Sequential algorithm for structural estimations with equilibrium constraints

Takeshi Fukasawa

Waseda University

December 11, 2025, DSE conference

Structural estimation with equilibrium constraints

$$\min_{\theta, Y} Q(\theta, Y) \; s.t. \; G(Y;\theta) = 0$$

- θ: Structural parameters; Y: Nuisance parameters
- $Q(\theta, Y)$: Statistical objective function (e.g., $(-1)\times$Log likelihood, GMM objective)
- $G(Y;\theta) = 0$: Constraint based on economic model

- Examples:
  - Dynamic discrete choice (DDC) models and games (e.g., Rust, 1987)
    - Value functions or CCPs corresponds to Y
  - Static / Dynamic BLP (e.g., Berry et al., 1995, Gowrisankaran and Rysman (2012))
    - Mean product utility corresponds to Y
  - General equilibrium models

NFXP Algorithm

- Widely applied approach: **Nested Fixed Point (NFXP) Algorithm**
  - Inner-loop: Given structural parameters θ, solve for $Y(\theta)$ s.t. $G\left(Y(\theta);\theta\right)=0$
  - Outer-loop: Search for θ_* minimizing the objective function $Q(\theta, Y(\theta))$
- Efficient implementations have been investigated by many studies
  - DDC: Iskhakov et al. (2016), Bray (2019), Aguirregabiria and Magesan (2023) etc.
  - BLP: Conlon and Gortmaker (2020), Fukasawa (2024) etc.

NFXP Algorithm

- Widely applied approach: **Nested Fixed Point (NFXP) Algorithm**
  - Inner-loop: Given structural parameters θ, solve for $Y(\theta)$ s.t. $G\left(Y(\theta);\theta\right) = 0$
  - Outer-loop: Search for θ_* minimizing the objective function $Q(\theta, Y(\theta))$
- Efficient implementations have been investigated by many studies
  - DDC: Iskhakov et al. (2016), Bray (2019), Aguirregabiria and Magesan (2023) etc.
  - BLP: Conlon and Gortmaker (2020), Fukasawa (2024) etc.

- However, **computational costs** can be very large...
  - Large datasets, Complicated structural models,
    Multiple initial values, Bootstrap for computing standard errors...
  - Sometimes take more than several days just to estimate parameters...

$\Rightarrow$ Obstacle to further economic analyses

Sequential estimation

Alternative to NFXP: "Sequential estimation"

- Sequentially update θ and Y
  - NPL (Aguirregabiria and Mira (2002) for single-agent DDC)
  - ABLP (Lee and Seo (2015) for Static BLP)
  - EPL (Dearing and Blevins (2025) for Dynamic game)
- Under "Zero-Jacobian property", the limiting point coincides with the NFXP estimator
- Kasahara and Shimotsu (2008) and Dearing and Blevins (2025) showed statistical convergence property of EPL/NPL
- Previous studies showed good performance
- However, their performance/applicability rely on the specific structure of the structural model...
  - Form of $G(Y;\theta)$ (NPL)
  - Linearity of $G(Y;\theta)$ wrt θ (EPL)

This study

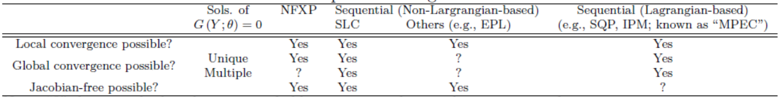
This study:

1. Investigate the convergence of sequential algorithms with Zero-Jacobian Property (ZJP)
    - Generalize the discussions in Kasahara and Shimotsu (2008) (NPL for single-agent DDC) and Dearing and Blevins (2025) (EPL)
    - If we consider **MLE or GMM**,
        - Fast (Near-quadratic) local convergence in large samples, **even absent initial consistent estimates** (θ, Y)
        - Initial consistent (θ, Y) available ⇒ Asymptotically **efficient after one iteration**
        - The property $\nabla_Y Q(\theta_*, Y_*) = O_p\left(N^{-\frac{1}{2}}\right)$ is the key, where N: Number of samples

2. Propose a novel **Sequential Linearly Constrained (SLC)** Algorithm
    - Applicable to broader structural models

Advantages of proposed SLC algorithm

Table 1: Comparison of algorithms



	Sols. of	NFXP	Sequential (Non-Largrangian-based)		Sequential (Lagrangian-based)
	$G(Y;\theta)=0$		SLC	Others (e.g., EPL)	(e.g., SQP, IPM; known as "MPEC")
Local convergence possible?		Yes	Yes	Yes	Yes
Global convergence possible?	Unique	Yes	Yes	?	Yes
	Multiple	?	Yes	?	Yes
Jacobian-free possible?		Yes	Yes	Yes	?

Note. "?" denotes whether the property holds or not is unclear in the previous studies.

- Compared to NFXP,
  - Robust even when multiple solutions of $G(Y;\theta)=0$ exist
  - Several times faster than NFXP in some models
- Compared to "MPEC" (Lagrangian-based),
  - Can be implemented without explicitly computing $\nabla_Y G(Y;\theta)$
    - The Jacobian matrix $\nabla_Y G(Y;\theta)$ can be high-dimensional $\Rightarrow$ Sometimes require large memory/ comp. time...
- Compared to Two-step methods (e.g., Hotz and Miller, 1993; Chernozhukov et al., 2022),
  - Applicable even without consistent initial Y
  - Robust even in small samples

Sequential algorithm

Sequential algorithm

Set initial values $\gamma_0 \equiv (\theta_0, Y_0)$. Iterate the following $(k = 0, 1, 2, \cdots)$:

1. Compute $\theta_{k+1} = \arg\min_\theta \widetilde{Q}\left(\theta; \gamma_k\right) \equiv Q\left(\theta, \Upsilon\left(\theta; \gamma_k\right)\right)$
2. Compute $Y_{k+1} = \Upsilon\left(\theta_{k+1}; \gamma_k\right)$.

Sequential algorithm

Sequential algorithm

Set initial values $\gamma_0 \equiv (\theta_0, Y_0)$. Iterate the following $(k = 0, 1, 2, \cdots)$:

1. Compute $\theta_{k+1} = \arg\min_\theta \widetilde{Q}\left(\theta; \gamma_k\right) \equiv Q\left(\theta, \Upsilon\left(\theta; \gamma_k\right)\right)$
2. Compute $Y_{k+1} = \Upsilon\left(\theta_{k+1}; \gamma_k\right)$.

Examples of $\Upsilon\left(\theta; \gamma_k\right)$:

- EPL: $\Upsilon\left(\theta; \gamma_k\right) \equiv Y_k - \left(\nabla_Y G\left(Y_k; \theta_k\right)\right)^{-1}\left(G\left(Y_k; \theta\right)\right)$
- ABLP: $\Upsilon\left(\theta; \gamma_k\right) \equiv Y_k - \left(\nabla_Y G\left(Y_k; \theta\right)\right)^{-1}\left(G\left(Y_k; \theta\right)\right)$
- NPL (single-agent DDC): $\Upsilon\left(\theta; \gamma_k\right) \equiv \Psi\left(Y_k; \theta\right)$ where Ψ satisfies $\Psi\left(Y; \theta\right) = Y \Rightarrow \nabla_Y \Psi\left(Y; \theta\right) = 0$. Let $G\left(Y; \theta\right) = Y - \Psi\left(Y; \theta\right)$.

They satisfy:
(a). $\Upsilon\left(\theta; \gamma = (\theta, Y)\right) = Y \Leftrightarrow G\left(Y; \theta\right) = 0$
(b). $G\left(Y; \theta\right) = 0 \Rightarrow \nabla_\gamma \Upsilon\left(\theta; \gamma = (\theta, Y)\right) = 0$ (Zero Jacobian property)
(c). $G\left(Y; \theta\right) = 0 \Rightarrow \nabla_\theta \Upsilon\left(\theta; \gamma = (\theta, Y)\right) = -\left(\nabla_Y G\left(Y; \theta\right)\right)^{-1}\left(\nabla_\theta G\left(Y; \theta\right)\right)$

Sol. of $\min_{\theta, Y} Q(\theta, Y)\ s.t.\ G(Y; \theta) = 0$: Fixed point of the iteration

Local convergence speed of sequential algorithms with ZJP

Let $\gamma_* \equiv (\theta_*, Y_*)$ be the solution. Then,

$$\gamma_k - \gamma_* = \exists A(\gamma_*)\left(\nabla_Y Q(\theta_*, Y_*)\right)\left(\nabla_{\theta\gamma'}\Upsilon(\theta_*, \gamma_*)\right)(\gamma_{k-1} - \gamma_*) + O\left(\|\gamma_{k-1} - \gamma_*\|^2\right)$$

where $A(\gamma_*) \equiv \begin{bmatrix} I \\ \nabla_\theta \Upsilon(\theta_*, \gamma_*) \end{bmatrix} \left(\nabla_{\theta\theta'}\widetilde{Q}(\theta_*, \gamma_*)\right)^{-1}$

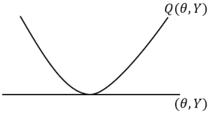
- If $\nabla_Y Q(\theta_*, Y_*) \approx 0$, $\gamma_k - \gamma_* \approx O\left(\|\gamma_{k-1} - \gamma_*\|^2\right)$⇒Near Quadratic convergence
- In statistical settings (MLE/GMM), $\nabla_Y Q(\theta_*, Y_*) \approx 0$ holds in large samples.

Property of $\nabla_Y Q(\theta_*, Y_*) = O_p\left(N^{-\frac{1}{2}}\right)$

We can easily show $\nabla_Y Q(\theta_*, Y_*) = O_p\left(N^{-\frac{1}{2}}\right)$ for MLE / GMM Derivations

Intuition:

- **In large samples**, MLE/GMM are equivalent to $\min_{\theta, Y} Q(\theta, Y)\ \ s.t.\ G(Y; \theta) = 0$, where $Q(\theta, Y) \geq 0$, $Q\left(\theta_{(true)}, Y_{(true)}\right) = 0$ $\Rightarrow \nabla_Y Q\left(\theta_{(true)}, Y_{(true)}\right) = 0$
  - MLE: Equivalent to minimizing the Kullback–Leibler Info. Criterion *KLIC*
    - $KLIC \geq 0$, $KLIC = 0$ at $\left(\theta_{(true)}, Y_{(true)}\right)$
  - GMM: GMM objective≥ 0, GMM objective$= 0$ at $\left(\theta_{(true)}, Y_{(true)}\right)$



Statistical / Computational Efficiency

- Suppose $\sqrt{N}\left(\theta_{*}-\theta_{(true)}\right) \to_{d} N\left(0, \Sigma_{NFXP}\right)$.
- **If initial consistent** $\gamma_0 \equiv \left(\theta_0, Y_0\right)$ **available,**
    - $\sqrt{N}\left(\theta_{k}-\theta_{(true)}\right) \to_{d} N\left(0, \Sigma_{NFXP}\right)$ (Same efficiency compared to NFXP)
    - $\gamma_0-\gamma_{(true)}=O_p\left(N^{-b}\right) \;\Rightarrow\; \gamma_k-\gamma_{*}=O_p\left(N^{-(k-1)/2-2b}\right)$ $\left(b\in(1/4,1/2]\right)$
    - $\sqrt{N}$-consistency of initial value γ_0 not required, unlike Dearing and Blevins (2025)
        - cf. Kasahara and Shimotsu (2008) (NPL for single-agent DDC)
- **Even when initial consistent** $\gamma_0 \equiv \left(\theta_0, Y_0\right)$ **not available,**
    - $\gamma_k-\gamma_{*}=O_p\left(N^{-1/2}\left\|\gamma_{k-1}-\gamma_{*}\right\|+\left\|\gamma_{k-1}-\gamma_{*}\right\|^{2}\right)$(near-quadratic local convergence in large samples)
- Proof: Use $\nabla_{Y} Q\left(\theta_{*}, Y_{*}\right)=O_p\left(N^{-\frac{1}{2}}\right)$ and ZJP

SLC algorithm

- Proposed SLC (Sequential Linearly Constrained) algorithm: Use

$$\begin{aligned}\Upsilon(\theta;\gamma_k) \quad \equiv \quad & Y_k - \left[(\nabla_Y G(Y_k;\theta_k))^{-1}(G(Y_k;\theta_k))\right] \\ & - \left[(\nabla_Y G(Y_k;\theta_k))^{-1}(\nabla_\theta G(Y_k;\theta_k))\right](\theta-\theta_k)\end{aligned}$$

  - $(\nabla_Y G(Y_k;\theta_k))^{-1}(G(Y_k;\theta_k))$ and $(\nabla_Y G(Y_k;\theta_k))^{-1}(\nabla_\theta G(Y_k;\theta_k))$ can be computed before the optimization $\min_\theta \widetilde{Q}(\theta;\gamma_k) \equiv Q(\theta, \Upsilon(\theta;\gamma_k))$
    $\Rightarrow$ Computationally much less costly
  - Coincide with EPL if $G(Y;\theta)$ is linear wrt θ Relations with EPL
- Remark: SLC is equivalent to sequentially solving the following:

$$\begin{aligned}\min_{\theta,Y} \quad & Q(\theta, Y) \\ s.t. \quad & (\nabla_\theta G(Y_k;\theta_k))(\theta-\theta_k) + (\nabla_Y G(Y_k;\theta_k))(Y-Y_k) + G(Y_k;\theta_k) = 0\end{aligned}$$

  - Constraint: Linearization of $G(Y;\theta) = 0$ around (Y_k, θ_k)
- Additional good property of SLC:
  - Descent direction given l_1-norm merit function $\phi_1(\theta, Y;\mu) \equiv Q(\theta, Y) + \mu \|G(Y;\theta)\|_1$ $(\mu > 0)$
    - cf. Nocedal and Wright (2006) (Textbook on numerical optimization)
    - Possible to show global/local convergence results by introducing a line search

Jacobian-free implementations of SLC/EPL

- To compute $\Upsilon\left(\theta;\gamma_k\right) \equiv Y_k - \left(\nabla_Y G\left(Y_k;\theta_k\right)\right)^{-1}\left(G\left(Y_k;\theta_k\right)\right) - \left(\nabla_Y G\left(Y_k;\theta_k\right)\right)^{-1}\left(\left(\nabla_\theta G\left(Y_k;\theta_k\right)\right)\left(\theta-\theta_k\right)\right)$, require computing the Jacobian matrix $\nabla_Y G\left(Y_k;\theta_k\right)$?
  - High dimensional $(n_Y \times n_Y)$ matrix, if n_Y is large? ⇒Large memory storage, Large computation time?
  - Need manually deriving the derivatives? Remarks on Automatic Differentiation

Jacobian-free implementations of SLC/EPL

- To compute $\Upsilon(\theta;\gamma_k) \equiv Y_k - (\nabla_Y G(Y_k;\theta_k))^{-1}(G(Y_k;\theta_k)) - (\nabla_Y G(Y_k;\theta_k))^{-1}((\nabla_\theta G(Y_k;\theta_k))(\theta-\theta_k))$, require computing the Jacobian matrix $\nabla_Y G(Y_k;\theta_k)$?
  - High dimensional $(n_Y \times n_Y)$ matrix, if n_Y is large? ⇒Large memory storage, Large computation time?
  - Need manually deriving the derivatives? Remarks on Automatic Differentiation

- ⇒We can avoid explicit computation of $\nabla_Y G(Y_k;\theta_k)$, by relying on Jacobian-free approach based on **Krylov method** Details
  - Krylov method: Packages available in many programming languages

Numerical experiments

- **Compare NFXP** (numerical derivatives) and **SLC** (Jacobian-free)
  - NFXP: Use fixed-point iteration acceleration method (Anderson acceleration)

1. **Dynamic demand model (Dynamic BLP)**
   - Dynamic BLP: Dynamic extension of Berry et al. (1995)'s static BLP model
     - e.g., Durable goods, Switching costs
   - Model of (perfectly) durable goods (cf. Sun and Ishihara (2019))
     - Random coefficients, Price endogeneity, Forward-looking decisions, Aggregate data
     - NFXP: Use the fixed-point mapping proposed by Fukasawa (2024)
     - Several times faster than traditional approach

2. **Dynamic game with time-varying unobserved heterogeneity**
   - Dynamic entry/exit game
     - Introduce time-varying unobserved states in Aguirregabiria and Mira (2007) and Dearing and Blevins (2025)'s setting
     - Similar to Arcidiacono and Miller (2011)'s setting

1. Dynamic demand: Results

	Comp. Time (sec)						Feval Q	Feval G	Converged	iter.
	Mean	Min.	25th	Median.	75th	Max.	(mean)	(mean)		SLC
SLC	9.4	4.7	6.5	9.2	11.8	17.3	102424.5	7142.6	100	10.7
NFXP	160	37.7	52.8	59.7	74.8	2969.5	16750.4	120641.6	100	-
Comp(G) (sec)						0.0070				
Comp(Q) (sec)						0.0021				

2. Dynamic game: Results

	Comp. Time (sec)						Feval Q	Feval G	Converged	iter.
	Mean	Min.	25th	Median.	75th	Max.	(mean)	(mean)		(SLC)
SLC	39.5	16.5	23.5	31.5	58.5	127.7	9279.2	4844.4	100	13.0
NFXP	229.7	23.6	164.7	212.4	263.4	973.7	778.1	28668.5	100	-
Comp(G) (sec)						0.0054				
Comp(Q) (sec)						0.0010				

Relations to Neyman orthogonality

- Recent studies proposed estimation methods utilizing Neyman orthogonality for statistically efficient structural estimations (e.g., Chernozhukov et al. (2022), Sawadogo (2025))

- Sequential estimations with ZJP satisfies:
  - MLE: $\left[\nabla_{Y} E\left[\nabla_{\theta} \widetilde{Q}\left(\theta_{(true)}; \gamma_{(true)}\right)\right]\right]\left(\gamma - \gamma_{(true)}\right) = 0\ \forall\gamma$ (Neyman orthogonality)
  - GMM: $\left[\nabla_{Y}\left[\nabla_{\theta} \widetilde{Q}\left(\theta_{(true)}; \gamma_{(true)}\right)\right]\right]\left(\gamma - \gamma_{(true)}\right) = O_p\left(N^{-\frac{1}{2}}\right)\ \forall\gamma$ (Asymptotic Neyman orthogonality)

- $\Rightarrow$ FOC $\nabla_{\theta} \widetilde{Q}\left(\theta_{k+1}; \gamma\right) = 0$ for $\theta_{k+1} = \arg\min_{\theta} \widetilde{Q}\left(\theta; \gamma_k\right) \equiv Q\left(\theta, \Upsilon\left(\theta; \gamma_k\right)\right)$ would be rarely affected by the nuisance parameter γ
  $\Rightarrow$ Locally efficient and stable estimation

Conclusions

- **Investigate convergence properties of sequential algorithms with ZJP**
  - e.g., NPL (single-agent DDC), ABLP, EPL, and SLC
  - Quadratic convergence in large samples for MLE/GMM
  - Initial consistent values available ⇒Asymptotically efficient even after one iteration
  - Initial consistent values not available ⇒ Work as algorithms to solve $\min_{\theta,Y} Q(\theta, Y) \;\; s.t. \;\, G(Y;\theta) = 0$
- **Propose SLC algorithm**
  - Jacobian-free approach available
  - Several times faster than NFXP in some dynamic models
  - Practical alternative to NFXP / "MPEC" (Lagrangian-based) in broader structural models

References I

Aguirregabiria, V. and Magesan, A. (2023). Solving discrete choice dynamic programming models using euler equations. Technical report, mimeo.

Aguirregabiria, V. and Mira, P. (2002). Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica*, 70(4):1519–1543.

Aguirregabiria, V. and Mira, P. (2007). Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53.

Arcidiacono, P. and Miller, R. A. (2011). Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79(6):1823–1867.

Berry, S., Levinsohn, J., and Pakes, A. (1995). Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890.

Bray, R. L. (2019). Strong convergence and dynamic economic models. *Quantitative Economics*, 10(1):43–65.

References II

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., and Robins, J. M. (2022). Locally robust semiparametric estimation. *Econometrica*, 90(4):1501–1535.

Conlon, C. and Gortmaker, J. (2020). Best practices for differentiated products demand estimation with PyBLP. *The RAND Journal of Economics*, 51(4):1108–1161.

Dearing, A. and Blevins, J. R. (2025). Efficient and convergent sequential pseudo-likelihood estimation of dynamic discrete games. *Review of Economic Studies*, 92(2):981–1021.

Fukasawa, T. (2024). Fast and simple inner-loop algorithms of static/dynamic BLP estimations. *arXiv preprint arXiv:2404.04494*.

Fukasawa, T. (2025). Jacobian-free Efficient Pseudo-Likelihood (EPL) Algorithm. *Economics Letters*, 112195.

Gowrisankaran, G. and Rysman, M. (2012). Dynamics of consumer demand for new durable goods. *Journal of Political Economy*, 120(6):1173–1219.

References III

Hotz, V. J. and Miller, R. A. (1993). Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529.

Hu, Y. and Shum, M. (2012). Nonparametric identification of dynamic models with unobserved state variables. *Journal of Econometrics*, 171(1):32–44.

Iskhakov, F., Lee, J., Rust, J., Schjerning, B., and Seo, K. (2016). Comment on constrained optimization approaches to estimation of structural models. *Econometrica*, 84(1):365–370.

Kasahara, H. and Shimotsu, K. (2008). Pseudo-likelihood estimation and bootstrap inference for structural discrete Markov decision models. *Journal of Econometrics*, 146(1):92–106.

Lee, J. and Seo, K. (2015). A computationally fast estimator for random coefficients logit demand models using aggregate data. *The RAND Journal of Economics*, 46(1):86–102.

References IV

Murtagh, B. A. and Saunders, M. A. (1982). A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Mathematical Programming Study*, 16:84–117.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization (Second Edition)*. Springer.

Pál, L. and Sándor, Z. (2023). Comparing procedures for estimating random coefficient logit demand models with a special focus on obtaining global optima. *International Journal of Industrial Organization*, 88:102950.

Rust, J. (1987). Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, 55(5):999–1033.

Sawadogo, S. (2025). Efficient and Debiased Estimation of Dynamic Discrete Choice Models. *Available at SSRN 5325574*.

Su, C.-L. and Judd, K. L. (2012). Constrained optimization approaches to estimation of structural models. *Econometrica*, 80(5):2213–2230.

Sun, Y. and Ishihara, M. (2019). A computationally efficient fixed point approach to dynamic structural demand estimation. *Journal of Econometrics*, 208(2):563–584.

Derivation of $\nabla_Y Q\left(\theta_*, Y_*\right) = O_p\left(N^{-\frac{1}{2}}\right)$(MLE)

$$\min_{\theta, Y} \quad Q\left(\theta, Y\right) = -\frac{1}{N}\sum_{i=1}^{N} \ln f\left(w_i | \theta, Y\right) \;\; s.t. \;\; G\left(Y;\theta\right) = 0$$

We assume w_i is generated from $f^{(true)}\left(w|\theta^{(true)}, Y^{(true)}\right) = f^{(true)}\left(w\right)$.

By differentiating the both sides of $1 = \int f^{(true)}\left(w; \theta_{(true)}, Y_{(true)}\right) dw$ with regard to Y, we have:

$$0 \quad = \quad \int \left(\nabla_Y f^{(true)}\left(w; \theta_{(true)}, Y_{(true)}\right)\right) dw = \int \left(\nabla_Y \ln f^{(true)}\left(w; \theta_{(true)}, Y_{(true)}\right)\right) f^{(true)}\left(w\right) dw$$

Then,

$$\begin{aligned}
& \nabla_Y Q\left(\theta_*, Y_*\right) \\
= \quad & -\frac{1}{N}\sum_{i=1}^{N} \nabla_Y \ln f\left(w_i | \theta_*, Y_*\right) \\
= \quad & -\left[\frac{1}{N}\sum_{i=1}^{N} \nabla_Y \ln\left(\frac{f\left(w_i|\theta_*, Y_*\right)}{f^{(true)}\left(w_i; \theta_{(true)}, Y_{(true)}\right)}\right)\right] + \\
& -\left[\frac{1}{N}\sum_{i=1}^{N} \nabla_Y \left(\ln f^{(true)}\left(w_i; \theta_{(true)}, Y_{(true)}\right)\right) - \int \left(\nabla_Y \ln f^{(true)}\left(w; \theta_{(true)}, Y_{(true)}\right)\right) f^{(true)}\left(w\right) dw\right]
\end{aligned}$$

The first and second terms: $O_p\left(N^{-\frac{1}{2}}\right)$ under $f = f^{(true)}$. $\Rightarrow$ $\nabla_Y Q\left(\theta_*, Y_*\right) = O_p\left(N^{-\frac{1}{2}}\right)$

(See paper for more precise derivation)

Derivation of $\nabla_Y Q(\theta_*, Y_*) = O_p\left(N^{-\frac{1}{2}}\right)$ (GMM)

$$\min_{\theta, Y} \quad Q(\theta, Y) = (m(\theta, Y))^T \widehat{W} (m(\theta, Y)) \ s.t. \ G(Y; \theta) = 0$$

where $m(\theta, Y) = \frac{1}{N}\sum_{i=1}^{N} m_i(\theta, Y)$. Then,

$$\begin{aligned} \nabla_Y Q(\theta_*, Y_*) &= \nabla_Y \left[(m(\theta_*, Y_*))' \widehat{W} (m(\theta_*, \gamma_*)) \right] \\ &= 2 (\nabla_Y m(\theta_*, Y_*))' \widehat{W} (m(\theta_*, \gamma_*)) \\ &= O_p\left(N^{-\frac{1}{2}}\right) \left(\because m(\theta, Y) = \frac{1}{N}\sum_{i=1}^{N} m_i(\theta, Y) \right) \end{aligned}$$

(See paper for more precise derivation)

Derivation of $\gamma_k - \gamma_* = A\left(\gamma_*\right)\left(\nabla_Y Q\left(\theta_*, Y_*\right)\right)\left(\nabla_{\theta\gamma'}\Upsilon\left(\theta_*, \gamma_*\right)\right)\left(\gamma_{k-1} - \gamma_*\right) + O\left(\left\|\gamma_{k-1} - \gamma_*\right\|^2\right)$

Let $H\left(\gamma\right) \equiv \begin{pmatrix} H_1\left(\gamma\right) \\ H_2\left(\gamma\right) \end{pmatrix} \equiv \begin{pmatrix} \widetilde{\theta}\left(\gamma\right) \\ \Upsilon\left(\widetilde{\theta}\left(\gamma\right);\gamma\right) \end{pmatrix}$, where $\widetilde{\theta}\left(\gamma\right) \equiv \arg\min_{\theta\in\Theta} \widetilde{Q}\left(\theta;\gamma\right) = Q\left(\theta, \Upsilon\left(\theta;\gamma\right)\right)$. Then,

$$\nabla_\gamma H\left(\gamma\right) = \begin{bmatrix} \nabla_\gamma \widetilde{\theta}\left(\gamma\right) \\ \left(\nabla_\theta \Upsilon\left(\widetilde{\theta}\left(\gamma\right),\gamma\right)\right)\left(\nabla_\gamma \widetilde{\theta}\left(\gamma\right)\right) + \left(\nabla_\gamma \Upsilon\left(\widetilde{\theta}\left(\gamma\right),\gamma\right)\right) \end{bmatrix}$$

holds. Because $\nabla_\theta \widetilde{Q}\left(\theta_*,\gamma_*\right) = 0$ holds, $\nabla_\gamma \widetilde{\theta}\left(\gamma\right) = \left(\nabla_{\theta\theta'}\widetilde{Q}\left(\theta_*,\gamma_*\right)\right)^{-1}\left(\nabla_{\theta\gamma'}\widetilde{Q}\left(\theta_*,\gamma_*\right)\right)$ holds by the implicit function theorem. By $\widetilde{Q}\left(\theta,\gamma\right) = Q\left(\theta, \Upsilon\left(\theta,\gamma\right)\right)$, $\nabla_\gamma \widetilde{Q}\left(\theta,\gamma\right) = \left(\nabla_Y\left(\theta, \Upsilon\left(\theta,\gamma\right)\right)\right)\left(\nabla_\gamma \Upsilon\left(\theta,\gamma\right)\right)$ holds, and hence:

$$\begin{aligned} \nabla_{\theta\gamma'}\widetilde{Q}\left(\theta,\gamma\right) = {} & \left(\nabla_{\theta Y'} Q\left(\theta, \Upsilon\left(\theta,\gamma\right)\right)\right)\left(\nabla_\gamma \Upsilon\left(\theta;\gamma\right)\right) + \\ & \left(\nabla_{YY'} Q\left(\theta, \Upsilon\left(\theta,\gamma\right)\right)\right)\left(\nabla_\theta \Upsilon\left(\theta;\gamma\right)\right)\left(\nabla_\gamma \Upsilon\left(\theta;\gamma\right)\right) + \\ & \left(\nabla_Y Q\left(\theta, \Upsilon\left(\theta,\gamma\right)\right)\right)\left(\nabla_{\theta\gamma'}\Upsilon\left(\theta,\gamma\right)\right) \end{aligned} \tag{1}$$

Derivation of $\gamma_k - \gamma_* = A(\gamma_*)(\nabla_Y Q(\theta_*, Y_*))(\nabla_{\theta\gamma'}\Upsilon(\theta_*, \gamma_*))(\gamma_{k-1} - \gamma_*) + O\left(\|\gamma_{k-1} - \gamma_*\|^2\right)$ (Continued)

By Taylor's theorem, $H(\gamma_{k-1}) - H(\gamma_*) = (\nabla_\gamma H(\gamma_*))(\gamma_{k-1} - \gamma_*) + O\left(\|\gamma_{k-1} - \gamma_*\|^2\right)$. By $H(\gamma_{k-1}) = \gamma_k$ and $H(\gamma_*) = \gamma_*$,

$$\gamma_k - \gamma_* = (\nabla_\gamma H(\gamma_*))(\gamma_{k-1} - \gamma_*) + O\left(\|\gamma_{k-1} - \gamma_*\|^2\right) \tag{2}$$

Because $\nabla_\gamma \Upsilon(\theta_*; \gamma_*) = 0$ holds under the ZJP assumption, (1) and $\Upsilon(\theta_*; \gamma_*) = Y_*$ implies:

$$\nabla_{\theta\gamma'} \widetilde{Q}(\theta_*, \gamma_*) \quad = \quad (\nabla_Y Q(\theta_*, Y_*))(\nabla_{\theta\gamma'}\Upsilon(\theta_*, \gamma_*)) \tag{3}$$

Hence,

$$\gamma_k - \gamma_* \quad = \quad A(\gamma_*)(\nabla_Y Q(\theta_*, Y_*))(\nabla_{\theta\gamma'}\Upsilon(\theta_*, \gamma_*))(\gamma_{k-1} - \gamma_*) + O\left(\|\gamma_{k-1} - \gamma_*\|^2\right),$$

where $A(\gamma_*) \equiv \begin{bmatrix} I \\ \nabla_\theta \Upsilon(\theta_*, \gamma_*) \end{bmatrix} \left(\nabla_{\theta\theta'} \widetilde{Q}(\theta_*, \gamma_*)\right)^{-1}$.

Extensions

Adding the following components possible:

1. Inequality constraint on θ
    - e.g., $\theta \geq 0$
2. Acceleration of SLC iteration
    - e.g., Spectral algorithm
        - Sometimes lead to even faster convergence

SLC algorithm in the numerical literature

- Murtagh and Saunders (1982) (Numerical analysis) recognized the SLC algorithm, but rarely used so far in numerical studies
  - Previously no guarantee of fast convergence?
  - In statistical applications (MLE/GMM), $\nabla_Y Q(\theta, Y) \approx 0 \Rightarrow$ Near-quadratic convergence
  - The current study justifies the use of SLC algorithm for statistical applications

Convexity

- $\widetilde{Q}\left(\theta;\gamma_k\right) \equiv Q\left(\theta, \Upsilon\left(\theta;\gamma_k\right)\right)$ is convex in large samples
  - Let $Q_{NFXP}\left(\theta\right)$ be the objective function using NFXP
  - We can show $\nabla_{\theta\theta'}\widetilde{Q}\left(\theta_*;\gamma_*\right) - \nabla_{\theta\theta'}Q_{NFXP}\left(\theta_*\right) = \sum_{i=1}^{n_Y} \frac{\partial Q(\theta_*, Y_*)}{\partial Y_i}\left(\nabla_{\theta\theta'}\Upsilon_i(\theta_*;\gamma_*) - \nabla_{\theta\theta'}\widehat{Y}_i(\theta_*)\right)$
  - $\Rightarrow \nabla_Y Q\left(\theta_*, Y_*\right) = O_p\left(N^{-\frac{1}{2}}\right)$ suggests $\nabla_{\theta\theta'}\widetilde{Q}\left(\theta_*;\gamma_*\right) - \nabla_{\theta\theta'}Q_{NFXP}\left(\theta_*\right) = O_p\left(N^{-\frac{1}{2}}\right)$
  - i.e., $\nabla_{\theta\theta'}Q_{NFXP}\left(\theta_*\right)$ positive definite (p.d.) $\Rightarrow\nabla_{\theta\theta'}\widetilde{Q}\left(\theta_*;\gamma_*\right)$: p.d. in large samples (Convexity of $\widetilde{Q}$ near $\left(\theta_*;\gamma_*\right)$)

Comparison with “MPEC”

- Previous studies proposing the “”MPEC” approach (e.g., Su and Judd, 2012) typically rely on Lagrangian-based solvers (Knitro)
  - e.g., Sequential Quadratic Programming (SQP), Interior Point Method (IP)
  - IP builds on SQP
  - Proposed SLC and SQP are closely related:
    - SLC: Sequentially solve

$$\begin{array}{rl} \min_{\gamma=(\theta,Y)} & Q(\gamma) \\ s.t. & (\nabla_\gamma G(\gamma_k))(\gamma-\gamma_k)+G(\gamma_k)=0 \end{array}$$

    - SQP: Sequentially solve

$$\begin{array}{rc} \min\limits_{\gamma} & (\gamma-\gamma_k)' W_k (\gamma-\gamma_k)+(\nabla_\gamma Q(\gamma_k))(\gamma-\gamma_k) \\ s.t. & (\nabla_\gamma G(\gamma_k))(\gamma-\gamma_k)+G(\gamma_k)=0 \end{array}$$

    where $W_k \equiv \nabla_{\gamma\gamma'}\mathcal{L}(\gamma_k;\lambda_k)$, λ_k: Lagrange multiplier
    - The only difference: Choice of the objective function to minimized
  - Advantage of SLC: SLC can be implemented without explicitly computing $\nabla_\gamma G(\gamma_k)$

Remarks on “MPEC”

- “MPEC” (Lagrangian-based) sometimes works well
- However, its performance largely relies on the sparsity in the case of high-dimensional models...

- Static BLP: “MPEC” is slower than NFXP (with efficient inner-loop)
  - cf. Pál and Sándor (2023)
- Dynamic BLP: “MPEC” is several times slower than NFXP (using numerical derivatives)
  - e.g., 1hour (NFXP) vs 1day (“MPEC”)
  - Explicit computation of Jacobian/Hessian is very costly...
  - Require over 30GB of memory just to store these matrices...
  - cf. Sun and Ishihara (2019)

Relations to Newton's method

- SLC/EPL is closely related to Newton's method for solving nonlinear equations
  - If θ_k is fixed throughout the iteration, coincide with Newton's method:
    - $Y_{k+1} = Y_k - \left(\nabla_Y G\left(Y_k; \theta_k\right)\right)^{-1} \left(G\left(Y_k; \theta_k\right)\right)$
  - Newton's method:
    - Fast (Quadratic) local convergence
    - Sometimes unstable convergence when far from the solution...
  - In the paper, discuss that convergence property of SLC/EPL is different from Newton
    - ⇒ Stable convergence of SLC/EPL is not unusual

Relations with EPL

- Dearing and Blevins (2025) proposed the EPL algorithm using the mapping $\Upsilon(\theta;\gamma_k) \equiv Y_k - (\nabla_Y G(Y_k;\theta_k))^{-1}(G(Y_k;\theta))$
    - Showed good performance by focusing on dynamic games w/o unobs. heterogeneity
    - However, the comp. performance of EPL critically relies on the linearity of $G(Y;\theta)$ wrt θ...
        - In the sequential algorithm, need to solve $\theta_{k+1} = \arg\min_{\theta \in \mathbb{R}^{n_\theta}} \widetilde{Q}(\theta;\gamma_k) \equiv Q(\theta, \Upsilon(\theta;\gamma_k))$
        - Many evaluations of $(\nabla_Y G(Y_k;\theta_k))^{-1}(G(Y_k;\theta))$ given candidate values of θ required...
    - Coincide with SLC under the linearity of $G(Y;\theta)$ wrt θ
        - $\Rightarrow$ EPL works well under the linearity

Comparison of computational costs

Compute Jacobians	Algorithm	Computational cost per iteration	memory size	
Yes	NFXP	$c\,(\text{Solve } G(Y;\theta)=0 \text{ for } Y)+$ $c\left(\nabla_Y G(Y,\theta)\right)+\sum_{i=1}^{n_\theta} c\left(\nabla_Y G(Y,\theta)\right)^{-1}\left(\nabla_{\theta_i} G(Y,\theta)\right)$	$O\left(\max\left(n_G,\,n_Y^2\right)\right)$	
	SLC	$c\left(\text{Solve } \theta_{k+1}=\arg\min Q\left(\theta;\Upsilon\left(\theta;\gamma\right)\right)\right)+$ $c\left(\nabla_Y G(Y,\theta)\right)+\sum_{i=1}^{n_\theta} c\left(\left(\nabla_Y G(Y,\theta)\right)^{-1}\left(\nabla_{\theta_i} G(Y,\theta)\right)\right)$	$O\left(\max\left(n_G,\,n_Y^2\right)\right)$	(N
	SQP	$c\left(\nabla_\gamma G(\gamma)\right)+c\left(\nabla_\gamma Q(\gamma)\right)+c\left(\nabla_{\gamma\gamma'}\mathcal{L}(\gamma)\right)+$ $c\left(\begin{pmatrix}\nabla_{\gamma\gamma'}\mathcal{L}(\gamma,\lambda) & -\left(\nabla_\gamma G(\gamma)\right)' \\ \nabla_\gamma G(\gamma) & 0\end{pmatrix}^{-1}\begin{pmatrix}-\nabla_\gamma Q(\gamma) \\ -G(\gamma)\end{pmatrix}\right)$	$O\left(\max\left(n_G,\,(n_Y+n_\theta)^2\right)\right)$	
No	NFXP	$c\,(\text{Solve } G(Y;\theta)=0 \text{ for } Y)+$ $\sum_{i=1}^{n_\theta} c\left(\text{Solve } G(Y;\theta_i+\epsilon_i,\theta_{-i})=0 \text{ for } Y\right)$	$O\left(n_G\right)$	
	SLC	$c\left(\text{Solve } \theta_{k+1}=\arg\min Q\left(\theta;\Upsilon\left(\theta;\gamma\right)\right)\right)+$ $\sum_{i=1}^{n_\theta} c\left(\text{Solve } \left(\nabla_Y G(Y,\theta)\right)v=\nabla_{\theta_i} G(Y,\theta) \text{ for } v\in\mathbb{R}^{n_Y}\right)$	$O\left(n_G\right)$	(N

Jacobian-free approach in Krylov method

- For a vector b, $\left(\nabla_Y G(Y;\theta)\right)^{-1} b$ is the solution of a linear equation $Ax = b$ wrt x, where $A = \nabla_Y G(Y;\theta)$
  - Krylov method can solve the linear eq. if we can compute Av for any vector v
  - $Av = \left(\nabla_Y G(Y;\theta)\right) v \approx \frac{G(Y+\epsilon v;\theta)-G(Y-\epsilon v;\theta)}{2\epsilon}$ by numerical derivative
  - Motivated by Jacobian-free Newton-Krylov for solving nonlinear equations
  - Krylov method: Packages available in many programming languages (e.g., MATLAB, Julia, Python)
  - cf. Fukasawa (2025): For EPL

Remarks on Automatic Differentiation (AD)

- Sometimes AD is slower than manually coded derivatives...
  - Numerical experiments: Compute the Jacobians of the following wrt V:
    - $\Phi(V) = \log(\exp(u_0 + \beta PV) + \exp(u_1))$
    - Motivated by DDC applications
    - Implemented using ForwardDiff package in Julia
    - Though such drawbacks might be solved by future technological progress on AD, practitioners should remain aware of the current limitations of AD-based implementations

Table: Comparison of computation methods ($|\chi| = 1000$)

		Time (ms)		memory
		Mean	Std	(kB)
Compute Jacobian	Automatic diff.	685.64	103.57	43.54
	Analytical deriv.	8.43	3.35	22.94
Compute JVP	(1) Numerical deriv.	1.07	0.31	0.12
	(2) Automatic diff. (not computing $\frac{\partial \Phi}{\partial V}$)	2.97	0.72	0.13
	(3) Automatic diff. (explicitly computing $\frac{\partial \Phi}{\partial V}$)	372.23	19.01	43.55
	(4) Analytical deriv. (not computing $\frac{\partial \Phi}{\partial V}$)	2.9	1.57	7.71
	(5) Analytical deriv. (explicitly computing $\frac{\partial \Phi}{\partial V}$)	7.03	2.34	22.95

Note. Based on 10,000 trials.

Dynamic BLP: Settings

Model of (perfectly) durable goods (cf. Sun and Ishihara (2019))

- Consumer i's utility when purchasing product j at time t:

$$U_{ijt} = X'_{jt}\theta_i + \xi_{jt} + \epsilon_{ijt}$$

- Consumer i's utility when purchasing nothing:

$$U_{i0t} = \beta E_t[V_{it+1}(\Omega_{t+1})|\Omega_t] + \epsilon_{i0t},$$

where

  - X_{jt}, ξ_{jt}: Observed/Unobserved product char.
  - ϵ: Idiosyncratic utility shock following Gumbel dist.
  - $\theta_i \sim N(\overline{\theta}, \Sigma)$: Random coefficients
  - V_{it}: Consumer i's value function at time t
  - Ω: State variables
  - β: Consumers' discount factor

- Estimate $(\overline{\theta}, \Sigma)$ by GMM
  - Moment conditions: $E[\xi|Z] = 0$, where Z: IVs
  - Suppose $\underbrace{S_{jt}^{(data)}}_{\text{Observed market share}} = \underbrace{s_{jt}\left(\xi, V, \theta\right)}_{\text{Predicted market share}}$

Dynamic BLP: Settings

- $J = 25$, $T = 25$, $I = 50$
- # of grid points: 50; # of simulation draws for computing expectations: 5
- No consumers own the products at time $t = 0$
  - Suppose $\underbrace{S_{jt}^{(data)}}_{\text{Observed market share}} = \underbrace{\sum_i w_i Pr_{it} s_{ijt}^{(ccp)}}_{\text{Market share predicted by the model}}$, where
    - w_i: Consumer i's weight
    - Pr_{it}: Fraction of consumers not owning durables at t
    - $s_{ijt}^{(ccp)}$: Consumer i's CCP of product j at time t
- Expectation formation: Assume Inclusive Value Sufficiency (cf. Gowrisankaran and Rysman (2012))
- Let $\delta_{jt} \equiv X_{jt}\overline{\theta} + \xi_{jt}$ and $\mu_{ijt} \equiv X'_{jt}\left(\theta_i - \overline{\theta}\right)$
  - As the fixed point mapping, use the mapping developed by Fukasawa (2024)

Dynamic BLP fixed-point mapping

- Concerning NFXP, Fukasawa (2024) proposed to solve for δ by

  1. Solve for V by repeatedly applying a novel mapping $\Phi_V(V) \equiv \Psi_{V\delta\to V}\left(V, \iota_{V\to\delta}(V)\right)$ which attains fast convergence
  2. Compute δ by $\delta = \iota_{V\to\delta}(V)$

  - In the case Pr_{it} is known,

$$
\begin{aligned}
\iota_{V\to\delta,jt}(V) &\equiv \log\left(S_{jt}^{(data)}\right) - \log\left(\sum_i w_i Pr_{it} \cdot \frac{\exp\left(\mu_{ijt}\right)}{\exp\left(V_{it}(\Omega_t)\right)}\right) \\
\Psi_{V\delta\to V_{it}}(V) &\equiv \log\left(\sum_j \exp\left(\delta_{jt} + \mu_{ijt}\right) + \exp\left(\beta E_t\left[V_{it+1}(\Omega_{t+1})|\Omega_t\right]\right)\right)
\end{aligned}
$$

- Use Anderson acceleration as the acceleration method

- Traditional approach: Needed to solve two types of variables... (V and δ)

  $\Leftrightarrow$ Proposed (Only V)

Performance of the new dynamic BLP mapping

- Results of Monte Carlo simulation (cf. Fukasawa (2024)):

	Func. Evals.						Mean
	Mean	Min.	25th	Median.	75th	Max.	CPU time (s)
New (+ Anderson)	238.1	192	214.5	239.5	258	296	0.93795
Traditional	2482.75	2036	2274	2417	2667.5	3000	9.0842

2. Dynamic discrete game: Settings

- Dynamic entry/exit (incomplete-information) game
    - cf. Aguirregabiria and Mira (2007), Dearing and Blevins (2025)
    - If enter the market, receive profit, but incur entry cost.
    - The profit depends on state variables z_t (number of competitors, observed market size, unobserved market states s_t).
- We assume s_t is unobserved (Time-varying unobserved heterogeneity):
    - $s_t \in \mathcal{S}$, where $\mathcal{S}$ is known, but the values of s_t and $Pr\left(s_{t+1}|s_t\right)$ are unknown
    - Estimate parameters by MLE
    - cf., Arcidiacono and Miller (2011), Hu and Shum (2012)
    - Ignoring unobserved heterogeneity may lead to biased implications

Dynamic game: Settings

- Let
  - x_t, s_t: Observed/Unobserved market state
  - $a_t^j \in \{0, 1\}$: Action of firm j (enter or exit the market)
- Utility:

$$\overline{u}^j(x_t, s_t, a_t^j, a_t^{-j}; \theta_u) \quad = \quad \begin{cases} \theta_{FC,j} + \theta_{RS}^{obs} x_t + \theta_{RS}^{unobs} s_t - \theta_{RN} \ln \left(1 + \sum_{l \neq j} a_t^l\right) - \theta_{EC} \left(1 - a_{t-1}^j\right) & \text{if } a_t^j = 1, \\ 0 & \text{if } a_t^j = 0. \end{cases}$$

- $x_t \in \{1, \cdots, 5\}$,$s_t \in \{1, \cdots, 3\}$, $\theta_{RS}^{obs} = 1, \theta_{RS}^{unobs} = 1$, $\theta_{RN} = 2.5$ $J = 3$.
- Prob. of the s_t remaining unchanged in successive periods is $\pi = 0.8$.
- Initial distribution $p^{steady} \equiv Pr\left(x_{t=1}, s_{t=1}\right)$ is also unobserved.
- Estimate parameters θ_u and π by MLE.
- Other than x_t and s_t, mostly the same as Aguirregabiria and Mira (2007) and Dearing and Blevins (2025).
- We can construct a fixed-point mapping $\left(v, p^{steady}, \pi\right) = \Phi\left(v, p^{steady}, \pi\right)$, where v denotes the choice-specific value functions.
  - Remark: Bray (2019)'s relative value iteration approach did not lead to large speed up when combining with the Anderson acceleration.