigenvalues near 1
  - The majority close to 0
- We estimate rate of contraction, λ, using (Judd, 1998)

$$\lambda = \left[\left\| \exp\left(\delta^{k+2\Delta}\right) - \exp\left(\delta^{k+\Delta}\right) \right\| / \left\| \exp\left(\delta^{k+\Delta}\right) - \exp\left(\delta^{k}\right) \right\| \right]^{1/\Delta}$$

- λ is always close to 1 and often exceeds it, i.e. the Berry's approximate mapping diverges
- Decreasing T and J slows convergence which should facilitate convergence for a block diagonal system unless increasing size of the system increases chance of finding a local basin of attraction due to MC simulation error
- An identification problem?
- It appears that MC creates local regions of attraction where incorrect solutions are possible



Remark: Berry's mapping has trouble making progress for larger share values ($\Delta = 25$)

Future Research

Custom Monomial Rules:

- Given importance of mixed logit, use modern tools to create custom rules which exploit structure of economic problems such as mixed logit with normally-distributed taste shocks
- Ideal theory + homotopy
- For maximum efficiency and accuracy

Future Research

Convergence Issues:

- Is the approximate Berry mapping a contraction?
- Does the solution from Berry's map agree with better methods for solving non-linear equations such as homotopy?
- What drives non-convergence of solvers? Lack of identification?

Identification:

- Under what conditions is BLP identified?
- When does poor integration mask identification problems in BLP

Conclusion

There are several dangers to poor numerical approximations of integrals:

- Inaccurate results
- Much greater computational costs
- Masking of identification problems

For multidimensional problems of moderate dimension:

- Monomial-rules and Sparse Grids Integration are best options for integrals unless integrating over a very large number of dimensions
- Polynomial-based rules provide superior efficiency and accuracy

Of the MC methods, qMC was significantly more efficient than pMC and had much lower variance

Conclusion

Use of pMC in BLP models cause several problems:

- Inaccurate share values
- Different solutions for different starting values
- Non-convergence of solver and Berry's mapping
- These problems are usually a sign of identification problems