Quantitative Economics 12 (2021), 1223–1271 1759-7331/20211223

# Imposing equilibrium restrictions in the estimation of dynamic discrete games

Victor Aguirregabiria
Department of Economics, University of Toronto and CEPR

Mathieu Marcoux
Département de sciences économiques, Université de Montréal, CIREQ, and CIRANO

Imposing equilibrium restrictions provides substantial gains in the estimation of dynamic discrete games. Estimation algorithms imposing these restrictions have different merits and limitations. Algorithms that guarantee local convergence typically require the approximation of high-dimensional Jacobians. Alternatively, the Nested Pseudo-Likelihood (NPL) algorithm is a fixed-point iterative procedure, which avoids the computation of these matrices, but—in games—may fail to converge to the consistent NPL estimator. In order to better capture the effect of iterating the NPL algorithm in finite samples, we study the asymptotic properties of this algorithm for data generating processes that are in a neighborhood of the NPL fixed-point stability threshold. We find that there are always samples for which the algorithm fails to converge, and this introduces a selection bias. We also propose a spectral algorithm to compute the NPL estimator. This algorithm satisfies local convergence and avoids the approximation of Jacobian matrices. We present simulation evidence and an empirical application illustrating our theoretical results and the good properties of the spectral algorithm.

Keywords. Dynamic discrete games, nested pseudo-likelihood, fixed-point algorithms, spectral algorithms, convergence, convergence selection bias.

JEL classification. C13, C57, C61, C73.

## 1. Introduction

Economic theory often delivers models in which the distribution of agents' decisions is implicitly defined as a fixed point. For instance, this is the case in models where agents'

Victor Aguirregabiria: victor.aguirregabiria@utoronto.ca
Mathieu Marcoux: mathieu.marcoux@umontreal.ca

We thank comments from three anonymous referees. We have also benefited from insightful discussions with Jason Blevins, Martin Burda, Adam Dearing, Limin Fang, Christian Gouriéroux, Ashique Habib, Fedor Iskhakov, Min Seong Kim, Adam Lavecchia, Patrick Mogensen, Ismael Mourifié, Benoit Perron, Thomas Russell, Eduardo Souza-Rodrigues, Bertel Schjerning, and Yuanyuan Wan, and from seminar participants at Stockholm School of Economics, CREST, McGill, Ohio State, Rice, Toronto, the 2020 World Congress of the Econometric Society, the Empirical Microeconomics Workshop (Banff), the CEA Annual Meetings (Vancouver), the CIREQ PhD Students Conference (Montreal), and the Doctoral Workshop in Applied Economics (Toronto). Financial support from Social Sciences and Humanities Research Council (SSHRC) is gratefully acknowledged.

 https://doi.org/10.3982/QE1735

decisions are interpreted via equilibrium conditions based on rational expectations. In this class of models, there are typically two different objects of interest: The conditional choice probabilities (CCPs), which are a probabilistic representation of agents' behavior, and the structural parameters of the model. In order to reduce the computational burden of estimating such models, it is common practice to first estimate the CCPs in a flexible way and then to estimate the structural parameters conditional on these estimates. While consistent in large samples, this two-step procedure is known to suffer from a few drawbacks in finite samples. In particular, imprecise estimates in the first step may affect the properties of the second-step estimator. Furthermore, consistent first-step estimates may not be available, as it is the case if choice probabilities depend on unobserved heterogeneity with continuous support.

These drawbacks motivated the Nested Pseudo-Likelihood (NPL) estimator proposed by Aguirregabiria and Mira (2002, 2007): a fixed point of the so-called NPL mapping, which embeds both the pseudo-maximum likelihood estimation of the structural parameters and the equilibrium restrictions for the CCPs. As a method to obtain the NPL estimator, Aguirregabiria and Mira (2002, 2007) proposed the *NPL fixed-point algorithm*, which iteratively updates the estimated CCPs using the NPL mapping.[1] As a fixed-point iterative method, this algorithm can only converge to fixed points satisfying some stability conditions. The algorithm is expected to fail to converge to the consistent NPL estimator when the data generating process corresponds to an unstable fixed point (Pesendorfer and Schmidt-Dengler (2008, 2010)). A perhaps more concerning issue is that Monte Carlo studies have recently shown that the NPL algorithm may fail to converge even when the data generating process does satisfy stability conditions (Egesdal, Lai, and Su (2015)). Unfortunately, the literature has so far failed to provide a convincing explanation in such cases.

In this paper, we take another look at the sequential estimation of dynamic discrete games in order to bridge the literature's incomplete understanding of the NPL algorithm's failure to converge. We depart from existing works in two ways.

First, we treat the NPL algorithm as being iterated in finite samples rather than in large samples. While the latter approach is commonly used in the literature, the former interpretation is better aligned with the motivation of imposing equilibrium restrictions to improve upon the finite sample performance of two-step estimation. This important distinction allows us to show that the convergence of the NPL algorithm is driven by the stability of the NPL mapping in the sample, as opposed to the NPL mapping in the population. As a result, the NPL algorithm may fail to converge even if the data generating process corresponds to a stable fixed point. The NPL algorithm is likely to fail to converge in finite samples if the data generating process is stable, but close to being unstable. To appreciate this property of the NPL algorithm, the Monte Carlo simulations included in this paper are repeated for a grid of data generating processes that satisfy stability conditions to different degrees. We find that Aguirregabiria and Mira's (2007) data generating process—commonly used as a benchmark for Monte Carlo experiments—is in fact close to being unstable.

[1] Throughout the paper, we use the terms *NPL algorithm* and *NPL fixed-point algorithm* as synonymous.

Second, we present a new analysis of the implications of the NPL algorithm's possible failure to converge on the asymptotic properties of the estimator obtained upon convergence. To do so, we study the asymptotic properties of this estimator for sequences of stable data generating processes whose limit corresponds to a data generating process exactly on the stability threshold. This local asymptotic analysis, that is, for data generating processes close to the stability threshold, leads to nontrivial probabilities of convergence for the fixed-point algorithm, even in large samples. Since convergence of the fixed-point iterations is a stochastic condition, the possibility of no convergence generates a selection problem, which we refer to as *convergence selection*. Random samples for which the NPL algorithm converges have statistical properties that are different from samples where convergence fails. We characterize the properties of this selection in terms of the empirical distribution of the spectral radius of the sample fixed-point mapping. In our Monte Carlo experiments, this convergence selection problem affects especially the estimation of the structural parameter that represents competition effects: It introduces an attenuation bias in this parameter.

An additional contribution of this paper is that we propose an alternative iterative algorithm to compute the NPL estimator. Instead of fixed-point iterations, the proposed algorithm uses a spectral approach to solve for the fixed points of the NPL mapping. Similarly, as for Jacobian-based algorithms that guarantee local convergence, fixed-point instability does not prevent the spectral algorithm to converge. However, just like the NPL fixed-point algorithm, the spectral algorithm avoids the approximation of high-dimensional Jacobians. Our simulation evidence suggests that this alternative algorithm performs very well, even if the data generating process admits multiple equilibria. It is also worth mentioning that its computational cost is often cheaper than the NPL fixed-point algorithm when this procedure converges. The good properties of the spectral approach are also confirmed in an empirical application to a dynamic game of fast food restaurant location featuring a relatively large dimensional vector of equilibrium CCPs (counting about 190,000 elements).

The rest of the paper is organized as follows. Section 2 reviews the related literature. The model of interest, a discrete choice dynamic game with incomplete information, is presented in Section 3. The NPL estimator and different sequential estimators, including our proposed spectral algorithm, are described in Section 4. Section 5 presents our main theoretical results. We provide necessary and sufficient conditions for the convergence of the NPL fixed-point algorithm, and use these conditions to characterize the statistical properties of the estimator upon convergence. We also describe the selection bias due to convergence. In Section 6, we present evidence that the selection problem generated by the lack of convergence is the most important source of finite sample bias when the data generating process is stable. Section 7 summarizes the empirical application and proposes a step-by-step guide for the implementation of the algorithms. Section 8 concludes. Codes in R for all our Monte Carlo experiments for different versions of the data generating processes in Aguirregabiria and Mira (2007) and Pesendorfer and Schmidt-Dengler (2008) as well as the material to replicate our empirical application are available here.

## 2. Related literature

A natural starting point to estimate dynamic games would be to extend a Nested Fixed-Point (NFXP) algorithm similar to Rust (1987) to a game setting to obtain a maximum likelihood estimator (MLE) of the structural parameters of interest. Multiplicity of equilibria in games however complicates the implementation of the NFXP algorithm compared to its typical application to single-agent dynamic discrete choice problems. An important reduction in the computational burden can be achieved by using a two-step approach in which reduced-form CCPs are estimated in the first step and then taken as given when estimating the structural parameters in the second step. This approach corresponds to an application of Hotz and Miller's (1993) CCP estimator to the dynamic games framework and was initially proposed by Aguirregabiria and Mira (2007).[2] Different two-step estimators have been proposed in the context of dynamic games: Bajari, Benkard, and Levin's (2007) simulated minimum distance estimator; Pakes, Ostrovsky, and Berry (2007) advocating for the use of method of moments estimators in the second step; and Pesendorfer and Schmidt-Dengler's (2008) asymptotic least squares estimator.

Despite being computationally convenient and having nice asymptotic properties, two-step approaches tend to perform poorly in small samples, especially when there is a large number of states in the model. It is also worth noting that—since they are reduced-form estimates—the first-step CCPs do not satisfy the fixed-point restrictions *in finite samples*. Aguirregabiria and Mira (2007) aimed at addressing these issues by proposing the NPL estimator and an iterative algorithm to compute this estimator.

The NPL estimator is defined as a vector of structural parameters and a vector of CCPs that satisfy two conditions: The vector of structural parameters maximizes a pseudo-likelihood function that depends on the CCPs; and the vector of CCPs satisfies the model equilibrium restrictions associated with the vector of structural parameters. These two conditions lead to a fixed-point representation: the vector of CCPs is a fixed point of the so-called NPL mapping. If the NPL mapping has multiple fixed points, then the NPL estimator is the solution with the maximum value of the pseudo-likelihood function.

To compute the NPL estimator, Aguirregabiria and Mira (2007) proposed a fixed-point iterative procedure, denoted as the *NPL (fixed-point) algorithm*. Given a vector of CCPs as input, a single iteration of the NPL algorithm provides a new vector of CCPs that is equal to players' best response to the input CCPs when the vector of structural parameters is the one that maximizes the pseudo-likelihood function for the input CCPs. Upon convergence, the NPL algorithm delivers a fixed point of the NPL mapping.

Aguirregabiria and Mira (2007) did not formally show the convergence of the NPL fixed-point algorithm. In a nutshell, the limitations of this algorithm can be summarized in two important criticisms. First, if the data generating process corresponds to an unstable fixed point of the NPL mapping, then none of the NPL fixed points found

[2] An assumption commonly maintained when implementing this CCP estimator is that, while the game theoretical model may admit multiple equilibria, the same equilibrium is realized whenever the same game is played.

by the NPL algorithm initiated at different values of the CCPs correspond to the NPL estimator in large samples. Then, if it converges, the NPL algorithm leads to an inconsistent estimator (Pesendorfer and Schmidt-Dengler (2008, 2010)). Second, even if the data generating process satisfies desirable stability conditions, the NPL algorithm may still fail to converge (Egesdal, Lai, and Su (2015)).[3]

It is important to highlight the fundamental difference between the NPL (fixed-point) algorithm and the NPL estimator. The NPL estimator, as defined by Aguirregabiria and Mira (2007, p. 20), is "the NPL fixed point in the sample with the maximum value of the pseudo-likelihood." The nice asymptotic properties of the NPL estimator (Aguirregabiria and Mira (2007, Proposition 2)) do not depend on stability conditions required for the NPL algorithm to converge. In other words, the NPL estimator exists regardless whether the NPL algorithm converges or not.

At this point, it is worth expanding on the two different ways of investigating the asymptotic properties of the sequences of estimates generated by the NPL algorithm, which were mentioned in the Introduction. The first one—which is the one commonly used in the literature (e.g., Kasahara and Shimotsu (2012), Bugni and Bunting (2021))—consists in considering an asymptotic approximation to the NPL algorithm, as if the iterations were performed in a large sample. Under this interpretation, the convergence of the NPL algorithm is determined by the stability of the equilibrium generating the data since one is effectively updating the choice probabilities using the NPL mapping in the population. A second interpretation— motivated by Pesendorfer and Schmidt-Dengler's (2010) criticism—rather iterates the sequential estimator in finite samples and then considers the sample size to grow to infinity. This is the interpretation that is considered in the current paper. Furthermore, to highlight how nonconvergence of the NPL algorithm may be more problematic around the stability threshold even in large samples, we study asymptotic properties of the estimator obtained upon convergence of the NPL algorithm for sequences of stable data generating processes whose limit is a data generating process exactly satisfying the stability threshold. To the best of our knowledge, we are the first to investigate such local asymptotic properties for the estimator defined by the NPL algorithm.

Some alternative algorithms or/and estimators have been proposed in hope of addressing the convergence issues associated with the NPL fixed-point algorithm. One of them is the *NPL relaxation method* from Kasahara and Shimotsu (2012). This iterative algorithm uses a relaxation method that perturbs the original NPL mapping according to a fixed parameter $\alpha$. Kasahara and Shimotsu (2012) provided simulation evidence that this relaxation method works well when the parameter $\alpha$ is set optimally. However, such an optimal value is not available in practice since it requires the knowledge of the NPL mapping in the population, which is unknown to the econometrician. Kasahara and Shimotsu (2012) therefore recommended using a small arbitrary value for this parameter.

[3]Mogensen (2015) provided simulation evidence that nonconverging sequences typically coincide with parameters' estimates for which the stability conditions required for local contraction do not hold. This observation is, to the best of our knowledge, the first attempt at understanding the NPL algorithm's possible failure to converge when desirable stability conditions hold.

Unfortunately, Egesdal, Lai, and Su (2015) have found the properties of this NPL relaxation algorithm to be much less appealing when using such a small arbitrary value.

Another alternative approach is a class of Mathematical Program with Equilibrium Constraints (MPEC) estimators which was proposed by Su and Judd (2012) for single-agent dynamic discrete choice models, and extended to dynamic games by Egesdal, Lai, and Su (2015). This approach consists in computing the MLE by maximizing the (pseudo) likelihood function with respect to the structural parameters and the vector of CCPs subject to the equilibrium constraints using state of the art Jacobian-based optimization algorithms with good local convergence properties. On one hand, MPEC has the advantage of delivering the MLE. On the other hand, it now requires optimizing over a large vector of parameters, which includes the structural parameters, the CCPs, and the Lagrange multipliers of all the equilibrium constraints. Furthermore, in contrast to the NPL algorithm, MPEC requires the computation of Jacobian matrices with respect to the vector of CCPs such that the complexity of this method—relative to NPL—increases fast with the state space.[4] Egesdal, Lai, and Su (2015) provided simulation evidence suggesting better convergence properties than the estimator obtained using the NPL fixed-point algorithm.[5]

Dearing and Blevins (2021) have recently proposed the Efficient Pseudo-Likelihood (EPL) algorithm as an alternative to NPL. This algorithm is similar to NPL, but it iterates in the space of conditional choice values, instead of the space of CCPs. The authors show that their algorithm converges in situations where NPL fixed-point iterations do not and is asymptotically equivalent to the MLE. However, since the iterative procedure used to obtain the EPL is based on a Newton step, it also requires approximating the inverse of a potentially large dimensional Jacobian.

While alternatives such as Kasahara and Shimotsu's (2012) modified NPL algorithm, Egesdal, Lai, and Su (2015)'s MPEC approach and Dearing and Blevins's (2021) estimator clearly have merit, there is still room for a better understanding of the NPL algorithm's properties. In applications with large state spaces, the NPL algorithm is arguably much easier to implement and has a substantially smaller computational cost. An important goal of the current paper is to improve our understanding of the NPL algorithm's failure to converge in hopes to identify in which settings this method can be safely used in empirical applications.[6]

[4]To deal with this curse of dimensionality, MPEC can leverage on the potential sparsity of the Jacobian matrices involved in the maximization problem. However, in many applications of dynamic games the Jacobian of the equilibrium mapping with respect to CCPs is not sparse. For instance, in a dynamic game of market entry, a player's best response probability at any state depends on other players' CCPs at any state.

[5]For single-agent dynamic discrete choice model, Iskhakov, Lee, Rustm Schjerning, and Seo (2016) provided simulation evidence suggesting that the NFXP algorithm may actually have better convergence properties than the MPEC estimator.

[6]It is worth noting that the NPL algorithm has been used in several empirical applications of single-agent dynamic models (Copeland and Monnet (2009), De Pinto and Nelson (2009), Tomlin (2014), Aguirregabiria and Alonso-Borrego (2014)), dynamic games (Sweeting (2013), Aguirregabiria and Ho (2012), Collard-Wexler (2013), Kano (2013), Huang and Smith (2014), Lin (2015), Gayle and Xie (2018)), static games (Ellickson and Misra (2008), Han and Hong (2011)), and networks (Lin and Xu (2017), Liu and Zhou (2017)).

## 3. The model

### 3.1 *General framework*

Let $\mathcal{Y}$ and $\mathcal{X}$ be two discrete and finite sets and let $P^0(\mathbf{y}|\mathbf{x})$ be the true probability distribution of the realization of a $N \times 1$ vector $\mathbf{y} \in \mathcal{Y}^N$, for $N \geq 1$, conditional on a vector of variables $\mathbf{x} \in \mathcal{X}$. Also, denote $\mathbf{P}^0 = \{P^0(\mathbf{y}|\mathbf{x}) : \mathbf{y} \in \mathcal{Y}^N, \mathbf{x} \in \mathcal{X}\} \in \mathcal{P}$. There is a parametric structural model for this vector of conditional probabilities. Consider a finite-dimensional vector of structural parameters $\boldsymbol{\theta} \in \Theta$ and denote the vector of parametrized probabilities as $\mathbf{P}(\boldsymbol{\theta}) = \{P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) : \mathbf{y} \in \mathcal{Y}^N, \mathbf{x} \in \mathcal{X}\}$. The true value of the structural parameters in the population, denoted $\boldsymbol{\theta}^0$, satisfies $\mathbf{P}(\boldsymbol{\theta}^0) = \mathbf{P}^0$.

There is no closed-form analytical expression for $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. The parametric distribution of $\mathbf{y}$ is implicitly defined as a fixed point of the following mapping in the probability space:

$$\mathbf{P}(\boldsymbol{\theta}) = \boldsymbol{\Psi}\big(\boldsymbol{\theta}, \mathbf{P}(\boldsymbol{\theta})\big), \tag{1}$$

where $\boldsymbol{\Psi}(\cdot) : \Theta \times \mathcal{P} \to \mathcal{P}$ and $\boldsymbol{\Psi}(\boldsymbol{\theta}, \mathbf{P}) = \{\Psi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{P}) : \mathbf{y} \in \mathcal{Y}^N, \mathbf{x} \in \mathcal{X}\}$ is a vector with elements organized in the same order as in $\mathbf{P}$.

Assumption 1 (Regularity Conditions). (A) *The mapping $\boldsymbol{\Psi}(\boldsymbol{\theta}, \mathbf{P})$ is twice continuously differentiable.* (B) $\boldsymbol{\theta}^0 \in \text{int}(\Theta)$. (C) *$\Theta$ is compact.* (D) *There is an open ball around $\mathbf{P}^0$ that does not include any other fixed point of the mapping $\boldsymbol{\Psi}(\boldsymbol{\theta}^0, \cdot)$.*

In empirical applications of dynamic games, the researcher typically has panel data from $M$ games (e.g., markets, locations) indexed by $m \in \{1, 2, \ldots, M\}$, $N$ players indexed by $i \in \{1, 2, \ldots, N\}$, and $T$ time periods indexed by $t \in \{1, 2, \ldots, T\}$. For each observation $(m, t)$, the data set includes the realizations of players' actions, $\mathbf{y}_{mt} = (y_{1mt}, y_{2mt}, \ldots, y_{Nmt}) \in \mathcal{Y}^N$, and the vector of observable state variables $\mathbf{x}_{mt} \in \mathcal{X}$. Typically, the asymptotic results depend on either the number of markets $M$ or the number of time periods $T$. For what follows, we consider asymptotic results as $M \to \infty$.

Assumption 2 summarizes the conditions on the data generating process typically supposed to be satisfied in the literature. In essence, these restrictions guarantee that the sample consists of independent draws generated from a single (given observed states) equilibrium.

Assumption 2 (Data Generating Process). (A) $P^0(\mathbf{y}|\mathbf{x}) = \Psi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}^0, \mathbf{P}^0), \forall (\mathbf{y}, \mathbf{x}) \in \mathcal{Y}^N \times \mathcal{X}$. (B) *For any $\mathbf{P}$ that solves $\mathbf{P} = \boldsymbol{\Psi}(\boldsymbol{\theta}, \mathbf{P})$, $\mathbf{P} \neq \mathbf{P}^0$ whenever $\boldsymbol{\theta} \neq \boldsymbol{\theta}^0$.* (C) *Observations in the data set $\{\mathbf{y}_{mt}, \mathbf{x}_{mt} : m = 1, \ldots, M; t = 1, \ldots, T\}$ are independent over $m$ and $\Pr(\mathbf{x}_{mt} = \mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}$.*

Throughout the paper, we use a hat superindex to indicate that an object is a statistic or a stochastic function that has sampling variation. Objects without a hat superindex are deterministic functions or parameters.

### 3.2 *Example: Dynamic game of market entry and exit*

In this section, we introduce one of the dynamic games of market entry and exit that will be used for the Monte Carlo experiments in Section 6. This model comes from Aguirregabiria and Mira (2007) and it is a particular case of the general framework presented above. It has been used in several simulation exercises, namely Pesendorfer and Schmidt-Dengler (2008), Kasahara and Shimotsu (2012), and Egesdal, Lai, and Su (2015). Our Monte Carlo simulations also include the simpler game of market entry and exit proposed by Pesendorfer and Schmidt-Dengler (2008).

Every period $t$, each firm $i$ decides whether or not to operate in a market $m$. In this context, $\mathcal{Y} = \{0, 1\}$ is each firm's choice set, such that $y_{imt} = 1$ if firm $i$ operates in market $m$ at time $t$. Let $\mathbf{y}_{mt}$ be the vector $(y_{imt} : i = 1, \ldots, N)$. A firm maximizes its expected and discounted stream of current and future profits in the market: $\mathrm{E}_t[\sum_{s=0}^{\infty} \beta^s U_{im,t+s}]$, where $\beta \in (0, 1)$ is the discount factor, and $U_{imt}$ is the period profit.

Suppose that a firm's decision to operate depends on the market size ($s_{mt}$), the number of competitors that are operating ($\sum_{j \neq i} y_{jmt}$), an entry cost, a firm-specific fixed cost, and the vector of variables $\boldsymbol{\varepsilon}_{imt} = [\varepsilon_{imt}(0), \varepsilon_{imt}(1)]'$, which are player $i$'s private information. Let $\boldsymbol{\varepsilon}_{mt}$ be the vector $(\boldsymbol{\varepsilon}_{imt} : i = 1, \ldots, N)$. The dynamic dimension to the firm's decision comes from the firm paying the entry cost only if it has not been operating in the market during the previous period.

An observation is a firm-market-period-tuple. Besides the decision of being active in the market, the econometrician observes the market size ($s_{mt}$) and the incumbency status of the firms ($\mathbf{y}_{m,t-1}$). To be consistent with the notation introduced above, denote the observable state variables in market $m$ at period $t$ as $\mathbf{x}_{mt} = [s_{mt}, \mathbf{y}'_{m,t-1}]'$.

Following Rust (1987), the literature of dynamic discrete choice structural models—and more specifically, dynamic discrete games—has used the assumptions of *additive separability* and *conditional independence* of the unobservable state variables $\boldsymbol{\varepsilon}_{imt}$. These two assumptions —together with the condition that the space of the observable state variables, $\mathcal{X}$, is discrete —imply that this class of dynamic models can be represented using the general framework presented in the previous section.

*Additive separability* The contemporaneous payoff function is additively separable in its observable and unobservable components: $U_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{imt}) = u_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}) + \boldsymbol{\varepsilon}_{imt}(y_{imt})$.

*Conditional independence* The transition probability $\Pr(\mathbf{x}_{m,t+1}, \boldsymbol{\varepsilon}_{im,t+1} | \mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{imt}, \mathbf{y}_{mt})$ factors as $\Gamma(\mathbf{x}_{m,t+1} | \mathbf{x}_{mt}, \mathbf{y}_{mt}) G(\boldsymbol{\varepsilon}_{im,t+1})$, where $G(\cdot)$ is the cumulative distribution function of $\boldsymbol{\varepsilon}_{imt}$, which is i.i.d. across players, markets, and time periods.

For instance, Aguirregabiria and Mira (2007) considered the following specification of the payoff function $U_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{imt})$:

$$\begin{cases} \theta_{\mathrm{RS}} s_{mt} - \theta_{\mathrm{RN}} \ln\left[1 + \sum_{j \neq i} y_{jmt}\right] - \theta_{\mathrm{EC}}(1 - y_{im,t-1}) - \theta_{\mathrm{FC},i} + \varepsilon_{imt}(1), & y_{imt} = 1, \\ \varepsilon_{imt}(0), & y_{imt} = 0, \end{cases} \tag{2}$$

where $\theta_{RS}$ measures the effect of the market size on the firm's payoff; $\theta_{RN}$ is referred to as the strategic interaction parameter; $\theta_{EC}$ corresponds to the entry cost; and $\theta_{FC,i}$ is the firm-specific fixed cost.

We assume that players' behavior can be rationalized via a Markov perfect equilibrium, which implies that player $i$'s strategies only depend on this player's payoff-relevant state variables, more precisely $\mathbf{x}_{mt}$ and $\boldsymbol{\varepsilon}_{imt}$. At a given state—defined by $\mathbf{x}_{mt}$ and $\boldsymbol{\varepsilon}_{imt}$—let $\sigma_i(\cdot) : \mathcal{X} \times \mathbb{R}^2 \mapsto \mathcal{Y}$ be player $i$'s strategy function. Given the other firms' strategies—represented as $\boldsymbol{\sigma}_{-i,mt} \equiv \boldsymbol{\sigma}_{-i}(\mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{-i,mt})$—firm $i$'s best response is the solution of a dynamic programming problem with the following value function and Bellman equation:

$$
\begin{aligned}
&V_i(\mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{imt}) \\
&\quad = \max_{y \in \{0,1\}} \Bigg\{ \mathrm{E}_{\boldsymbol{\varepsilon}_{-i,mt}} \Bigg[ U_i\big(y, \boldsymbol{\sigma}_{-i,mt}, \mathbf{x}_{mt}, \varepsilon_{imt}(y)\big) \\
&\qquad + \beta \int \sum_{\mathbf{x}_{m,t+1} \in \mathcal{X}} V_i(\mathbf{x}_{m,t+1}, \boldsymbol{\varepsilon}_{im,t+1}) \Gamma\big(\mathbf{x}_{m,t+1} | \mathbf{x}_{mt}, y, \boldsymbol{\sigma}_{-i,mt}\big)\, dG(\boldsymbol{\varepsilon}_{im,t+1}) \Bigg] \Bigg\}. \qquad (3)
\end{aligned}
$$

In equilibrium, players' strategies are best responses to each other and—taking the strategies of the other players as given—a player's best response is the solution of a single-agent dynamic programming problem. The set of players' strategies also determine their CCPs such that:

$$
P\big(y_i | \mathbf{x}_{mt}, \boldsymbol{\theta}\big) = \int \mathbb{1}\big\{y_i = \sigma_i(\mathbf{x}_{mt}, \boldsymbol{\varepsilon}_{imt})\big\}\, dG(\boldsymbol{\varepsilon}_{imt}). \qquad (4)
$$

Aguirregabiria and Mira (2007, representation lemma) show that, in this class of dynamic games, a Markov perfect equilibrium can be represented as a fixed point of a mapping in the space of the vector of CCPs for all players and observable states.

## 4. NPL estimator and sequential estimators

In this section, we introduce the NPL mapping and use it to define the (algorithm-free) NPL estimator. We also present a series of sequential estimators including the Hotz–Miller estimator, the $K$-step estimator, the NPL algorithm, the relaxation algorithm, and a new algorithm to compute the NPL estimator.

### 4.1 *NPL mapping and estimator*

Let $\hat{Q}(\boldsymbol{\theta}, \mathbf{P}) \equiv M^{-1} \sum_{m=1}^{M} \ln[\Psi(\mathbf{y}_m | \mathbf{x}_m, \boldsymbol{\theta}, \mathbf{P})]$ be the pseudo log-likelihood function for any $\boldsymbol{\theta} \in \Theta$ and any $\mathbf{P} \in \mathcal{P}$. Let $Q^0(\boldsymbol{\theta}, \mathbf{P}) \equiv \mathrm{E}_{\mathbf{P}^0}[\ln[\Psi(\mathbf{y}_m | \mathbf{x}_m, \boldsymbol{\theta}, \mathbf{P})]]$ be its population counterpart, where $\mathrm{E}_{\mathbf{P}^0}[\cdot]$ is computed with respect to the true probability distribution $\mathbf{P}^0$. By maximizing these functions over $\Theta$ for a given $\mathbf{P} \in \mathcal{P}$, one obtains the pseudo-maximum likelihood estimator $\hat{\boldsymbol{\vartheta}}(\mathbf{P}) = \arg\max_{\boldsymbol{\theta} \in \Theta} \hat{Q}(\boldsymbol{\theta}, \mathbf{P})$ and its population counterpart $\boldsymbol{\vartheta}^0(\mathbf{P}) = \arg\max_{\boldsymbol{\theta} \in \Theta} Q^0(\boldsymbol{\theta}, \mathbf{P})$.

By maximizing the pseudo log-likelihood function evaluated at a consistent reduced-form estimate of the CCPs, say $\hat{\mathbf{P}}_0$, one obtains the two-step pseudo MLE or Hotz–Miller estimator applied to dynamic discrete games. This estimator is formally defined in Definition 1.

DEFINITION 1 (Two-Step Pseudo MLE). For a given consistent reduced-form estimate of the CCPs $\hat{\mathbf{P}}_0$, the two-step pseudo MLE estimator $\hat{\boldsymbol{\theta}}_1$ satisfies

$$\hat{\boldsymbol{\theta}}_1 \equiv \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_0) = \arg\max_{\boldsymbol{\theta}\in\Theta} \hat{Q}(\boldsymbol{\theta}, \hat{\mathbf{P}}_0). \tag{5}$$

The vectors $\hat{\boldsymbol{\vartheta}}(\mathbf{P})$ and $\boldsymbol{\vartheta}^0(\mathbf{P})$ are also used to define the sample and the population NPL mappings. As it is formally stated in Definition 2, for any $\mathbf{P} \in \mathcal{P}$, the NPL mapping is a function that returns the vector of best response CCPs at the corresponding pseudo-maximum likelihood estimator of $\boldsymbol{\theta}$. The only difference between the population NPL mapping and its sample counterpart is whether the pseudo-maximum likelihood estimator of $\boldsymbol{\theta}$ is computed in the population or in the sample.

DEFINITION 2 (NPL Mapping). For any $\mathbf{P} \in \mathcal{P}$, the sample NPL mapping is $\hat{\boldsymbol{\varphi}}(\mathbf{P}) \equiv \boldsymbol{\Psi}(\hat{\boldsymbol{\vartheta}}(\mathbf{P}), \mathbf{P})$; its population counterpart is $\boldsymbol{\varphi}^0(\mathbf{P}) \equiv \boldsymbol{\Psi}(\boldsymbol{\vartheta}^0(\mathbf{P}), \mathbf{P})$.

Notice that these NPL mappings are continuous functions on a compact set, that is, from $\mathcal{P}$ to itself. By Brouwer's fixed-point theorem, a fixed point of the NPL mapping always exists, both for the sample and the population mappings. In other words, the vector mappings $\hat{\boldsymbol{\phi}}(\mathbf{P}) \equiv \mathbf{P} - \hat{\boldsymbol{\varphi}}(\mathbf{P})$ and $\boldsymbol{\phi}^0(\mathbf{P}) \equiv \mathbf{P} - \boldsymbol{\varphi}^0(\mathbf{P})$ both have at least one root. However, uniqueness is not guaranteed. Definition 3 formally states the conditions satisfied by a fixed point of the sample NPL mapping.

DEFINITION 3 (Fixed Point of the Sample NPL Mapping). $\hat{\mathbf{P}}_*$ is a fixed point of the sample NPL mapping if $\hat{\mathbf{P}}_* = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_*)$. This implies that for $\hat{\boldsymbol{\theta}}_* \equiv \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_*)$ the following conditions hold: (A) $\hat{\boldsymbol{\theta}}_* = \arg\max_{\boldsymbol{\theta}\in\Theta} \hat{Q}(\boldsymbol{\theta}, \hat{\mathbf{P}}_*)$; and (B) $\hat{\mathbf{P}}_* = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_*, \hat{\mathbf{P}}_*)$.

The NPL estimator is formally stated in Definition 4.

DEFINITION 4 (NPL Estimator). The NPL estimator $(\hat{\boldsymbol{\theta}}_{\text{NPL}}, \hat{\mathbf{P}}_{\text{NPL}})$ is defined as the fixed point of the sample NPL mapping achieving the largest value of the pseudo-likelihood function. That is, it satisfies the following conditions: (A) $\hat{\mathbf{P}}_{\text{NPL}} = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_{\text{NPL}})$; (B) $\hat{\boldsymbol{\theta}}_{\text{NPL}} = \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_{\text{NPL}})$; and (C) $(\hat{\boldsymbol{\theta}}_{\text{NPL}}, \hat{\mathbf{P}}_{\text{NPL}})$ maximizes $\hat{Q}(\boldsymbol{\theta}, \mathbf{P})$ within the set of fixed points of the sample NPL mapping.

Aguirregabiria and Mira (2007, Proposition 2) showed that the NPL estimator is consistent and asymptotically normal. Furthermore, if the spectral radius, that is, the largest eigenvalue in absolute value of the Jacobian matrix $\partial\boldsymbol{\varphi}^0(\mathbf{P})/\partial\mathbf{P}$ evaluated at $\mathbf{P}^0$ is smaller than one, it is asymptotically more efficient than the two-step estimator.

It is worth comparing the properties of the NPL estimator with those of the MLE that jointly maximizes the likelihood function with respect to $\mathbf{P}$ and $\boldsymbol{\theta}$ subject to the constraint $\mathbf{P} = \boldsymbol{\Psi}(\boldsymbol{\theta}, \mathbf{P})$. In particular, we want to emphasize that they are different estimators for models of dynamic games. The NPL estimator is the maximizer of $\hat{Q}(\boldsymbol{\theta}, \mathbf{P})$ among $(\hat{\boldsymbol{\theta}}_{\text{NPL}}, \hat{\mathbf{P}}_{\text{NPL}})$ satisfying two conditions: (i) $\hat{\mathbf{P}}_{\text{NPL}} = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\text{NPL}}, \hat{\mathbf{P}}_{\text{NPL}})$ and (ii) $\hat{\boldsymbol{\theta}}_{\text{NPL}} = \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_{\text{NPL}})$. The constrained MLE is the maximizer of $\hat{Q}(\boldsymbol{\theta}, \mathbf{P})$ among $(\hat{\boldsymbol{\theta}}_{\text{ML}}, \hat{\mathbf{P}}_{\text{ML}})$ satisfying three conditions: (i) $\hat{\mathbf{P}}_{\text{ML}} = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\text{ML}}, \hat{\mathbf{P}}_{\text{ML}})$, (ii) $\nabla_{\boldsymbol{\theta}} \hat{Q}_{(\hat{\boldsymbol{\theta}}_{\text{ML}}, \hat{\mathbf{P}}_{\text{ML}})} - \boldsymbol{\lambda}' \nabla_{\boldsymbol{\theta}} \boldsymbol{\Psi}_{(\hat{\boldsymbol{\theta}}_{\text{ML}}, \hat{\mathbf{P}}_{\text{ML}})} = \mathbf{0}$,

and (iii) $\nabla_{\mathbf{P}}\hat{Q}_{(\hat{\boldsymbol{\theta}}_{\mathrm{ML}},\hat{\mathbf{P}}_{\mathrm{ML}})} - \boldsymbol{\lambda}'\nabla_{\mathbf{P}}\boldsymbol{\Psi}_{(\hat{\boldsymbol{\theta}}_{\mathrm{ML}},\hat{\mathbf{P}}_{\mathrm{ML}})} = \mathbf{0}$. While conditions (i) and (ii) are asymptotically equivalent across both estimators, the NPL estimator does not impose the Lagrange first-order conditions with respect to $\mathbf{P}$ corresponding to the constrained MLE's condition (iii). This difference potentially leads to substantial computational savings in large state spaces at the cost of losing asymptotic efficiency. This lost of efficiency only arises when the NPL estimator is applied to empirical games. For single-agent models, the zero-Jacobian property (Aguirregabiria and Mira (2002)) implies that the NPL estimator always satisfies condition (iii) such that both estimators are equivalent in that setting.

The main challenge in obtaining the NPL estimator is how to compute the fixed point(s) of the sample NPL mapping. Aguirregabiria and Mira (2007) proposed to use the NPL fixed-point algorithm that is defined in the next subsection.

### 4.2 *NPL fixed-point algorithm*

Let $\hat{\mathbf{P}}_0$ be a vector of nonparametric or reduced-form estimates of the CCPs. Let $\hat{\boldsymbol{\theta}}_1$ be the two-step pseudo-maximum likelihood estimator—or Hotz-Miller estimator—as in Definition 1. These estimates can be used to update the CCPs using $\hat{\mathbf{P}}_1 = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_1, \hat{\mathbf{P}}_0)$, construct a new pseudo log-likelihood function $\hat{Q}(\boldsymbol{\theta}, \hat{\mathbf{P}}_1)$ to be maximized, and so on. Such an iterative process is referred to as the *NPL fixed-point algorithm*. When iterated for a fixed $K$ number of times, this approach delivers the $K$-step estimator.

Definition 5 (NPL Fixed-Point Algorithm and $K$-Step Estimator). The $k$th iteration of the NPL fixed-point algorithm for $k \in \mathbb{N}$ is defined as $\hat{\mathbf{P}}_k = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_{k-1})$, or equivalently,

$$\hat{\mathbf{P}}_k = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_k, \hat{\mathbf{P}}_{k-1}) \tag{6}$$

with

$$\hat{\boldsymbol{\theta}}_k \equiv \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_{k-1}) = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \hat{Q}(\boldsymbol{\theta}, \hat{\mathbf{P}}_{k-1}) \tag{7}$$

and $\hat{\mathbf{P}}_0$ is a reduced-form estimate of the CCPs. The $K$-step estimator iterates until $k = K$.

It is easy to see that, upon convergence, the NPL fixed-point algorithm delivers a fixed point of the sample NPL mapping. It is worth repeating that the estimator obtained upon convergence of the NPL algorithm does not necessarily correspond to the NPL estimator in Definition 4 as the NPL estimator is one of the potentially multiple fixed points of the sample NPL mapping.

It is important to emphasize that fixed-point iterations in the NPL mapping is only one possible method to obtain an NPL fixed point. The NPL algorithm failing to converge should therefore not be interpreted as evidence against the existence of a NPL fixed point since—as already mentioned above—the sample NPL mapping has at least one fixed point. Furthermore, even if the NPL algorithm may converge to different NPL fixed points when initiated at distinct starting values, it may still fail to obtain all NPL

fixed points. The reasons for such lack of convergence and failure to deliver the full set of NPL fixed points will be presented in Section 5.

Nonetheless, there are two cases where the NPL algorithm is guaranteed to converge to the consistent NPL estimator. First, when applied to single-agent dynamic discrete choice models, convergence follows from the zero-Jacobian property shown by Aguirregabiria and Mira (2002, Proposition 2). Second, Aguirregabiria and Mira (2007, Example 5) highlighted several appealing properties of the NPL algorithm when the number of structural parameters in $\boldsymbol{\theta}$ is exactly equal to the number of free probabilities in $\mathbf{P}$. In fact, the NPL mapping then has a unique fixed point, this fixed point is stable, and the NPL algorithm reaches this consistent fixed point in only two iterations.

### 4.3 *Relaxation algorithm*

Since it will be included in the simulation exercises in Section 6, we now define one of the modified NPL algorithms proposed by Kasahara and Shimotsu (2012).

Definition 6 (Relaxation Algorithm and $K$-Step Relaxation Estimator). The $k$th iteration of the relaxation algorithm for $k \in \mathbb{N}$ is defined as

$$\hat{\mathbf{P}}_k = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_k, \hat{\mathbf{P}}_{k-1})^{\alpha} * (\hat{\mathbf{P}}_{k-1})^{1-\alpha} \tag{8}$$

with

$$\hat{\boldsymbol{\theta}}_k \equiv \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_{k-1}) = \arg\max_{\boldsymbol{\theta}\in\Theta} \hat{Q}(\boldsymbol{\theta}, \hat{\mathbf{P}}_{k-1}), \tag{9}$$

where $\hat{\mathbf{P}}_0$ is a reduced-form estimate of the CCPs and $*$ denotes Hadamard product. The $K$-step relaxation estimator iterates until $k = K$.

In other words, this modified NPL algorithm updates the CCPs by using a log-linear combination of the $k$th CCPs and their corresponding best response. As pointed out by Kasahara and Shimotsu (2012), such a log-linear combination is known as the relaxation method in numerical analysis. For the rest of the paper, we will therefore refer to this algorithm as the relaxation algorithm. The mapping used to update the CCPs will be the relaxation mapping

Ideally, one would like to choose $\alpha$ optimally by using the value that minimizes the spectral radius of the Jacobian of the relaxation mapping evaluated at the true CCPs and structural parameters. Since these values are unknown in practice, Kasahara and Shimotsu (2012, footnote 13, p. 2314) proposed using $\alpha \approx 0$.

### 4.4 *Spectral algorithm*

In this section, we propose an alternative algorithm to compute the NPL estimator. Computing the NPL estimator requires solving a system of nonlinear equations. We now describe how a derivative-free nonmonotone spectral residual method—proposed by La Cruz, Martinez, and Raydan (2006)—can be used to find a solution to this system of

equations. In contrast to the NPL algorithm, the proposed method is not based on fixed-point iterations and can be used to find unstable fixed points. Also very importantly, in contrast to Newton's method, the spectral algorithm does not require the computation nor the approximation of high dimension Jacobian matrices.

We are interested in finding the solution(s) to $\hat{\boldsymbol{\phi}}(\mathbf{P}) = \mathbf{0}$, where $\hat{\boldsymbol{\phi}}(\cdot) : \mathcal{P} \mapsto (-1,1)^{\dim\{\mathcal{P}\}}$ is a nonlinear continuous function. This boils down to solving a large dimensional system of nonlinear equations. A derivative-free spectral approach is especially appealing in the current setting for two reasons. First, it does not require the computation, nor the approximation of the Jacobian of $\hat{\boldsymbol{\phi}}(\cdot)$. Second, its small memory needs make it applicable to large dimensional $\mathbf{P}$'s.

Spectral methods initially proposed by Barzilai and Borwein (1988) are most easily understood by comparing them with other existing methods commonly used to solve nonlinear systems of equations. For instance, Newton's method would update $\hat{\mathbf{P}}_k$ to $\hat{\mathbf{P}}_{k+1}$ as

$$\hat{\mathbf{P}}_{k+1} = \hat{\mathbf{P}}_k - [\nabla \hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}]^{-1} \hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k) \tag{10}$$

which unfortunately requires computing and inverting the Jacobian matrix $\nabla \hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}$ at each iteration. Alternatively, quasi-Newton methods such as Broyden's would use

$$\hat{\mathbf{P}}_{k+1} = \hat{\mathbf{P}}_k - \mathbf{B}_k^{-1} \hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k), \tag{11}$$

where the matrices $\mathbf{B}_k$ are defined sequentially as

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_{k+1}) \boldsymbol{\Delta}\hat{\mathbf{P}}'_{k+1}}{\boldsymbol{\Delta}\hat{\mathbf{P}}'_{k+1} \boldsymbol{\Delta}\hat{\mathbf{P}}_{k+1}} \tag{12}$$

with $\boldsymbol{\Delta}\hat{\mathbf{P}}_{k+1} \equiv \hat{\mathbf{P}}_{k+1} - \hat{\mathbf{P}}_k$. While rank-one updates could be used to obtain $\mathbf{B}_{k+1}^{-1}$, the inverse of $\nabla \hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}$ is still typically approximated at the first iteration and the approach requires carrying potentially large dimensional matrices in large state spaces. Spectral methods rather propose to update the CCPs using:

$$\hat{\mathbf{P}}_{k+1} = \hat{\mathbf{P}}_k - \alpha_k \hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k), \tag{13}$$

where $\alpha_k$ is the spectral steplength. A key feature of the spectral step length is that it is a scalar. Different $\alpha_k$'s have been proposed in the literature. Let $\boldsymbol{\Delta}\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k) \equiv \hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k) - \hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_{k-1})$. Barzilai and Borwein (1988) proposed using

$$\alpha_k = \frac{\boldsymbol{\Delta}\hat{\mathbf{P}}'_k \boldsymbol{\Delta}\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)}{\boldsymbol{\Delta}\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)' \boldsymbol{\Delta}\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)}. \tag{14}$$

Alternatively, La Cruz, Martinez, and Raydan (2006) proposed

$$\alpha_k = \frac{\boldsymbol{\Delta}\hat{\mathbf{P}}'_k \boldsymbol{\Delta}\hat{\mathbf{P}}_k}{\boldsymbol{\Delta}\hat{\mathbf{P}}'_k \boldsymbol{\Delta}\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)}. \tag{15}$$

Finally, Varadhan and Roland (2008) proposed using

$$\alpha_k = \text{sgn}\big(\Delta\hat{\mathbf{P}}_k'\Delta\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)\big)\frac{\|\Delta\hat{\mathbf{P}}_k\|}{\|\Delta\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)\|}, \tag{16}$$

where $\text{sgn}(a) = a/|a|$ if $a \neq 0$; $\text{sgn}(a) = 0$ otherwise. Commonly used values for the initial $\alpha_0$ are $\alpha_0 = 1$ or $\alpha_0 = \min\{1, 1/|\hat{\boldsymbol{\phi}}(\mathbf{P}_0)|\}$.

A consequence of using spectral step lengths as the ones presented above is that they generate nonmonotone updating processes. An important contribution of La Cruz, Martinez, and Raydan (2006) has been to propose a nonmonotone line search as a globalization strategy that allows for such nonmonotone behavior. More precisely, for some merit function $f(\cdot)$ such that $f(\mathbf{P}) = 0 \Leftrightarrow \hat{\boldsymbol{\phi}}(\mathbf{P}) = \mathbf{0}$, each step must satisfy

$$f(\hat{\mathbf{P}}_{k+1}) \leq \max_{0\leq j\leq J-1} f(\hat{\mathbf{P}}_{k-j}) + \eta_k - \gamma\alpha_k^2 f(\hat{\mathbf{P}}_k), \tag{17}$$

where $J$ is a nonnegative integer, $0 < \gamma < 1$ and $\sum_k \eta_k < \infty$. This globalization strategy is a safeguard that can improve the performance of the algorithm.

La Cruz, Martinez, and Raydan (2006) have shown attractive local convergence properties for this spectral residual approach. A solution $\hat{\mathbf{P}}_*$ is said to be isolated if there exists $\kappa > 0$ such that $\hat{\boldsymbol{\phi}}(\mathbf{P}) \neq \mathbf{0}$ whenever $\|\mathbf{P} - \hat{\mathbf{P}}_*\| \leq \kappa$. If an isolated solution is a limit point of the sequence of $\hat{\mathbf{P}}$'s generated by the algorithm, then the whole sequence converges to this solution (La Cruz, Martinez, and Raydan (2006, Theorem 3)). Furthermore, a solution $\hat{\mathbf{P}}_*$ is said to be strongly isolated if $\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_*) = \mathbf{0}$ and there exists $\kappa > 0$ such that $\|\mathbf{P} - \hat{\mathbf{P}}_*\| \leq \kappa$ implies that $\hat{\boldsymbol{\phi}}(\mathbf{P})'[\nabla\hat{\boldsymbol{\phi}}_{(\mathbf{P})}]'\hat{\boldsymbol{\phi}}(\mathbf{P}) \neq 0$. If the point used to initialize the algorithm is close enough to some strongly isolated solution, the whole sequence converges to this solution (La Cruz, Martinez, and Raydan (2006, Theorem 4)).

As it can be seen by comparing (10) with (13), the spectral algorithm described here uses a scalar $1/\alpha_k$ to approximate the matrix $\nabla\hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}$. A useful intuition to understand why such a scalar helps in finding the solution of a nonlinear system of equations is given by Fletcher (1990). For instance, notice that the formula for $\alpha_k$ proposed by La Cruz, Martinez, and Raydan (2006) as written in (15) implies that

$$\frac{1}{\alpha_k} = \frac{\Delta\hat{\mathbf{P}}_k'\Delta\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k)}{\Delta\hat{\mathbf{P}}_k'\Delta\hat{\mathbf{P}}_k} \approx \frac{\Delta\hat{\mathbf{P}}_k'\nabla\hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}\Delta\hat{\mathbf{P}}_k}{\Delta\hat{\mathbf{P}}_k'\Delta\hat{\mathbf{P}}_k} = \frac{\Delta\hat{\mathbf{P}}_k'\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)}\Delta\hat{\mathbf{P}}_k}{\Delta\hat{\mathbf{P}}_k'\Delta\hat{\mathbf{P}}_k}, \tag{18}$$

where the approximation follows from $\Delta\hat{\boldsymbol{\phi}}(\hat{\mathbf{P}}_k) \approx \nabla\hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}\Delta\hat{\mathbf{P}}_k$, that is, the secant formula for the Jacobian, and $\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)} = [\nabla\hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)} + \nabla\hat{\boldsymbol{\phi}}'_{(\hat{\mathbf{P}}_k)}]/2$ is the symmetric part of $\nabla\hat{\boldsymbol{\phi}}_{(\hat{\mathbf{P}}_k)}$. In other words, $1/\alpha_k$ is the Rayleigh quotient with respect to a secant approximation of $\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)}$. Since $\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)}$ is symmetric, it is diagonalizable with only real eigenvalues. Well-known properties of Rayleigh quotients imply that $1/\alpha_k$ is bounded by the smallest and the largest eigenvalues of $\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)}$. Furthermore, one can interpret the last equality in (18) as a weighted average of the eigenvalues of $\nabla\hat{\boldsymbol{\phi}}^s_{(\hat{\mathbf{P}}_k)}$ with weights given by the square of the corresponding coordinate of $\Delta\hat{\mathbf{P}}_k$ in the eigenbasis. This weighted average therefore assigns larger weights to the elements of $\mathbf{P}$ in the eigenbasis, which varied the most between step $k-1$ and $k$.

The updating of the spectral step length $\alpha_k$ at each iteration is also a relevant distinctive feature of the spectral approach when compared to the relaxation algorithm described in Section 4.3. Notice that, based on the definition of the relaxation method (Definition 6), $\ln[\hat{\mathbf{P}}_{k+1}]$ can be written as

$$\ln[\hat{\mathbf{P}}_{k+1}] = \ln[\hat{\mathbf{P}}_k] - \alpha\big\{\ln[\hat{\mathbf{P}}_k] - \ln\big[\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{k+1}, \hat{\mathbf{P}}_k)\big]\big\}. \tag{19}$$

Since $\ln[\hat{\mathbf{P}}] - \ln[\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}(\hat{\mathbf{P}}), \hat{\mathbf{P}})] = 0$ if and only if $\hat{\mathbf{P}} = \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}(\hat{\mathbf{P}}), \hat{\mathbf{P}})$, the part of equation (19) inside braces corresponds to a residual function which is equal to $\mathbf{0}$ only at the NPL fixed point(s). One can therefore interpret the log-linearized version of the relaxation algorithm as a "spectral method" that updates $\ln[\hat{\mathbf{P}}_k]$ using a *constant* "spectral step length" $\alpha$. Our comparison of the performance of the relaxation and the spectral algorithms shows that updating the scalar spectral step length as part of the iterative process—once again, similarly as Newton and quasi-Newton methods update approximations of the inverse of a matrix— leads to substantially better convergence properties. Furthermore, a log-linearized version of the relaxation algorithm as in (19) that would update the spectral step length according to the expressions of $\alpha_k$ reported above would be locally convergent in a neighborhood of a strongly isolated solution (once again by La Cruz, Martinez, and Raydan (2006, Theorem 4)).

There are three other points that are worth mentioning. First, since there is no guarantee that the NPL sample mapping has a unique fixed point, this iterative procedure should be initiated at different starting values. Second, the procedure described above is also iterative, just like the NPL algorithm. However, it does not make fixed-point iterations over the NPL mapping. For this reason, the spectral approach is able to find unstable fixed points that would not be attainable via the NPL algorithm. Third, it should be noted that the spectral approach is not much harder to code than NPL: one basically uses the code of a $K = 1$ step iteration to construct the system of nonlinear equations to be solved.

### 4.5 *Comparing domains of attraction*

The spectral algorithm described above is therefore an alternative approach that can be used to compute the NPL estimator. In order to appreciate how its convergence properties compare to the NPL algorithm and to other iterative approaches available in the literature, it is useful to highlight how a given solution's domain of attraction may vary across different methods.

There is an important consequence of using the NPL algorithm to find fixed points of the sample NPL mapping: if the stability condition required for convergence—which will be provided below—fails to hold, the relevant domain of attraction is a singleton. In other words, the only way to find the NPL fixed point corresponding to the NPL estimator would be to initiate the algorithm exactly at the solution. As a result, the domain of attraction of the NPL estimator has zero Lebesgue measure when stability fails. This feature is a consequence of using fixed-point iterations in an NPL mapping that—in the case of games—is not a contraction.

As discussed above, fixed-point iterations are not the only method available to solve for the fixed points of a nonlinear system of equations. Under very general conditions—which hold in our problem—Newton or quasi-Newton algorithms have desirable local convergence properties. In fact, for these algorithms, the solution's domain of attraction is a singleton if and only if the Jacobian matrix of the system of equations to solve is singular when evaluated at the solution. Under very mild conditions, the set of such unregular solutions has zero Lebesgue measure, such that the domain of attraction of any solution is generically larger than a singleton. This observation is true even if stability fails. It follows that, when stability fails, estimating algorithms based on Newton methods are associated with domains of attraction leading to a consistent estimator that are larger than the NPL algorithm's.

This distinction between fixed-point iterations and Newton methods is key to understand the properties of different algorithms used to estimate dynamic discrete games. For instance, the properties of the NFXP applied to dynamic games crucially depend on the method used to solve for the fixed point in the inner algorithm. If one uses fixed-point iterations, instability of the best response mapping may prevent the inner algorithm to converge unless this algorithm is initiated at the fixed point of interest. In such cases, the inner algorithm's domain of attraction being a singleton implies that the domain of attraction of the NFXP algorithm must also have zero Lebesgue measure. For this reason, it may be preferable to implement a NFXP algorithm that uses Newton methods to solve for the fixed point to ensure that the algorithm locally converges to the consistent estimator.

MPEC methods can also be interpreted as a set of variations of Newton methods used to find the solution of a constrained optimization problem. As a result, the MPEC algorithm applied to dynamic discrete games should locally converge to a solution. Of course, this algorithm may still fail to converge for a given initial value. However, the set of starting values leading to a consistent estimator has a strictly positive Lebesgue measure. In fact, simulation evidence from Egesdal, Lai, and Su (2015) suggests that the MPEC could always be initiated at a starting value leading to a converging sequence, despite the algorithm failing to converge for some sequences.

To sum up, one way to guarantee convergence to the NPL estimator would be to apply a Newton algorithm to solve for a zero of the system of equations $\hat{\boldsymbol{\phi}}(\mathbf{P}) = 0$. The Newton method does not rely on fixed-point iterations and guarantees local convergence under mild regularity conditions. It is worth emphasizing a considerable advantage of the spectral approach over Newton and quasi-Newton algorithms that is especially relevant in the context of dynamic discrete games with large state spaces: replacing the approximation of the Jacobian to be inverted by a scalar that has a closed-form expression. In that sense, spectral methods offer a generic approach that is implemented in practice with limited effort. While Newton methods may have higher rates of convergence and may require a smaller number of iterations, the burden of computing or approximating the inverse of large Jacobian matrices and the tricks available to reduce this burden vary across models. For instance, MPEC methods typically leverage sparsity properties. However, it should be noted that the Jacobian of the $\boldsymbol{\Psi}(\boldsymbol{\theta}, \mathbf{P})$ with respect to $\mathbf{P}$ is typically not

sparse in games.[7] Furthermore, the dimensionality of the problem solved by the spectral algorithm remains smaller than implementing the MPEC estimator of Egesdal, Lai, and Su (2015). While solving the system of nonlinear equations requires searching over the CCPs' space, MPEC is searching over the spaces of the CCPs, the parameters, and the Lagrange multipliers.

Should one conclude that the NPL fixed-point algorithm is necessarily dominated by algorithms based on Newton or spectral methods? A single iteration of the NPL algorithm is computationally much cheaper than such alternative methods in very large state spaces. Better understanding under which conditions the NPL algorithm can be applied is therefore needed.

## 5. Local asymptotic properties of the NPL algorithm

We now turn to studying the asymptotic properties of the estimator generated by the NPL fixed-point algorithm. To shed new light on mixed simulation results found in the literature, our analysis departs from existing work in two ways. First, we are assessing the convergence properties of the NPL algorithm when it is iterated in small samples. Second, we derive asymptotic properties that are local to the stability threshold determining the convergence of the NPL algorithm in the population.

We start by distinguishing between the stability of the data generating process and the convergence of the NPL algorithm iterated in small samples. This point is key to explain why the NPL algorithm may fail to converge even if the equilibrium generating the data is stable. We show that convergence of the NPL algorithm to the NPL estimator depends on a particular property of the sample NPL mapping. We characterize the probability that the NPL algorithm delivers the NPL estimator for data generating processes that are around the stability threshold. Standard asymptotic arguments would conclude that this probability is either zero or one depending on the stability of the data generating process. Our local asymptotic analysis allows us to characterize the nondegenerate probability of the NPL algorithm delivering the NPL estimator even in large samples, which is consistent with our Monte Carlo simulations.

Furthermore, we investigate how the NPL algorithm could lead to an estimator that is asymptotically different from the NPL estimator. First, the algorithm could converge to an inconsistent fixed point of the sample NPL mapping. Second, the condition that determines the convergence of the algorithm to the consistent NPL estimator is stochastic and is not independent of the NPL estimator itself. This dependence introduces a selection bias in the NPL fixed-point algorithm estimator. To the best of our knowledge, the implications of this selection have never been explored in the literature. The relative importance of these two sources of bias depends on the data generating process, as it is highlighted in Section 6.

[7]This is an important distinction with single-agent models for which the Jacobian is the zero matrix. Since a given player's payoffs depend on the other players' decisions, derivatives with respect to other players' choice probabilities are nonzero. As a result, the Jacobian matrix displays blocks of zeros along the diagonal, but the remaining off-diagonal blocks are nonzero.

DEFINITION 7 (NPL Fixed Points in the Population and in the Sample). $\mathcal{F}^0$ is the set of NPL fixed points of the mapping $\boldsymbol{\varphi}^0(\cdot)$. For any $\mathbf{P}^0_* \in \mathcal{F}^0$, let $\boldsymbol{\theta}^0_* \equiv \boldsymbol{\vartheta}^0(\mathbf{P}^0_*)$. Similarly, $\widehat{\mathcal{F}}$ is the set of NPL fixed points of the mapping $\hat{\boldsymbol{\varphi}}(\cdot)$. For any $\hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$, let $\hat{\boldsymbol{\theta}}_* \equiv \hat{\boldsymbol{\vartheta}}(\hat{\mathbf{P}}_*)$.

The stability of fixed points of the NPL mapping is defined in Definition 8. For a given square matrix $\mathbf{A}$, let $\rho(\mathbf{A})$ denote its spectral radius, that is, its largest eigenvalue in absolute value. The condition $\rho(\mathbf{A}) < 1$ is sufficient, but not necessary, for $\mathbf{A}^K$ to converge to $\mathbf{0}$ for large enough $K$ (where $\mathbf{A}^K$ denotes the matrix product of $K$ matrices $\mathbf{A}$).

DEFINITION 8 (Stable Fixed Points of the Population and the Sample NPL Mappings). Under Assumptions 1 and 2: (A) the data generating process is stable if $\rho(\nabla \boldsymbol{\varphi}^0_{(\mathbf{P}^0)}) < 1$ and (B) a fixed-point $\hat{\mathbf{P}}_*$ of the sample NPL mapping $\hat{\boldsymbol{\varphi}}(\cdot)$ is stable if $\rho(\nabla \hat{\boldsymbol{\varphi}}_{(\hat{\mathbf{P}}_*)}) < 1$.

By the local contraction theorem, condition $\rho(\nabla \boldsymbol{\varphi}^0_{(\mathbf{P}^0)}) < 1$ implies that there is a neighborhood of $\mathbf{P}^0$—say $\mathcal{N}(\mathbf{P}^0)$—such that an algorithm that starts at a vector within $\mathcal{N}(\mathbf{P}^0)$ and iterates in the population NPL mapping must converge to $\mathbf{P}^0$. Similarly, under condition $\rho(\nabla \hat{\boldsymbol{\varphi}}_{(\hat{\mathbf{P}}_*)}) < 1$, there is a neighborhood $\mathcal{N}(\hat{\mathbf{P}}_*)$ such that starting at a vector within this neighborhood and iterating in the sample NPL mapping eventually converges to the fixed- point $\hat{\mathbf{P}}_*$. Very importantly, the stability condition in the population NPL mapping is not sufficient to guarantee the convergence of the NPL algorithm in finite samples.

Kasahara and Shimotsu (2012, Proposition 1) studied the properties of the NPL algorithm estimator under the condition that the sample size is large enough such that: (i) the sample NPL mapping can be approximated arbitrarily well using the population NPL mapping and (ii) the algorithm starts at a consistent nonparametric estimator that is arbitrarily close to $\mathbf{P}^0$. This is the reason why their convergence conditions are stated in terms of $\rho(\nabla \boldsymbol{\varphi}^0_{(\mathbf{P}^0)})$ as in Definition 8(A). In contrast, our approach studies the properties of estimators when convergence of the algorithm is determined in small samples according to Definition 8(B). Our approach echoes criticisms to the asymptotic approximation—formulated by Pesendorfer and Schmidt-Dengler (2010)—and it is more in line with the idea of using the NPL algorithm as a device to reduce finite sample bias.

To alleviate the notation, let $\hat{\rho}_{\text{NPL}} \equiv \rho(\nabla \hat{\boldsymbol{\varphi}}_{(\hat{\mathbf{P}}_{\text{NPL}})})$ and $\rho^0 \equiv \rho(\nabla \boldsymbol{\varphi}^0_{(\mathbf{P}^0)})$. Since the sample NPL mapping is a random object, one cannot guarantee that the stability of the equilibrium generating the data implies that $\hat{\rho}_{\text{NPL}} < 1$ for a fixed sample size $M$. The randomness of the sample NPL mapping implies that—for finite $M$—there is always a strictly positive probability that $\hat{\rho}_{\text{NPL}} > 1$, even when the population is such that $\rho^0 < 1$. This observation can explain the lack of convergence of the NPL algorithm when the data generating process is stable. Nonetheless, standard asymptotic arguments would conclude that, as $M \to \infty$, $\Pr(\hat{\rho}_{\text{NPL}} < 1)$ is either 0 or 1 depending on the sign of $1 - \rho^0$. To preserve nondegenerate probabilities of $\hat{\rho}_{\text{NPL}}$ being smaller than 1 when $\rho^0 > 1$ even in large samples, we consider an asymptotic analysis that is local to $\rho^0 = 1$. In other words, we consider a sequence of stable data generating processes, which depend on the sample size $M$ and asymptotically converge to a data generating process satisfying $\rho^0 = 1$, in which the equilibrium generating the data is unstable.

The local asymptotics framework has been used in different branches of literature in econometrics. For instance, it has been applied in seminal contributions studying the problems of specification testing (e.g., Hausman (1978)), unit roots in autoregressive models (e.g., Phillips (1987), Phillips and Perron (1988), Elliott, Rothenberg, and Stock (1996)) and weak instruments (e.g., Staiger and Stock (1997)), among others. In many cases, the local asymptotics framework is used to analyze the power of statistical tests. Recently, Bugni and Ura (2019) used local asymptotics to study the effect of local misspecification on sequential estimation for single-agent dynamic discrete choice models. In our setting, we leverage such asymptotic arguments to derive the distribution of the estimator obtained upon convergence of the NPL algorithm when the data generating process is local to $\rho^0 = 1$. We are focusing on this specific value of $\rho^0$ since it is the threshold defining stability of NPL fixed points in the population.

Our local asymptotics consider that there is a sequence of data generating processes $\{\mathbf{P}_M^0 : M \geq 1\}$ defined by

$$\mathbf{P}_M^0 = \mathbf{P}^0 + \frac{\mathbf{c_P}}{\sqrt{M}} \tag{20}$$

for some unknown constant vector $\mathbf{c_P}$. This sequence of data generating processes is such that $\mathbf{P}_M^0 - \mathbf{P}^0 = O(M^{-1/2})$. Let $Q_M^0(\boldsymbol{\theta}, \mathbf{P}) \equiv \mathrm{E}_{\mathbf{P}_M^0}[\ln[\Psi(\mathbf{y}_m|\mathbf{x}_m, \boldsymbol{\theta}, \mathbf{P})]]$, where $\mathrm{E}_{\mathbf{P}_M^0}[\cdot]$ is computed with respect to $\mathbf{P}_M^0$. By maximizing $Q_M^0(\boldsymbol{\theta}, \mathbf{P})$ over $\Theta$ for a given $\mathbf{P} \in \mathcal{P}$, one obtains $\boldsymbol{\vartheta}_M^0(\mathbf{P}) \equiv \arg\max_{\boldsymbol{\theta}\in\Theta} Q_M^0(\boldsymbol{\theta}, \mathbf{P})$. Let $\boldsymbol{\theta}_M^0 \equiv \boldsymbol{\vartheta}_M^0(\mathbf{P}_M^0)$, and let $\rho_M^0 \equiv \rho(\nabla \boldsymbol{\varphi}_{M,(\mathbf{P}_M^0)}^0)$ where $\nabla \boldsymbol{\varphi}_{M,(\mathbf{P}_M^0)}^0$ is the Jacobian of $\boldsymbol{\varphi}_M^0(\mathbf{P}) = \boldsymbol{\Psi}(\boldsymbol{\vartheta}_M^0(\mathbf{P}), \mathbf{P})$ evaluated at $\mathbf{P}_M^0$.

The following Lemma 1 provides useful results about the local asymptotic properties of NPL fixed points. Parts 1(A)–1(B) establish that the sequence of data generating processes $\{\mathbf{P}_M^0 : M \geq 1\}$ implies sequences of sets of fixed points denoted $\mathcal{F}_M^0$. Parts 1(C)–1(D) derive the local asymptotic properties of the (algorithm-free) statistics $\hat{\mathbf{P}}_*$ and $\hat{\boldsymbol{\theta}}_*$ as $M \to \infty$ under $\{\mathbf{P}_M^0 : M \geq 1\}$.

Lemma 1 (Local Asymptotic Properties of NPL Fixed Points). *Let Assumptions 1 and 2 be satisfied. As $M \to \infty$ under $\{\mathbf{P}_M^0 : M \geq 1\}$, the following statements hold*:

(A) *Let $\{\mathcal{F}_M^0 : M \geq 1\}$ be the sequence of the set of fixed points of the mapping $\boldsymbol{\varphi}_M^0(\cdot)$. Every set in this sequence is nonempty. Furthermore, this sequence converges to $\mathcal{F}^0$.*

(B) *For any $\mathbf{P}_{M*}^0 \in \mathcal{F}_M^0$, there exists a $\mathbf{P}_*^0 \in \mathcal{F}^0$ and some constant vectors $\mathbf{c}_\mathbf{P}^*$, $\mathbf{c}_{\boldsymbol{\theta}}^*$ such that $\mathbf{P}_{M*}^0 = \mathbf{P}_*^0 + \mathbf{c}_\mathbf{P}^*/\sqrt{M} + o(M^{-1/2})$ and $\boldsymbol{\theta}_{M*}^0 = \boldsymbol{\theta}_*^0 + \mathbf{c}_{\boldsymbol{\theta}}^*/\sqrt{M} + o(M^{-1/2})$ where $\boldsymbol{\theta}_{M*}^0 \equiv \boldsymbol{\vartheta}_M^0(\mathbf{P}_{M*}^0)$.*

(C) *For any $\hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$, there exists a $\mathbf{P}_{M*}^0 \in \mathcal{F}_M^0$ such that $\sqrt{M}(\hat{\mathbf{P}}_* - \mathbf{P}_{M*}^0)$ and the corresponding $\sqrt{M}(\hat{\boldsymbol{\theta}}_* - \boldsymbol{\theta}_{M*}^0)$ both converge in distribution to vectors of normal random variables with zero means.*

(D) *For any $\hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$, there exists a $\mathbf{P}_*^0 \in \mathcal{F}^0$ such that $\hat{\mathbf{P}}_* \xrightarrow{p} \mathbf{P}_*^0$, $\hat{\boldsymbol{\theta}}_* \xrightarrow{p} \boldsymbol{\theta}_*^0$.*

Proof. See Appendix B.1 of the Online Supplementary Material (Aguirregabiria and Marcoux (2021)).

Since the spectral radius of the Jacobian of the NPL mapping is key to determine whether the NPL algorithm converges to a given fixed point, we state its local asymptotic properties in Lemma 2. Using a notation similar to the one above, let $\hat{\rho}_* \equiv \rho(\triangledown \hat{\boldsymbol{\varphi}}_{(\hat{\mathbf{P}}_*)})$ for $\hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$, $\rho^0_* \equiv \rho(\triangledown \boldsymbol{\varphi}^0_{(\mathbf{P}^0_*)})$ for $\mathbf{P}^0_* \in \mathcal{F}^0$, and $\rho^0_{M*} \equiv \rho(\triangledown \boldsymbol{\varphi}^0_{M,(\mathbf{P}^0_{M*})})$ for $\mathbf{P}^0_{M*} \in \mathcal{F}^0_M$. Lemma 2 establishes the asymptotic distribution of the statistic $\hat{\rho}_*$ as $M \to \infty$ under $\{\mathbf{P}^0_M : M \geq 1\}$.

LEMMA 2 (Local Asymptotic Properties of the Spectral Radius). *Let Assumptions 1 and 2 be satisfied and let $\mathbf{P}^0_M = \mathbf{P}^0 + \mathbf{c}_\mathbf{P}/\sqrt{M}$ for some unknown constant vector $\mathbf{c}_\mathbf{P}$. Then, as $M \to \infty$ under $\{\mathbf{P}^0_M : M \geq 1\}$, for $\hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$, there exist a $\mathbf{P}^0_{M*} \in \mathcal{F}^0_M$ and a $\mathbf{P}^0_* \in \mathcal{F}^0$ such that*:

(A) $\rho^0_{M*} = \rho^0_* + c^*_\rho/\sqrt{M} + o(M^{-1/2})$;

(B) $\sqrt{M}(\hat{\rho}_* - \rho^0_{M*}) \xrightarrow{d} \text{Normal}(0, \sigma^2_{\rho^0_*})$ *where $\sigma^2_{\rho^0_*}$ is defined in Appendix* B.2; *and*

(C) $\Pr(\hat{\rho}_* > \rho^0_*) \to \Phi(\frac{c^*_\rho}{\sigma_{\rho^0_*}})$ *where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal.*[8]

PROOF. See Appendix B.2.

Lemmas 1 and 2 apply to all the NPL fixed points, including the NPL estimator. Therefore, a corollary of Lemmas 1 and 2 is that, as $M \to \infty$ under $\{\mathbf{P}^0_M : M \geq 1\}$, the (algorithm-free) NPL estimator $(\hat{\boldsymbol{\theta}}_{\text{NPL}}, \hat{\mathbf{P}}_{\text{NPL}})$ converges in probability to $(\boldsymbol{\theta}^0, \mathbf{P}^0)$. Moreover, $\sqrt{M}(\hat{\boldsymbol{\theta}}_{\text{NPL}} - \boldsymbol{\theta}^0_M)$ and $\sqrt{M}(\hat{\mathbf{P}}_{\text{NPL}} - \mathbf{P}^0_M)$ converges in distribution to vectors of random variables with zero means. Finally, $\Pr(\hat{\rho}_{\text{NPL}} > \rho^0) \to \Phi(\frac{c^0_\rho}{\sigma_{\rho^0}})$ where $c^0_\rho$ is such that $\rho^0_M = \rho^0 + c^0_\rho/\sqrt{M} + o(M^{-1/2})$.

The rest of this section presents the local asymptotic properties of the estimator generated by the NPL fixed-point algorithm, that we represent as $(\hat{\boldsymbol{\theta}}_{\text{FP}}, \hat{\mathbf{P}}_{\text{FP}})$ to distinguish it from the NPL estimator. In cases where the NPL algorithm finds multiple fixed points, the FP estimator is the one that maximizes the pseudo log-likelihood among all fixed points found by the algorithm. When the sequence $\{\rho^0_M : M \geq 1\}$ satisfies the stability condition $\rho^0_M < 1$ but its limit is $\rho^0_M \to \rho^0 = 1$, the probability of $\hat{\rho}_{\text{NPL}} > 1$ is strictly positive as $M \to \infty$, such that the NPL estimator may not be a stable fixed point of the sample mapping $\hat{\boldsymbol{\varphi}}(\cdot)$ even if the data generating process satisfies $\rho^0_M < 1$. We can distinguish three possible scenarios for the FP estimator.

*Scenario 1 ($\hat{\rho}_{\text{NPL}} < 1$)* The NPL estimator is a stable fixed point of the sample NPL mapping and the NPL algorithm converges to this fixed point such that $\hat{\boldsymbol{\theta}}_{\text{FP}} = \hat{\boldsymbol{\theta}}_{\text{NPL}}$.

[8] It is worth noting that, for any $\bar{\rho} \neq \rho^0_*$,

$$\lim_{M\to\infty} \Pr(\hat{\rho}_* > \bar{\rho}) = \Phi\left(-\frac{\lim_{M\to\infty} \sqrt{M}[\bar{\rho} - \rho^0_{M*}]}{\sigma_{\rho^0_*}}\right)$$

is either 0 or 1 under $\{\mathbf{P}^0_M : M \geq 1\}$ (depending on the sign of $\bar{\rho} - \rho^0_{M*}$). The reason why $\lim_{M\to\infty} \Pr(\hat{\rho}_* > \bar{\rho})$ is degenerate while $\lim_{M\to\infty} \Pr(\hat{\rho}_* > \rho^0_*)$ is nondegenerate is because $\rho^0_{M*} - \rho^0_* = O(M^{-1/2})$.

*Scenario 2 ($\hat{\rho}_{\text{NPL}} \geq 1$ and $\exists \hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$ such that $\hat{\rho}_* < 1$)* The NPL algorithm does not converge to the NPL estimator, but it converges to the stable fixed-point $\hat{\mathbf{P}}_*$ (that maximizes the pseudo log-likelihood among all fixed points found by the algorithm). The resulting FP estimator $\hat{\boldsymbol{\theta}}_{\text{FP}} \neq \hat{\boldsymbol{\theta}}_{\text{NPL}}$ is not a consistent estimator of $\boldsymbol{\theta}^0$.

*Scenario 3 ($\hat{\rho}_{\text{NPL}} \geq 1$ and $\hat{\rho}_* \geq 1$ $\forall \hat{\mathbf{P}}_* \in \widehat{\mathcal{F}}$)* The NPL fixed-point algorithm does not converge and the FP estimator $\hat{\boldsymbol{\theta}}_{\text{FP}}$ does not exist.

For what follows, we consider that the NPL fixed-point algorithm always finds all the stable fixed points in $\widehat{\mathcal{F}}$. In practice, finding all such stable fixed points is achieved by initializing the NPL algorithm at sufficiently many starting values. For the sake of simplicity, but without loss of generality for our results, we consider that the mapping $\hat{\boldsymbol{\varphi}}(\cdot)$ has at most only one fixed point other than the NPL estimator. In other words, we consider that $\widehat{\mathcal{F}} = \{\hat{\mathbf{P}}_{\text{NPL}}, \hat{\mathbf{P}}_*\}$, but we allow for cases where $\hat{\mathbf{P}}_* = \hat{\mathbf{P}}_{\text{NPL}}$. The following equation summarizes the three possible scenarios.

$$\hat{\boldsymbol{\theta}}_{\text{FP}} = \begin{cases} \hat{\boldsymbol{\theta}}_{\text{NPL}} & \text{if } \hat{\rho}_{\text{NPL}} < 1, \\ \hat{\boldsymbol{\theta}}_* \neq \hat{\boldsymbol{\theta}}_{\text{NPL}} & \text{if } \hat{\rho}_{\text{NPL}} \geq 1 \text{ and } \hat{\rho}_* < 1, \\ \text{does not exist} & \text{if } \min\{\hat{\rho}_{\text{NPL}}, \hat{\rho}_*\} \geq 1. \end{cases} \tag{21}$$

Equation (21) shows two sources of bias in the FP estimator. First, $\hat{\boldsymbol{\theta}}_{\text{FP}}$ is a mixture of the consistent estimator $\hat{\boldsymbol{\theta}}_{\text{NPL}}$ and the inconsistent estimator $\hat{\boldsymbol{\theta}}_*$. Second, the FP estimator exists only if condition $\min\{\hat{\rho}_{\text{NPL}}, \hat{\rho}_*\} < 1$ holds. This condition introduces a truncation in the distribution of the FP estimator. Even when $\hat{\boldsymbol{\theta}}_{\text{NPL}}$ is the only NPL fixed point—such that there is not the first source of bias—the distribution of the FP estimator is equal to the distribution of $\hat{\boldsymbol{\theta}}_{\text{NPL}}$ conditional on $\hat{\rho}_{\text{NPL}} < 1$. This truncation introduces a selection bias in the FP estimator.

Proposition 1 establishes the local asymptotic distribution of the FP estimator conditional on the existence of this estimator, that is, conditional on $\min\{\hat{\rho}_{\text{NPL}}, \hat{\rho}_*\} < 1$. This Proposition includes three different cases of local asymptotics. In all these cases, the limit of $\rho_M^0$ is $\rho^0 = 1$. The three cases correspond to different values for the limit $\rho_*^0$ of the sequence $\{\rho_{M*}^0\}$: (1) $\rho_*^0 < 1$, (2) $\rho_*^0 = 1$, and (3) $\rho_*^0 > 1$.

Proposition 1 (Local Asymptotic Distribution of the FP Estimator). *Let Assumptions 1 and 2 be satisfied and let $\mathbf{P}_M^0 = \mathbf{P}^0 + \mathbf{c}_{\mathbf{P}}/\sqrt{M}$ for some unknown constant vector $\mathbf{c}_{\mathbf{P}}$. Under $\{\mathbf{P}_M^0 : M \geq 1\}$ as $M \to \infty$, Lemma 1(C) implies that $\sqrt{M}(\hat{\boldsymbol{\theta}}_{\text{NPL}} - \boldsymbol{\theta}_M^0) \xrightarrow{d} \boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0)$ and $\sqrt{M}(\hat{\boldsymbol{\theta}}_* - \boldsymbol{\theta}_{M*}^0) \xrightarrow{d} \boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}_*^0)$ where $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0)$ and $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}_*^0)$ are vectors of normal random variables with zero means. Furthermore, Lemma 2(B) implies that $\sqrt{M}(\hat{\rho}_{\text{NPL}} - \rho_M^0) \xrightarrow{d} \xi_\rho(\mathbf{P}^0)$ and $\sqrt{M}(\hat{\rho}_* - \rho_{M*}^0) \xrightarrow{d} \xi_\rho(\mathbf{P}_*^0)$ where $\xi_\rho(\mathbf{P}^0)$ and $\xi_\rho(\mathbf{P}_*^0)$ are scalar normal random variables. Then, when the limit of the sequence $\{\rho_M^0 : M \geq 1\}$ is $\rho^0 = 1$, the following results hold:*

(A) *The limiting distribution of the event* $E_M = \{\min\{\hat{\rho}_{\text{NPL}}, \hat{\rho}_*\} < 1\}$ *is*

$$\lim_{M\to\infty} \Pr(E_M) = \begin{cases} 1 & \text{if } \rho_*^0 < 1, \\ \Pr\big(\min\big\{\xi_\rho(\mathbf{P}^0) + c_\rho^0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^*\big\} < 0\big) & \text{if } \rho_*^0 = 1, \\ \Pr\big(\xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big) & \text{if } \rho_*^0 > 1. \end{cases} \tag{22}$$

(B) *For any set* $\mathcal{B}$, *the limit of* $\Pr(\sqrt{M}(\hat{\boldsymbol{\theta}}_{\text{FP}} - \boldsymbol{\theta}_M^0) \in \mathcal{B} | E_M)$ *as* $M \to \infty$ *is as follows.*

*If* $\rho_*^0 < 1$,

$$\begin{aligned} &\Pr\big(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0) \in \mathcal{B} | \xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big) \Pr\big(\xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big) \\ &\quad + \Pr\Big(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}_*^0) + \lim_{M\to\infty} \sqrt{M}(\boldsymbol{\theta}_{M*}^0 - \boldsymbol{\theta}_M^0) \in \mathcal{B} | \xi_\rho(\mathbf{P}^0) + c_\rho^0 \geq 0\Big) \\ &\quad \times \Pr\big(\xi_\rho(\mathbf{P}^0) + c_\rho^0 \geq 0\big). \end{aligned} \tag{23}$$

*If* $\rho_*^0 = 1$,

$$\begin{aligned} &\Pr\big(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0) \in \mathcal{B} | \xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big) \frac{\Pr\big(\xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big)}{\Pr\big(\min\big\{\xi_\rho(\mathbf{P}^0) + c_\rho^0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^*\big\} < 0\big)} \\ &\quad + \Pr\Big(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}_*^0) + \lim_{M\to\infty} \sqrt{M}(\boldsymbol{\theta}_{M*}^0 - \boldsymbol{\theta}_M^0) \in \mathcal{B} \big| \xi_\rho(\mathbf{P}^0) + c_\rho^0 \geq 0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^* < 0\Big) \\ &\quad \times \frac{\Pr\big(\xi_\rho(\mathbf{P}^0) + c_\rho^0 \geq 0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^* < 0\big)}{\Pr\big(\min\big\{\xi_\rho(\mathbf{P}^0) + c_\rho^0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^*\big\} < 0\big)}. \end{aligned} \tag{24}$$

*If* $\rho_*^0 > 1$,

$$\Pr\big(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0) \in \mathcal{B} | \xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0\big). \tag{25}$$

PROOF. See Appendix B.3.

Proposition 1 shows that, for sequences of data generating processes local to $\rho^0 = 1$, the limiting distribution of $\hat{\boldsymbol{\theta}}_{\text{FP}}$ is a mixture of truncated normal distributions. The exact mixture depends on the stability of the other fixed point(s) of the NPL mapping in the population, that is, $\rho_*^0$. The mixed distributions are: the truncated distribution of the NPL estimator (Scenario 1) and the truncated distribution of the inconsistent estimator $\hat{\boldsymbol{\theta}}_*$ (Scenario 2). The truncations are due to the fact that the FP estimator only exists under some conditions (Scenario 3). The weights associated with the mixture correspond to $\Pr(\hat{\rho}_{\text{NPL}} < \rho^0)$ (from Lemma 2) conditional on the existence of $\hat{\boldsymbol{\theta}}_{\text{FP}}$.

The simplest case is when $\rho_*^0 > 1$, that is, the other fixed point is "sufficiently" unstable. Then the limiting distribution of $\hat{\boldsymbol{\theta}}_{\text{FP}}$ is simply the asymptotic distribution of the NPL estimator, conditional on the existence of $\hat{\boldsymbol{\theta}}_{\text{FP}}$. In other words, the only source of bias in that case is the convergence selection.

If the other fixed point is stable, then there is an additional—and more severe—source of bias: the possibility of the NPL algorithm converging to an inconsistent estimator. This source of bias has already been highlighted in the literature by Pesendorfer

and Schmidt-Dengler (2008, 2010). Since we are considering the case where $\rho^0 = 1$, our asymptotic results emphasize that inconsistency does not arise with probability 1.

Finally, if both fixed points in $\mathcal{F}^0$ are equal to 1, both convergence selection and convergence to an inconsistent estimator are potential sources of asymptotic bias.[9] The weights associated with each of the mixed distributions differ from the ones in the case with $\rho_*^0 < 1$ since $\rho^0 = \rho_*^0 = 1$ implies that both $\hat{\rho}_{\text{NPL}} < 1$ and $\hat{\rho}_* < 1$ can happen with nondegenerate probabilities.

Furthermore, Proposition 1 leads to interesting conclusions that shed light on convergence issues associated with the NPL fixed-point algorithm. These implications, which are supported by the simulation evidence reported in Section 6, are summarized as follows.

First, having a sequence of stable data generating processes is not sufficient for consistency of the FP estimator. If this sequence is local around $\rho^0 = 1$, there is a strictly positive probability that the fixed-point algorithm does not converge to the consistent NPL estimator. In that sense, the theoretical results presented above explain why the NPL algorithm may fail to converge or may converge to an inconsistent estimator even if the data are generated from a stable fixed point in the population. To the best of our knowledge, we are the first to provide a theoretical explanation for this property of the FP estimator, which has been observed in simulation experiences previously reported in the literature without convincing explanations. The local asymptotic approach around $\rho^0 = 1$ is especially relevant to understand cases where the data generating process is stable, but close to being unstable. Even for relatively large $M$, the randomness in the sample NPL mapping may make the iterative procedure unstable despite the stability of the fixed point in the population.

Second, the fact that the FP estimator only exists under Scenarios 1 and 2 introduces a selection bias, which we refer to as the convergence selection bias. Even in the unfeasible case where the researcher could know $\hat{\rho}_{\text{NPL}}$ and select the FP estimator only when the convergence condition $\hat{\rho}_{\text{NPL}} < 1$ holds, the convergence selection would still introduce a discrepancy between the FP and the NPL estimators. To the best of our knowledge, the effect of this convergence selection on the asymptotic properties of the FP estimator had not been studied prior to the current paper.

## 6. Monte Carlo experiments

In our Monte Carlo experiments, we consider two sets of data generating processes that have been commonly used in the literature to compare the performance of different algorithms. The first set is based on Aguirregabiria and Mira (2007); the second one is from Pesendorfer and Schmidt-Dengler (2008).

[9] It is also worth noting that $\rho_*^0 = 1$ includes $\mathcal{F}^0$ being a singleton as a special case. Then, since $\xi_\rho(\mathbf{P}^0) + c_\rho^0 = \xi_\rho(\mathbf{P}_*^0) + c_\rho^*$, $\Pr(\xi_\rho(\mathbf{P}^0) + c_\rho^0 \geq 0, \xi_\rho(\mathbf{P}_*^0) + c_\rho^* < 0) = 0$ and the limiting distribution is $\Pr(\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{P}^0) \in \mathcal{B} | \xi_\rho(\mathbf{P}^0) + c_\rho^0 < 0)$ as for the case $\rho_*^0 > 1$.

### 6.1 *Experiments I: Aguirregabiria and Mira (2007)*

The first set of data generating processes are based on Experiment 3 in Aguirregabiria and Mira (2007). The same data generating process, or a simplified version of it, was also used by Kasahara and Shimotsu (2012) and by Egesdal, Lai, and Su (2015).

The model corresponds to the dynamic game of market entry and exit presented in Section 3 above. A number $N = 5$ of firms must decide whether or not to operate in $M$ independent markets. There are $N + 1 = 6$ observable state variables: market size ($s_{mt}$) and the incumbency status of each player ($\mathbf{y}_{m,t-1}$). The support of $s_{mt}$ is $\mathcal{S} = \{1, 2, 3, 4, 5\}$. The observable state space is denoted by $\mathcal{X} \equiv \mathcal{S} \times \mathcal{Y}^N$ and the total number of possible states is $\dim(\mathcal{X}) = 5 \times 2^5 = 160$. The transition probability of incumbency statuses trivially corresponds to the firms' decisions CCPs. The transition matrix for market size $s_{mt}$ is

$$\begin{bmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{bmatrix}. \tag{26}$$

The values of the parameters in the payoff function are $\theta_{\text{RS}} = 1$, $\theta_{\text{EC}} = 1$, $\theta_{\text{FC},1} = 1.9$, $\theta_{\text{FC},2} = 1.8$, $\theta_{\text{FC},3} = 1.7$, $\theta_{\text{FC},4} = 1.6$, and $\theta_{\text{FC},5} = 1.5$. The discount rate is assumed to be known and equal to $\beta = 0.95$.

The strategic interaction parameter $\theta_{\text{RN}}$ has an important role in the stability of the data generating process. Note that $\theta_{\text{RN}} = 0$ corresponds to a single-agent dynamic discrete choice model, and in this case stability—in the sense of Definition 8(A)—is guaranteed through the zero-Jacobian property (Aguirregabiria and Mira (2002, Proposition 2)). Figure 1 presents the spectral radius $\rho^0$ for different values of $\theta_{\text{RN}}$.[10]

Simulation exercises available in the literature—that have used this data generating process—have typically focused on two specific values of $\theta_{\text{RN}}$: $\theta_{\text{RN}} = 2$, which will be referred to as the *mildly stable* case since the corresponding spectral radius at the true CCPs is 0.9237 and $\theta_{\text{RN}} = 4$, which will be the *very unstable* case with spectral radius of 1.6748. We present below Monte Carlo experiments and convergence results for these two cases. Furthermore, to provide a finer picture of the properties of the sequential estimators, we also consider a *very stable* case by setting $\theta_{\text{RN}} = 1$, corresponding spectral radius 0.4623, and a *mildly unstable* case where $\theta_{\text{RN}} = 2.4$ spectral radius equal to 1.1168. These four cases allow us to study how different degrees of (in)stability at the population level affect the statistical properties of the sample NPL mapping and the estimators. Table C1 in Appendix C.1 reports summary statistics from the simulated data.

The equilibrium CCPs needed to generate the data solve the fixed-point mapping in (1) when the unobservable state variables follow the extreme value type 1 distribution. The equilibrium CCPs can be found by using a built-in solver for a system of nonlin-

[10]The Jacobians used to compute these spectral radii have been approximated using numerical derivatives.

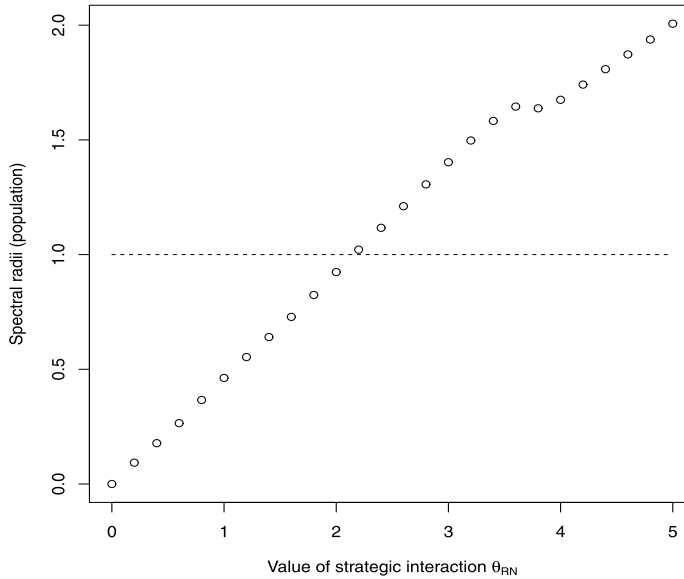



FIGURE 1. Spectral radii for different $\theta_{\text{RN}}$—Population NPL mapping.

ear equations.[11] For each version of the data generating process, we initiated the solver at 100 randomly picked starting values. All starting values returned the same vector of equilibrium CCPs.

For each version of the data generating process, we take a random sample of $M$ markets from the ergodic distribution of the state variables, and use the equilibrium CCPs to generate firms' entry/exit decisions in each market. We implement experiments with $M = 400$ and with $M = 5000$ markets. For each experiment, we draw 500 Monte Carlo samples.

We compute estimates using three different algorithms: the NPL fixed point, the relaxation, and the spectral algorithms. We iterate each algorithm at most $K = 100$ times and study the properties of the sequences of estimates. All algorithms were initialized with 5 different starting values, including the frequency estimator of the CCPs as one of them.[12] For a given algorithm, if different starting points generate different estimates, then the solution with the highest value of the pseudo-likelihood is selected as the estimate for that algorithm. We have also considered the case where the frequency estimate of CCPs is the only starting value, and we report these results in Appendix C.3.[13]

[11]We used R's BBsolve function with its default settings. Similar solvers can be found in other softwares. BBsolve is an off-the-shelf function that implements a spectral algorithm with some extra safeguards. It is the same function that is referred to as the "spectral solver" below.

[12]Estimated zeros and ones are replaced with $10^{-10}$ and $1 - 10^{-10}$, respectively.

[13]Our results in the experiments in Appendix C.3 show that using multiple starting values is necessary for the relative good performance of the spectral algorithm and solver when they are applied to data generating processes featuring multiple fixed points, such as our Experiments II, but not for our Experiments I. In that sense, the need for multiple starting values has a similar justification as when maximizing a log-likelihood function that has multiple local maxima.

The spectral approach was implemented in two different ways. First, we consider a simple version using the spectral step length proposed by La Cruz, Martinez, and Raydan (2006), without their nonmonotone line search. We refer to this version as the "spectral algorithm." It was used to reduce the computational burden when comparing the sequence of estimates generated by the spectral approach and by the NPL fixed-point algorithm. Second, spectral estimates are obtained using R's BBsolve function (Varadhan and Gilbert (2009)) which incorporates the nonmonotone line search along with other safeguards. This version is referred to as the "spectral solver." In each model specification (very stable, mildly stable, mildly unstable, and very unstable) and each sample size considered, the BBsolve algorithm only failed to converge in at most 0.4% Monte Carlo samples. Interestingly, the estimates obtained from the simpler version of the spectral approach (i.e., the spectral algorithm) always coincided with the more involved implementation (i.e., the spectral solver). More precisely, the $L_\infty$ distance between the two vector of estimates is always smaller than $10^{-2}$.

Finally, when implementing the relaxation algorithm, we follow Kasahara and Shimotsu's (2012) advice to use $\alpha \approx 0$ and we fix $\alpha = 0.05$.

### 6.1.1 *Results*

*Convergence to the NPL estimator* We first assess whether, upon convergence, the algorithms deliver the NPL estimator. For each Monte Carlo sample, we compute the fixed points (i.e., upon convergence) that maximizes the log-likelihood function for each algorithm. We call the NPL estimator the estimates that correspond to the fixed point that maximizes the log-likelihood across all estimators. Then we compare the estimates obtained from each algorithm with this NPL estimator. We consider that an algorithm has reached the NPL estimator if the difference in absolute value for each parameter is not larger than $10^{-2}$. The results are reported in Table 1. We find that whenever the spectral solver converges (which happens in at least 99.6% of the samples in each experiment) it delivers the NPL estimator. Furthermore, whenever the NPL fixed-point algorithm converges (which happens almost always in the very stable case, but almost never in the very unstable case), it delivers the NPL estimator. The relaxation algorithm never finds the NPL estimator within $K = 100$. Of course, one caveat of defining the NPL estimator as

TABLE 1. Convergence to the NPL estimator—Experiments I.

| | Very Stable | | Mildly Stable | | Mildly Unstable | | Very Unstable | |
|---|---|---|---|---|---|---|---|---|
| | 400 | 5K | 400 | 5K | 400 | 5K | 400 | 5K |
| % NPL fixed-point algorithm | 92.2 | 100.0 | 58.0 | 71.8 | 43.2 | 26.6 | 2.2 | 0.0 |
| % Relaxation algorithm | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| % Spectral algorithm | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| % Spectral solver | 100.0 | 99.8 | 100.0 | 99.6 | 99.8 | 99.8 | 99.6 | 99.6 |

*Note*: Percentages computed over 500 Monte Carlo samples. For each sample, the NPL estimate corresponds to the fixed point of the sample NPL mapping that maximizes the log-likelihood function among all fixed points found by the fixed-point algorithm, the relaxation algorithm, the spectral algorithm, and the spectral solver. An estimator is deemed converging to the NPL estimator if the $L_\infty$ distance between its estimates and the NPL estimates is not greater than $10^{-2}$.

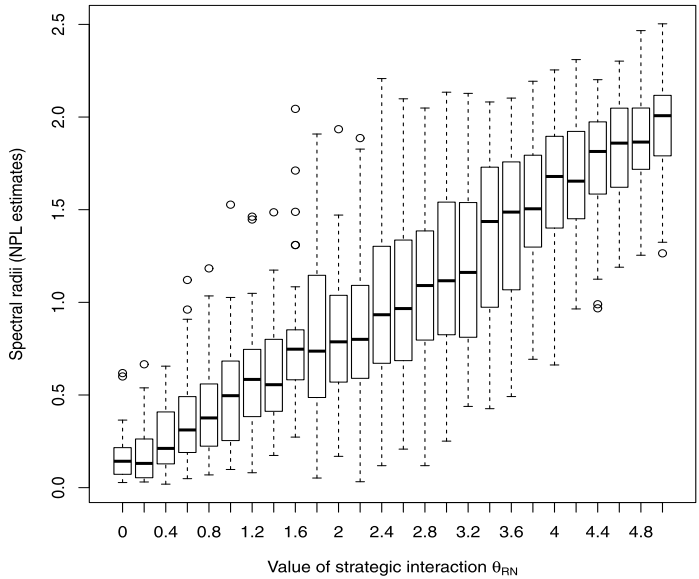



FIGURE 2. Boxplots of sample spectral radii at NPL estimates—$M = 400$.

we do is that we cannot rule out the case where none of the estimators we consider actually finds the NPL fixed point that truly maximizes the log-likelihood function. However, the nice properties of the spectral solver estimator that will be presented below suggest that the spectral algorithm succeeds in finding the NPL estimator in our simulation experiments.

*Empirical distribution of* $\hat{\rho}_{\text{NPL}}$ Figure 2 illustrates how the distribution of $\hat{\rho}_{\text{NPL}}$ varies with $\rho^0$. For each value $\rho^0$ over a grid, we draw boxplots of $\hat{\rho}_{\text{NPL}}$ from 50 Monte Carlo samples of $M = 400$ markets.[14] The results are consistent with Lemma 2. As expected, $\hat{\rho}_{\text{NPL}} < 1$ is a more likely event in more stable data generating processes. However, a large fraction of the sample spectral radii can be on both sides of 1 when the data generating process is close to the stability cut-off.

*Convergence rates* Table 2 compares the algorithms' convergence rates. We consider a sequence to have converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k \leq 100$. Table 2 also reports the fractions of Monte Carlo samples for which the spectral radius evaluated at the NPL estimates is smaller than 1.[15] The results support that the convergence properties of the NPL fixed-point algorithm are driven by the spectral radius of the sample NPL mapping evaluated at the NPL estimator. In fact, the fraction of samples for which this algorithm converges and the share of $\hat{\rho}_{\text{NPL}}$ being smaller than 1 both increase when the data generating process is further from being unstable.

We find that the relaxation algorithm never converges within $K = 100$ iterations. This disappointing performance of the relaxation algorithm in the current setting is likely due

[14]For $\theta_{\text{RN}} = 1.4$, there was a value of $\hat{\rho}_{\text{NPL}}$ which was close to 6. We dropped this outlier when drawing Figure 2 to make the figure easier to read.

[15]Given our results in Table 1, the spectral solver estimates are used as $\hat{\mathbf{P}}_{\text{NPL}}$.

TABLE 2. Sample stability and convergence—Experiments I.

| | Very Stable | | Mildly Stable | | Mildly Unstable | | Very Unstable | |
|---|---|---|---|---|---|---|---|---|
| | 400 | 5K | 400 | 5K | 400 | 5K | 400 | 5K |
| Convergence rates | | | | | | | | |
| % NPL fixed-point algorithm | 92.2 | 99.8 | 58.0 | 71.6 | 43.2 | 26.6 | 2.2 | 0.0 |
| % Relaxation algorithm | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| % Spectral algorithm | 100.0 | 99.8 | 100.0 | 99.6 | 99.8 | 99.8 | 99.6 | 99.6 |
| % Spectral solver | 100.0 | 99.8 | 100.0 | 99.6 | 99.8 | 99.8 | 99.6 | 99.6 |
| Spectral radius <1 | 95.8 | 99.6 | 64.2 | 88.0 | 50.0 | 47.0 | 4.6 | 0.0 |

*Note*: Percentages computed over 500 Monte Carlo samples. "Spectral radius" refers to $\hat{\rho}_{\text{NPL}}$, which is computed using the spectral solver estimates. The NPL, relaxation, and spectral algorithms are deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k \leq 100$.

to the fact that—following Kasahara and Shimotsu's (2012) advice—we use $\alpha \approx 0$ instead of using its optimal value. However, it is worth noting that computing the optimal $\alpha$ would require knowing $\mathbf{P}^0$ and $\boldsymbol{\theta}^0$ therefore making this approach infeasible in practice.

An important feature of the spectral algorithm and the spectral solver that is observable from Table 2 is that they almost always converge, regardless of the data generating process and the value of the sample spectral radius. This observation confirms the importance of separating the properties of the NPL estimator itself from the properties of the NPL fixed-point algorithm. The definition of the NPL estimator does not depend on the stability of the sample NPL fixed point: It exists even if the NPL algorithm may fail to deliver it.

In Proposition 1, we highlighted two potential sources of discrepancy between the asymptotic properties of the FP and the NPL estimators. On one hand, Proposition 1 implies that the NPL algorithm could converge to an inconsistent estimator when $\hat{\rho}_{\text{NPL}} \geq 1$. This would be the case if there exist some stable fixed points of the sample NPL mapping in addition to the unstable NPL fixed point defining the NPL estimator. According to Table 2, the NPL algorithm almost never converges when $\hat{\rho}_{\text{NPL}} \geq 1$. It follows that convergence to an inconsistent estimator is unlikely to be a convincing explanation of the properties of the NPL algorithm estimates in our experiments. On the other hand, Proposition 1 suggests that the converged NPL fixed-point algorithm sequences represent a selected subset of sequences that may have properties that differ from those of the NPL estimator. Table 2 shows that the NPL algorithm does not converge in a nonnegligible share of samples even when the data generating process is stable—28.4% in the mildly stable case with $M = 5000$—and it does converge in a nonnegligible proportion of samples when the population is unstable—26.6% in the mildly unstable case with $M = 5000$. It seems that convergence selection is a suitable candidate to explain how the FP estimator may differ from the consistent NPL estimator in Experiments I.

*Average estimates and standard errors* Table 3 reports average estimates and standard errors of several estimators of interest for $M = 400$. The results for $M = 5000$ are reported in Table C3 in Appendix C.2. First, we report two-step estimates (i.e., $\hat{\boldsymbol{\theta}}_1$) obtained using the simple frequency count estimators for the CCPs. The results suggest that this estimator is heavily biased, even with the larger sample size of $M = 5000$. This observation is

TABLE 3. Simulation results—$M = 400$, Experiments I.

| | $\theta_{RS} = 1$ | $\theta_{RN}$ | $\theta_{EC} = 1$ | $\theta_{FC,1} = 1.9$ | $\theta_{FC,2} = 1.8$ | $\theta_{FC,3} = 1.7$ | $\theta_{FC,4} = 1.6$ | $\theta_{FC,5} = 1.5$ |
|---|---|---|---|---|---|---|---|---|
| | | | | *Two-step estimates* | | | | |
| Very stable | 0.4338 | 0.1674 | 1.2006 | 1.1414 | 1.1069 | 1.0617 | 1.0195 | 0.9905 |
| ($\theta_{RN} = 1$) | (0.1241) | (0.3510) | (0.1316) | (0.2255) | (0.2281) | (0.2053) | (0.2067) | (0.2035) |
| Mildly stable | 0.3843 | 0.1659 | 1.1645 | 1.3746 | 1.3115 | 1.2473 | 1.1799 | 1.1110 |
| ($\theta_{RN} = 2$) | (0.1041) | (0.3349) | (0.1214) | (0.2290) | (0.2080) | (0.2096) | (0.2000) | (0.1960) |
| Mildly unstable | 0.3694 | 0.1709 | 1.1988 | 1.4514 | 1.3669 | 1.2919 | 1.2131 | 1.1301 |
| ($\theta_{RN} = 2.4$) | (0.0980) | (0.3373) | (0.1230) | (0.2257) | (0.2166) | (0.2097) | (0.1914) | (0.1869) |
| Very unstable | 0.3457 | 0.3327 | 1.4784 | 1.7093 | 1.5856 | 1.4510 | 1.2712 | 0.9046 |
| ($\theta_{RN} = 4$) | (0.0917) | (0.4230) | (0.1538) | (0.2978) | (0.2738) | (0.2581) | (0.2327) | (0.2036) |
| | | | | *Converged $K = 100$ NPL fixed-point algorithm estimates* | | | | |
| Very stable | 1.0002 | 0.9782 | 1.0048 | 1.9194 | 1.8349 | 1.7250 | 1.6165 | 1.5238 |
| ($\theta_{RN} = 1$) | (0.1916) | (0.5822) | (0.1158) | (0.2249) | (0.2222) | (0.2101) | (0.1994) | (0.1935) |
| Mildly stable | 0.8324 | 1.3676 | 1.0535 | 1.9132 | 1.8119 | 1.6990 | 1.5851 | 1.4726 |
| ($\theta_{RN} = 2$) | (0.1489) | (0.5223) | (0.1101) | (0.2090) | (0.1934) | (0.1897) | (0.1931) | (0.1874) |
| Mildly unstable | 0.7653 | 1.4513 | 1.0985 | 1.9035 | 1.7862 | 1.6705 | 1.5493 | 1.4259 |
| ($\theta_{RN} = 2.4$) | (0.1226) | (0.4603) | (0.1121) | (0.2107) | (0.1923) | (0.1842) | (0.1859) | (0.1873) |
| Very unstable | 0.6151 | 1.7496 | 1.3533 | 1.9620 | 1.8595 | 1.6507 | 1.4566 | 1.1281 |
| ($\theta_{RN} = 4$) | (0.0921) | (0.2360) | (0.0796) | (0.2831) | (0.3244) | (0.3112) | (0.3031) | (0.2909) |
| | | | | *All $K = 100$ NPL fixed-point algorithm estimates* | | | | |
| Very stable | 1.0355 | 1.0882 | 0.9990 | 1.9058 | 1.8206 | 1.7124 | 1.6076 | 1.5156 |
| ($\theta_{RN} = 1$) | (0.2217) | (0.6761) | (0.1177) | (0.2304) | (0.2313) | (0.2181) | (0.2055) | (0.1998) |
| Mildly stable | 0.9603 | 1.8406 | 1.0181 | 1.9082 | 1.8046 | 1.7043 | 1.6004 | 1.4957 |
| ($\theta_{RN} = 2$) | (0.1953) | (0.6930) | (0.1098) | (0.2218) | (0.2130) | (0.2062) | (0.2069) | (0.2059) |
| Mildly unstable | 0.9154 | 2.0475 | 1.0476 | 1.8998 | 1.7899 | 1.6834 | 1.5802 | 1.4721 |
| ($\theta_{RN} = 2.4$) | (0.1640) | (0.6134) | (0.1133) | (0.2291) | (0.2165) | (0.2100) | (0.2077) | (0.2083) |
| Very unstable | 0.7892 | 2.7388 | 1.2489 | 1.9232 | 1.8002 | 1.6744 | 1.5110 | 1.2762 |
| ($\theta_{RN} = 4$) | (0.0869) | (0.2833) | (0.1105) | (0.2769) | (0.2620) | (0.2491) | (0.2426) | (0.2311) |
| | | | | *All $K = 100$ relaxation algorithm estimates* | | | | |
| Very stable | 1.5330 | 2.5713 | 0.8939 | 1.8395 | 1.7690 | 1.6830 | 1.6012 | 1.5350 |
| ($\theta_{RN} = 1$) | (0.6963) | (2.0664) | (0.2138) | (0.2818) | (0.2771) | (0.2641) | (0.2527) | (0.2513) |
| Mildly stable | 1.9311 | 5.2818 | 0.6627 | 2.1403 | 2.0882 | 2.0427 | 2.0125 | 1.9916 |
| ($\theta_{RN} = 2$) | (0.6812) | (2.3969) | (0.2940) | (0.3894) | (0.3982) | (0.4195) | (0.4504) | (0.4842) |
| Mildly unstable | 1.8782 | 5.7878 | 0.6053 | 2.1786 | 2.1297 | 2.0985 | 2.0833 | 2.0810 |
| ($\theta_{RN} = 2.4$) | (0.5786) | (2.2225) | (0.3066) | (0.3996) | (0.4061) | (0.4239) | (0.4537) | (0.4930) |
| Very unstable | 1.2755 | 5.5171 | 0.7397 | 1.9935 | 1.9106 | 1.8474 | 1.8310 | 1.9230 |
| ($\theta_{RN} = 4$) | (0.1709) | (0.7769) | (0.1883) | (0.3489) | (0.3337) | (0.3309) | (0.3342) | (0.3608) |
| | | | | *All $K = 100$ spectral algorithm estimates* | | | | |
| Very stable | 1.0416 | 1.1071 | 0.9973 | 1.9043 | 1.8192 | 1.7113 | 1.6064 | 1.5148 |
| ($\theta_{RN} = 1$) | (0.2377) | (0.7264) | (0.1204) | (0.2328) | (0.2337) | (0.2205) | (0.2075) | (0.2014) |
| Mildly stable | 1.0308 | 2.0941 | 0.9869 | 1.9209 | 1.8212 | 1.7240 | 1.6244 | 1.5258 |
| ($\theta_{RN} = 2$) | (0.2922) | (1.0461) | (0.1383) | (0.2300) | (0.2220) | (0.2189) | (0.2232) | (0.2269) |
| Mildly unstable | 1.0135 | 2.4333 | 0.9951 | 1.9230 | 1.8200 | 1.7192 | 1.6218 | 1.5230 |
| ($\theta_{RN} = 2.4$) | (0.2722) | (1.0528) | (0.1509) | (0.2416) | (0.2318) | (0.2289) | (0.2336) | (0.2429) |
| Very unstable | 0.9818 | 3.8546 | 1.0316 | 1.9337 | 1.8250 | 1.7229 | 1.6106 | 1.4913 |
| ($\theta_{RN} = 4$) | (0.1591) | (0.7949) | (0.1793) | (0.3046) | (0.2882) | (0.2769) | (0.2762) | (0.2989) |

(*Continues*)

TABLE 3. *Continued.*

| | $\theta_{RS}=1$ | $\theta_{RN}$ | $\theta_{EC}=1$ | $\theta_{FC,1}=1.9$ | $\theta_{FC,2}=1.8$ | $\theta_{FC,3}=1.7$ | $\theta_{FC,4}=1.6$ | $\theta_{FC,5}=1.5$ |
|---|---|---|---|---|---|---|---|---|
| | | | | *Spectral solver estimates* | | | | |
| Very stable | 1.0416 | 1.1071 | 0.9973 | 1.9043 | 1.8192 | 1.7113 | 1.6064 | 1.5148 |
| ($\theta_{RN}=1$) | (0.2377) | (0.7264) | (0.1204) | (0.2328) | (0.2337) | (0.2205) | (0.2075) | (0.2014) |
| Mildly Stable | 1.0308 | 2.0941 | 0.9869 | 1.9209 | 1.8212 | 1.7240 | 1.6244 | 1.5258 |
| ($\theta_{RN}=2$) | (0.2922) | (1.0461) | (0.1383) | (0.2300) | (0.2220) | (0.2189) | (0.2232) | (0.2269) |
| Mildly Unstable | 1.0135 | 2.4331 | 0.9953 | 1.9228 | 1.8199 | 1.7190 | 1.6217 | 1.5231 |
| ($\theta_{RN}=2.4$) | (0.2725) | (1.0539) | (0.1510) | (0.2418) | (0.2320) | (0.2291) | (0.2338) | (0.2432) |
| Very unstable | 0.9813 | 3.8541 | 1.0320 | 1.9315 | 1.8235 | 1.7213 | 1.6089 | 1.4898 |
| ($\theta_{RN}=4$) | (0.1592) | (0.7964) | (0.1796) | (0.3031) | (0.2878) | (0.2763) | (0.2754) | (0.2987) |

*Note*: Averages and standard errors (in brackets) computed over 500 Monte Carlo samples. The $K=100$ NPL algorithm is deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for some $k \leq 100$. The relaxation algorithm never converged by $K=100$. Since the spectral algorithm almost always converged, the results conditional on convergence are very similar to the ones obtained from all samples and are not reported.

not surprising given that there is a small number of markets in each realized state therefore making the estimated CCPs very imprecise. This poor performance of the two-step estimator motivates imposing equilibrium restrictions in the estimation.

We then report the results for the NPL fixed-point algorithm. We consider two cases. First, we focus only on the estimates obtained from the converged NPL algorithm, where a sequence is deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k \leq 100$. Second, we include all estimates at $K=100$, regardless whether the sequence of estimates has converged or not.

We find that restricting the NPL algorithm's estimates to converged sequences can introduce a substantial bias, especially when the DGP is unstable. The results are more encouraging for the converged NPL algorithm estimates in the "very stable" data generating process. By comparing the results from the "very stable" and the "mildly stable" cases, it is obvious that the properties of the converged NPL algorithm estimator vary within the category of stable data generating processes, with more stable equilibria being more suitable for this estimator. While this insight is quite intuitive, the current simulation exercise seems to be the first one to make this point.

Interestingly, except in the "very stable" case, one typically improves upon the properties of the NPL algorithm estimator by considering any sequence of estimates, regardless whether it converged, instead of focusing on converged ones. We are not aware of existing work in the literature that has made this comparison. This feature of the NPL algorithm is related to the convergence selection exposed in Proposition 1.

Since the relaxation algorithm has been proposed as a modified NPL algorithm, we also include the results obtained from this estimator (with $\alpha=0.05$). As already mentioned above, no sequence of estimates has converged by $K=100$. When looking at the last iteration estimates, we also find that they are heavily biased and very imprecise.

Finally, we report estimates obtained from the spectral algorithm and the spectral solver. Since these two algorithms feature convergence frequencies close to 100%, we do not include separate results focusing only on the converging sequences of estimates.

For most cases, the spectral approach has nice properties as one expects from the frequency at which this approach delivers the NPL estimator. In particular, it has much better properties than the NPL algorithm estimator when the equilibrium generating the data is not stable. This observation is coherent with the fact that the NPL estimator should be consistent even if stability of the sample NPL mapping fails at the NPL estimates.

*Distribution of estimates* We now compare the empirical distributions of estimates from the converged NPL fixed point and the spectral solver algorithms. Since the strategic interaction parameter $\theta_{\text{RN}}$ is the most problematic one according to the results presented in Tables 3 and C3, we focus our attention on this parameter.[16]

Figure 3 compares the histograms of $\hat{\theta}_{\text{RN}} - \theta^0_{\text{RN}}$ obtained from the converged NPL algorithm estimates and the spectral solver's estimates. While the distributions are practically identical for the "very stable" case, sizeable differences appear as the data generating process becomes less stable. Sample size does affect the level of similitude between distributions. While the distributions are still quite similar in the "mildly stable" data generating process with $M = 5000$, they are already quite different with $M = 400$.

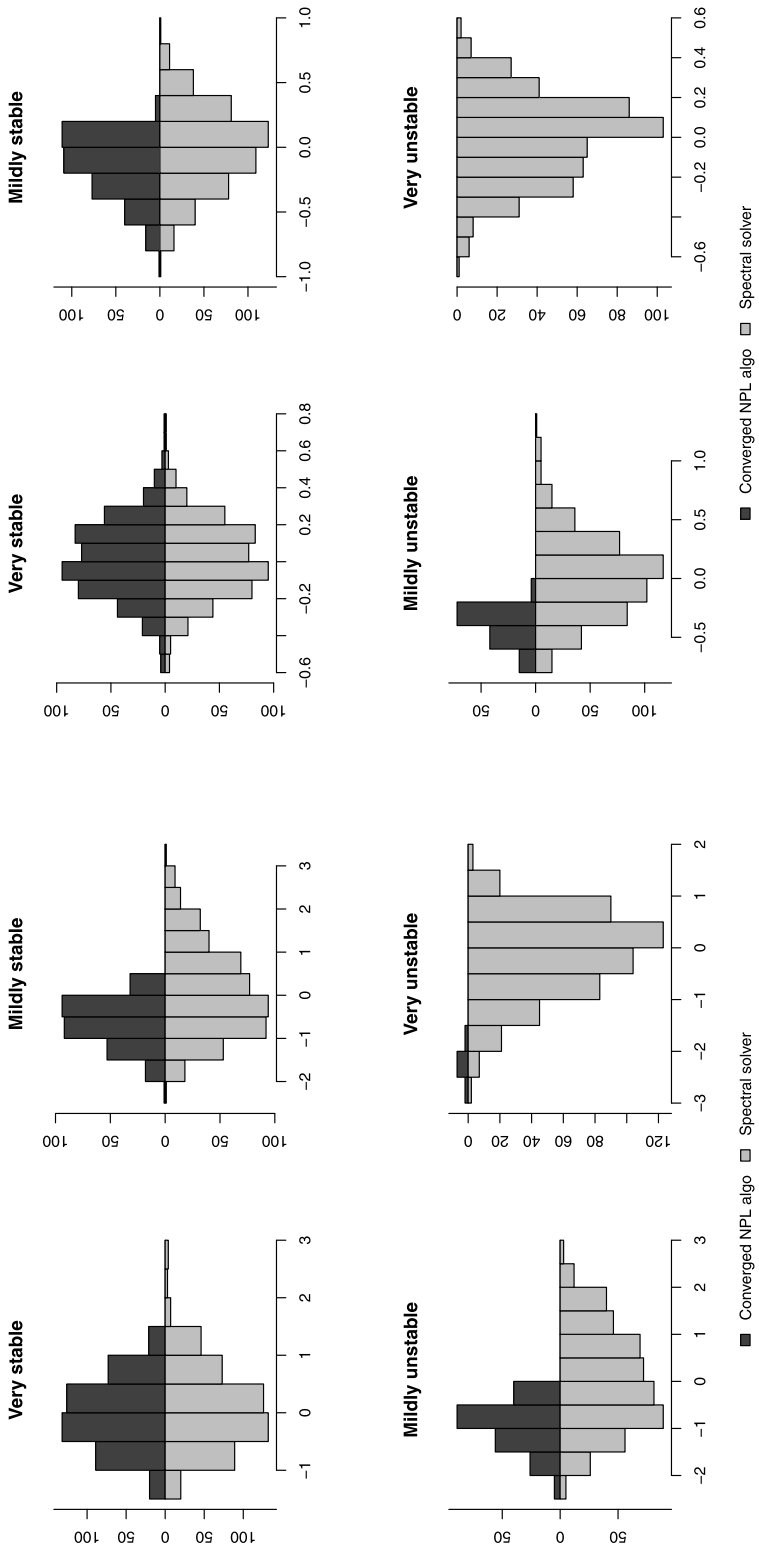
The distribution of the spectral approach estimator is roughly normal (perhaps a bit skewed in smaller samples) and centered close to the true value of the parameter. This observation is coherent with the properties of the NPL estimator. However, the distributions of the converged NPL algorithm estimates tend to be shifted away from the true value in more unstable cases and look like truncated normals. Such a truncation is consistent with Proposition 1. The direction of the truncation is also as expected. Converging sequences of the NPL algorithm tend to be associated with smaller values of the estimated strategic interaction parameter. A similar finding was noted by Mogensen (2015). This pattern is not surprising given the spectral radii reported in Figure 1. It also offers an obvious explanation for the direction of the bias associated with the NPL algorithm estimator restricted to converging sequences: Since converging sequences typically lead to smaller values of estimated strategic interaction parameters, the convergence selection implies an attenuation bias.

Histograms in Figure C1, in Appendix C.4, split the distributions of spectral solver estimates in two groups: With $\hat{\rho}_{\text{NPL}} < 1$, and with $\hat{\rho}_{\text{NPL}} \geq 1$. It emphasizes that the NPL estimator exists regardless the stability of the sample NPL fixed point defining the NPL estimator. As predicted by Proposition 1, there are noticeable truncations in the distributions conditional on $\hat{\rho}_{\text{NPL}} < 1$.

*Computational cost* The results obtained from the spectral algorithm and the spectral solver are quite encouraging. They suggest that the spectral approach delivers the NPL estimator, which has nice properties, even if the data generating process does not satisfy desirable stability conditions. One important question remains: how costly is it compared with other methods?[17]

[16]Similar distributions for all other parameters are available from the authors upon request.

[17]All the Monte Carlo experiments have been implemented using R programming language in a Dell Precision T7500 Workstation, with Dual Six Core Intel Xeon Processor X5680 3.33 GHz, 48 Gb of RAM (1333 MHz), and Windows 10 Professional (64 bit) Operating System.




(a) $M = 400$ (b) $M{=}5000$

FIGURE 3. Histograms of $\hat{\theta}_{\text{RN}} - \theta^0_{\text{RN}}$ (Converged NPL algorithm vs. spectral solver)—Experiments I.

Table 4 summarizes each algorithm's computational burden per starting value.[18] When comparing the time needed to perform a single iteration, we find that the NPL fixed point, the relaxation, and the spectral algorithms are quite similar, with the spectral algorithm being slightly faster. However, both the average number of iterations needed to reach convergence and its standard deviation are much smaller for the spectral algorithm than for the NPL algorithm.[19] As a result, the total time until convergence is on average much smaller for the spectral algorithm than for the NPL algorithm and we conclude that the former is computationally cheaper than the latter. Even if the computational cost of the spectral algorithm increases for unstable data generating processes, it remains cheaper to implement than the NPL algorithm.

We also report the total time required by the spectral solver to converge. Since we are directly implementing R's BBsolve function from the BB package, we did not compute the computational time needed to perform a single iteration. Nonetheless, the total time needed to reach convergence is reported in Table 4. Since the spectral solver adds a nonmonotone search and other safeguards, it does not converge as fast as the manually coded spectral algorithm, but it is still cheaper than the NPL algorithm.

*Fixed-point multiplicity* For these experiments, we did not find evidence of multiple fixed points in the sample NPL mappings. Upon convergence, each sequence initiated at five randomly chosen starting values lead to the same fixed point. Once again, this observation supports that the disappointing performance of the NPL algorithm for some data generating processes is explained by the failure to converge rather than converging to an inconsistent NPL fixed point.

In order to detect as many multiple fixed points in the sample NPL mapping as possible, we reran a smaller-scale Monte Carlo simulation (25 samples) in which the algorithms were initiated at a larger number of starting values (25 instead of 5) and a larger number of iterations ($K = 500$ instead of 100) was allowed for. As before, the NPL, the relaxation and the spectral algorithms are deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k \leq 500$. For the spectral solver, R's BBsolve default tolerance (i.e., the $L_2$ norm of $\hat{\boldsymbol{\phi}}(\mathbf{P})$ being smaller than $\sqrt{\dim(\mathbf{P})} \times 10^{-7}$) is used. After having identified all fixed points across the 25 different starting values (the first set being the frequency count estimator of the CCPs and the 24 remaining ones being randomly chosen), we measure the distance between two fixed points as the maximum absolute value of the difference between the vectors of fixed point CCPs, that is, the $L_\infty$ distance. Two fixed points are deemed different if the $L_\infty$ distance between them is greater than $10^{-5}$. The average number of different fixed points are computed across Monte Carlo samples and reported in Table 5. This table also shows the maximum distance between the fixed point maximizing the log-likelihood function and all the other fixed points found across different starting values and across Monte Carlo samples.

[18] In Table 4, the reported averages and standard deviations of the number of iterations, the time per iteration, and the time until convergence are computed using only the starting value that leads to the highest log-likelihood among all the starting values that are used.

[19] The number of iterations to reach convergence is not available for the relaxation algorithm because the algorithm failed to converge within $K = 100$.

TABLE 4. Computational cost—Experiments I.

| | Very Stable | | Mildly Stable | | Mildly Unstable | | Very Unstable | |
|---|---|---|---|---|---|---|---|---|
| | 400 | 5K | 400 | 5K | 400 | 5K | 400 | 5K |
| | | | | *NPL fixed-point algorithm* | | | | |
| Number of iterations | 16.11 | 12.63 | 25.64 | 41.86 | 27.30 | 56.50 | 33.91 | – |
| | (13.55) | (2.80) | (18.85) | (19.12) | (19.11) | (21.62) | (15.23) | (–) |
| Time per iteration | | | | | | | | |
| Full iteration | 1.4209 | 2.1368 | 1.7561 | 2.7194 | 1.8554 | 2.7578 | 1.8225 | 2.8106 |
| | (0.1806) | (0.2888) | (0.1899) | (0.3321) | (0.1692) | (0.2617) | (0.1199) | (0.1824) |
| Update parameters | 1.4179 | 2.1340 | 1.7530 | 2.7163 | 1.8522 | 2.7548 | 1.8194 | 2.8077 |
| | (0.1807) | (0.2889) | (0.1899) | (0.3320) | (0.1692) | (0.2617) | (0.1199) | (0.1823) |
| Update CCPs | 0.0030 | 0.0028 | 0.0030 | 0.0030 | 0.0032 | 0.0030 | 0.0030 | 0.0028 |
| | (0.0019) | (0.0017) | (0.0012) | (0.0009) | (0.0010) | (0.0007) | (0.0006) | (0.0006) |
| Time until convergence | 23.287 | 27.041 | 44.902 | 114.15 | 50.951 | 154.16 | 61.906 | – |
| | (20.968) | (7.273) | (33.934) | (55.479) | (37.137) | (61.865) | (27.129) | (–) |
| | | | | *Relaxation algorithm* | | | | |
| Number of iterations | – | – | – | – | – | – | – | – |
| | (–) | (–) | (–) | (–) | (–) | (–) | (–) | (–) |
| Time per iteration | | | | | | | | |
| Full iteration | 1.6361 | 2.2188 | 1.8498 | 2.7212 | 1.8894 | 2.7805 | 1.8764 | 2.8376 |
| | (0.1186) | (0.1657) | (0.0977) | (0.2671) | (0.0893) | (0.1876) | (0.0724) | (0.1653) |
| Update parameters | 1.6329 | 2.2158 | 1.8466 | 2.7179 | 1.8860 | 2.7773 | 1.8732 | 2.8345 |
| | (0.1186) | (0.1657) | (0.0976) | (0.2671) | (0.1893) | (0.1875) | (0.0724) | (0.1653) |
| Update CCPs | 0.0032 | 0.0030 | 0.0032 | 0.0032 | 0.0033 | 0.0031 | 0.0032 | 0.0030 |
| | (0.0006) | (0.0006) | (0.0006) | (0.0006) | (0.0006) | (0.0006) | (0.0006) | (0.0006) |
| Time until convergence | – | – | – | – | – | – | – | – |
| | (–) | (–) | (–) | (–) | (–) | (–) | (–) | (–) |
| | | | | *Spectral algorithm* | | | | |
| Number of iterations | 9.99 | 9.06 | 13.52 | 11.89 | 14.68 | 12.94 | 23.11 | 22.32 |
| | (2.23) | (0.64) | (4.41) | (1.07) | (4.72) | (1.46) | (6.97) | (2.21) |
| Time per iteration | | | | | | | | |
| Full iteration | 1.4167 | 2.0886 | 1.7171 | 2.6572 | 1.8416 | 2.7616 | 1.8349 | 2.8164 |
| | (0.1713) | (0.2444) | (0.1741) | (0.2906) | (0.1499) | (0.2181) | (0.1278) | (0.2100) |
| Update parameters | 1.4134 | 2.0856 | 1.7140 | 2.6538 | 1.8383 | 2.7583 | 1.8317 | 2.8133 |
| | (0.1712) | (0.2443) | (0.1742) | (0.2905) | (0.1500) | (0.2181) | (0.1279) | (0.2100) |
| Update CCPs | 0.0033 | 0.0030 | 0.0031 | 0.0034 | 0.0033 | 0.0033 | 0.0031 | 0.0030 |
| | (0.0021) | (0.0021) | (0.0017) | (0.0020) | (0.0017) | (0.0017) | (0.0013) | (0.0013) |
| Time until convergence | 14.253 | 18.924 | 23.329 | 31.613 | 27.164 | 35.737 | 42.387 | 62.843 |
| | (4.0135) | (2.628) | (8.349) | (4.588) | (9.328) | (5.043) | (13.127) | (7.594) |
| | | | | *Spectral solver* | | | | |
| Time until convergence | 20.578 | 24.300 | 36.915 | 43.935 | 40.640 | 48.963 | 58.776 | 82.576 |
| | (14.193) | (4.071) | (29.897) | (24.211) | (28.001) | (6.873) | (18.539) | (9.014) |

*Note*: Number of iterations needed to converge (conditional on convergence) and computational time in seconds. Averages and standard deviations (in brackets) computed over 500 Monte Carlo samples. The NPL, relaxation, and spectral algorithms are deemed having converged at $k$th iteration if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$. Time to perform a full iteration is divided in two parts: updating the parameters $\boldsymbol{\theta}$ given the CCPs (a pseudo-maximum likelihood problem) and updating the CCPs given the parameters (using the algorithm-specific rule). For the spectral solver, the total computation time to reach convergence (with R's BBsolve default tolerance, i.e., the $L_2$ norm of $\hat{\boldsymbol{\phi}}(\mathbf{P})$ being smaller than $\sqrt{\dim(\mathbf{P})} \times 10^{-7}$) is reported.

TABLE 5. Fixed-point multiplicity/uniqueness in sample NPL mapping.[1] Experiments I.

| | Very Stable | | Mildly Stable | | Mildly Unstable | | Very Unstable | |
|---|---|---|---|---|---|---|---|---|
| | 400 | 5K | 400 | 5K | 400 | 5K | 400 | 5K |
| | | | | *NPL fixed-point algorithm*[3] | | | | |
| Avg # of fixed points[2] | 1.00 | 1.00 | 1.00 | 1.05 | 1.00 | 1.00 | 1.00 | – |
| Max distance[5] | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | – |
| | | | | *Relaxation algorithm*[3] | | | | |
| Avg # of fixed points | 25.00 | 25.00 | 24.96 | 25.00 | 24.83 | 25.00 | 20.96 | 24.96 |
| Max distance | 0.0009 | 0.0004 | 0.0015 | 0.0006 | 0.0017 | 0.0007 | 0.0020 | 0.0019 |
| | | | | *Spectral algorithm*[3] | | | | |
| Avg # of fixed points | 3.08 | 1.20 | 4.16 | 3.68 | 6.24 | 5.92 | 10.04 | 8.04 |
| Max distance | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0004 | 0.0000 | 0.0005 | 0.0001 |
| | | | | *Spectral solver*[4] | | | | |
| Avg # of fixed points | 1.00 | 1.00 | 1.00 | 1.04 | 1.00 | 1.00 | 1.000 | 1.00 |
| Max distance | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

*Note*: (1) In an attempt to detect multiple fixed points in the sample NPL mapping, the algorithms are initiated at a larger number of starting values (25 instead of 5) and a larger number of iterations ($K = 500$ instead of 100) is allowed for. (2) Averages and maxima computed over 25 Monte Carlo samples. (3) The NPL, relaxation, and spectral algorithms are deemed having converged to a fixed point if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k \leq 500$. (4) For the spectral solver, R's BBsolve default tolerance (i.e., the $L_2$ norm of $\hat{\boldsymbol{\phi}}(\mathbf{P})$ being smaller than $\sqrt{\dim(\mathbf{P})} \times 10^{-7}$) is used. (5) The distance between two fixed points is measured by the maximum absolute value of the difference between the vectors of fixed point CCPs. Two fixed points are deemed different if their distance is greater than $10^{-5}$. The reported maximum distance is the largest distance between the fixed point maximizing the likelihood function and all the other fixed points found across different starting values and across Monte Carlo samples.

The results in Table 5 show that for Experiments I, while multiple fixed points may be found, they are all numerically very similar. For most Monte Carlo samples, the NPL algorithm and the spectral solver find a single fixed point across the 25 different starting values. The relaxation and the spectral algorithms do find multiple fixed points. However, for all algorithms, $L_\infty$ distance between the vectors of fixed point is always smaller than $10^{-2}$, and for most data generating processes it is smaller than $10^{-5}$.[20]

### 6.2 *Experiments II: Pesendorfer and Schmidt-Dengler (2008)*

The data generating processes based on Aguirregabiria and Mira (2007) have shown that the spectral approach is an appealing alternative algorithm to obtain the NPL estimator regardless of the stability of the data generating process. For those data generating processes, convergence selection is the only explanation for the bad performance of the NPL algorithm when the data are generated from an unstable equilibrium. In this second part of our Monte Carlo experiments, we compare the algorithms' performance using a different dynamic game of market entry and exit proposed by Pesendorfer and Schmidt-Dengler (2008). This alternative data generating process has been used in the

[20]A result from the experiments presented in Table 5 that is also worth mentioning is that when the relaxation algorithm is allowed to iterate up to $K = 500$ (instead of 100) it almost always converges. We have found that convergence of this algorithm requires more than 200 iterations, which is why in our main experiments with $K = 100$ this algorithm always failed to converge.

literature to show that the NPL algorithm may converge to an inconsistent fixed point of the NPL mapping.

More precisely, we use the same reparameterization as in Dearing and Blevins (2021). There are $N = 2$ players in a dynamic game of market entry and exit. Their current utility $U_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}, \varepsilon_{imt}(y_{imt}))$ is such that

$$\begin{cases} \theta_{\mathrm{M}} + \theta_{\mathrm{C}}\, y_{2-i,mt} + \theta_{\mathrm{EC}}(1 - y_{im,t-1}) + \varepsilon_{imt}(1), & y_{imt} = 1, \\ \theta_{\mathrm{SV}}\, y_{im,t-1} + \varepsilon_{imt}(0), & y_{imt} = 0. \end{cases} \tag{27}$$

The only state variables are players' incumbency status. The total number of possible states is therefore $\dim(\mathcal{X}) = 2^2 = 4$. The values of the parameters in the payoff function are $\theta_{\mathrm{M}} = 1.2$, $\theta_{\mathrm{C}} = -2.4$, $\theta_{\mathrm{EC}} = -0.2$, $\theta_{\mathrm{SV}} = 0.1$, and $\beta = 0.9$, and $\varepsilon_{imt}(0)$, $\varepsilon_{imt}(1)$ are independent and identically distributed Normal$(0, 0.5)$.

Pesendorfer and Schmidt-Dengler (2008, p. 920) described five different equilibria of the game. Initializing a built-in solver for systems of nonlinear equations[21] at 500 randomly picked starting values, we found all five equilibria. In our data generating processes, we consider equilibria referred to as (i), (ii), and (iii) by Pesendorfer and Schmidt-Dengler (2008). Only equilibrium (i) is stable according to Definition 8(A). Table C2 in Appendix C.1 reports summary statistics from the simulated data.

For each of these three equilibria, we take a random sample of $M$ markets from the ergodic distribution of the incumbency statuses and use the equilibrium CCPs to generate players' decisions. We draw 500 Monte Carlo samples of size $M = 100$ and $M = 1000$.

We slightly modify the implementation of the algorithms. Multiple randomly picked starting values must be used in order to find multiple sample NPL fixed points. Since the computational burden of a single estimation is very small, we use 100 different starting values. The first one is the frequency count estimator and the second one is $\hat{\mathbf{P}}_k$ for $k = 1$. The other 98 starting values are randomly chosen. Once again, Appendix C.3 reports a comparison with the case where the frequency count estimates of the CCPs are used as the only set of starting values. Also because a single iteration is relatively cheap to compute, we increase the number of iterations to $K = 500$. This large increase has proven to be useful to appreciate the properties of the relaxation algorithm.

We are still reporting two different algorithms for the spectral approach. While the non-monotone line search and the other safeguards included in the spectral solver seemed to lead to the same estimates as with the manually coded spectral algorithm in Experiments I, they can differ in the current set of data generating processes. This observation is especially true for $M = 100$.

### 6.2.1 *Results*

*Convergence to the NPL estimator* Table 6 shows how often each algorithm succeeds in reaching the NPL estimator. Once again, the spectral solver almost always reaches the sample NPL fixed point that is associated with the largest log-likelihood function. Most algorithms are able to reach the NPL estimator in the stable equilibriun (i.e., equilibrium (i)) especially in large samples. However, the NPL fixed point and the relaxation

[21] Once again, we use R's BBsolve function.

TABLE 6. Convergence to the NPL estimator—Experiments II.

| | Eq (i) | | Eq (ii) | | Eq (iii) | |
|---|---|---|---|---|---|---|
| | 100 | 1K | 100 | 1K | 100 | 1K |
| % NPL algorithm | 88.0 | 100.0 | 32.8 | 1.0 | 28.4 | 0.2 |
| % Relaxation algorithm | 53.6 | 81.8 | 27.6 | 1.8 | 23.4 | 0.8 |
| % Spectral algorithm | 88.2 | 100.0 | 71.0 | 79.4 | 68.6 | 91.0 |
| % Spectral solver | 100.0 | 100.0 | 96.0 | 95.8 | 97.4 | 94.0 |

*Note*: Percentages computed over 500 Monte Carlo samples. For each sample, the NPL estimates correspond to the fixed point of the sample NPL mapping that maximizes the log-likelihood function among all fixed points found by the NPL algorithm, the relaxation algorithm, the spectral algorithm, and the spectral solver. An estimator is deemed converging to the NPL estimator if the maximum of the absolute difference between its estimates and the NPL estimates is at most $10^{-2}$.

algorithms typically converge to another fixed point if the data are generated from an unstable equilibrium. Finally, the nonmonotone line search and the other safeguards implemented with the spectral solver seems to make it more likely to reach the NPL estimator compared to the spectral algorithm.

*Comparing convergence rates* One of the most important differences between the results from the two sets of experiments can be found in Table 7, which reports the convergence rates of the different algorithms: even in unstable data generating processes, the NPL algorithm always converges. This observation suggests that the properties of the NPL algorithm estimates in these data generating processes are likely not affected by the convergence selection problem, but rather by the convergence to an inconsistent NPL fixed point. Another important difference is that the relaxation algorithm now converges in most Monte Carlo samples. As shown below, this convergence usually happens after a large number of iterations.

*Average estimates and standard errors* Table 8 summarizes the simulation results for both sample sizes used. While the two-step estimates are biased and imprecise when $M = 100$, they are more convincing when $M = 1000$. This observation highlights that, while two-step estimation may be useful in some settings, it typically requires a large

TABLE 7. Sample stability and convergence—Experiments II.

| | Eq (i) | | Eq (ii) | | Eq (iii) | |
|---|---|---|---|---|---|---|
| | 100 | 1K | 100 | 1K | 100 | 1K |
| Convergence rates | | | | | | |
| % NPL algorithm | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| % Relaxation algorithm | 55.4 | 81.8 | 92.0 | 98.8 | 93.0 | 99.4 |
| % Spectral algorithm | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| % Spectral solver | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Spectral radius <1 | 88.0 | 100.0 | 33.4 | 1.6 | 28.4 | 0.2 |

*Note*: Percentages computed over 500 Monte Carlo samples. "Spectral radius" refers to $\hat{\rho}_{\text{NPL}}$, which is computed using the spectral solver estimates. The NPL, relaxation, and spectral algorithms are deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for $k = 500$.

TABLE 8. Simulation results—$M = 100$ and $M = 1000$, Experiments II.

| | $M = 100$ | | | | $M = 1000$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $\theta_M = 1.2$ | $\theta_C = -2.4$ | $\theta_{EC} = -0.2$ | $\theta_{SV} = 0.1$ | $\theta_M = 1.2$ | $\theta_C = -2.4$ | $\theta_{EC} = -0.2$ | $\theta_{SV} = 0.1$ |
| | | | | *Two-step estimates* | | | | |
| Eq (i) | 1.0415 | −2.0807 | −0.2661 | 0.0497 | 1.1891 | −2.3733 | −0.2072 | 0.0939 |
| | (0.3188) | (0.6005) | (0.1710) | (0.1537) | (0.1094) | (0.1973) | (0.0572) | (0.0507) |
| Eq (ii) | 0.8957 | −1.7680 | −0.3162 | 0.0143 | 1.1511 | −2.2903 | −0.2200 | 0.0849 |
| | (0.3917) | (0.8408) | (0.2298) | (0.1983) | (0.1478) | (0.3320) | (0.0929) | (0.0799) |
| Eq (iii) | 0.9019 | −1.7850 | −0.3142 | 0.0156 | 1.1532 | −2.2975 | −0.2191 | 0.0856 |
| | (0.3766) | (0.8196) | (0.2314) | (0.2013) | (0.1466) | (0.3278) | (0.0928) | (0.0798) |
| | | | | *All $K = 500$ NPL algorithm estimates* | | | | |
| Eq (i) | 1.1639 | −2.3227 | −0.2358 | 0.0678 | 1.1960 | −2.3910 | −0.2033 | 0.0971 |
| | (0.3054) | (0.4216) | (0.1059) | (0.0812) | (0.0917) | (0.0986) | (0.0237) | (0.0165) |
| Eq (ii) | 0.9534 | −1.6759 | −0.3884 | −0.0637 | 0.9858 | −1.7460 | −0.3576 | −0.036 |
| | (0.2355) | (0.3279) | (0.1124) | (0.0943) | (0.0702) | (0.0769) | (0.0326) | (0.0273) |
| Eq (iii) | 0.9542 | −1.6674 | −0.3952 | −0.0706 | 0.9747 | −1.7180 | −0.3702 | −0.047 |
| | (0.2372) | (0.3139) | (0.1113) | (0.0930) | (0.0685) | (0.0749) | (0.0330) | (0.0278) |
| | | | | *Converged $K = 500$ relaxation algorithm estimates* | | | | |
| Eq (i) | 1.1358 | −2.2504 | −0.2579 | 0.0485 | 1.1956 | −2.3877 | −0.2042 | 0.0962 |
| | (0.3196) | (0.4594) | (0.1123) | (0.0854) | (0.0938) | (0.1019) | (0.0243) | (0.0168) |
| Eq (ii) | 0.9660 | −1.6943 | −0.3828 | −0.0594 | 0.9889 | −1.7520 | −0.3561 | −0.0350 |
| | (0.2369) | (0.3321) | (0.1121) | (0.0939) | (0.0697) | (0.0769) | (0.0326) | (0.0273) |
| Eq (iii) | 0.9623 | −1.6832 | −0.3911 | −0.0674 | 0.9776 | −1.7249 | −0.3685 | −0.0462 |
| | (0.2363) | (0.3144) | (0.1112) | (0.0932) | (0.0689) | (0.0812) | (0.0337) | (0.0286) |
| | | | | *All $K = 500$ relaxation algorithm estimates* | | | | |
| Eq (i) | 1.1645 | −2.3270 | −0.2337 | 0.0699 | 1.1966 | −2.3917 | −0.2031 | 0.0972 |
| | (0.3004) | (0.4064) | (0.1033) | (0.0794) | (0.0920) | (0.0995) | (0.0239) | (0.0167) |
| Eq (ii) | 0.9561 | −1.6831 | −0.3869 | −0.0625 | 0.9883 | −1.7525 | −0.3563 | −0.0351 |
| | (0.2355) | (0.3279) | (0.1124) | (0.0943) | (0.0698) | (0.0770) | (0.0325) | (0.0273) |
| Eq (iii) | 0.9565 | −1.6733 | −0.3941 | −0.0697 | 0.9773 | −1.7249 | −0.3686 | −0.0463 |
| | (0.2376) | (0.3137) | (0.1113) | (0.0930) | (0.0689) | (0.0810) | (0.0337) | (0.0286) |
| | | | | *All $K = 500$ spectral algorithm estimates* | | | | |
| Eq (i) | 1.1653 | −2.3316 | −0.2334 | 0.0701 | 1.1960 | −2.3910 | −0.2033 | 0.0971 |
| | (0.3062) | (0.4585) | (0.1196) | (0.0971) | (0.0917) | (0.0986) | (0.0237) | (0.0165) |
| Eq (ii) | 1.1372 | −2.2497 | −0.2388 | 0.0675 | 1.1753 | −2.3318 | −0.2158 | 0.0866 |
| | (0.4481) | (1.0049) | (0.2771) | (0.2380) | (0.1575) | (0.3693) | (0.1014) | (0.0867) |
| Eq (iii) | 1.1356 | −2.2413 | −0.2430 | 0.0634 | 1.1904 | −2.3690 | −0.2083 | 0.0926 |
| | (0.4147) | (0.9567) | (0.2732) | (0.2375) | (0.1561) | (0.3621) | (0.1024) | (0.0879) |
| | | | | *Spectral solver estimates* | | | | |
| Eq (i) | 1.1959 | −2.3840 | −0.2309 | 0.0695 | 1.1960 | −2.3910 | −0.2033 | 0.0971 |
| | (0.3382) | (0.4968) | (0.1102) | (0.0826) | (0.0917) | (0.0986) | (0.0237) | (0.0165) |
| Eq (ii) | 1.2283 | −2.4820 | −0.1889 | 0.1083 | 1.1940 | −2.3809 | −0.2055 | 0.0950 |
| | (0.5376) | (1.1835) | (0.3226) | (0.2752) | (0.1547) | (0.3486) | (0.0979) | (0.0836) |
| Eq (iii) | 1.2502 | −2.5356 | −0.1763 | 0.1187 | 1.2011 | 2.3986 | −0.2020 | 0.0979 |
| | (0.5357) | (1.1805) | (0.3309) | (0.2836) | (0.1486) | (0.3313) | (0.0956) | (0.0820) |

*Note*: Averages and standard errors (in brackets) computed over 500 Monte Carlo samples. The $K = 500$ NPL algorithm is deemed having converged if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$ for some $k \leq 500$. Since the NPL algorithm, the spectral algorithm and the spectral solver always converged, the results conditional on convergence are the same as the ones obtained from all samples and are therefore not reported.

number of observations per state (about 250 when $M = 1000$), which may not be the case in many empirical applications.

Our results illustrate how the instability of the equilibrium generating the data alters the properties of the NPL fixed-point algorithm estimates. While the NPL algorithm performs very well in equilibrium (i), it is severely biased in equilibria (ii) and (iii). The NPL algorithm seems to be converging to an inconsistent NPL fixed point in the latter cases. The observed bias is not a consequence of the convergence selection since the NPL algorithm converges in all Monte Carlo samples.

We also find that the relaxation algorithm is not performing better than the NPL algorithm. While it performs well in equilibrium (i), it seems to be converging to an inconsistent NPL fixed point in equilibria (ii) and (iii). In fact, the results obtained from the relaxation algorithm do not differ much from the NPL algorithm's estimates. This finding may be due to using $\alpha = 0.05$ as opposed to using the unfeasible optimal value.

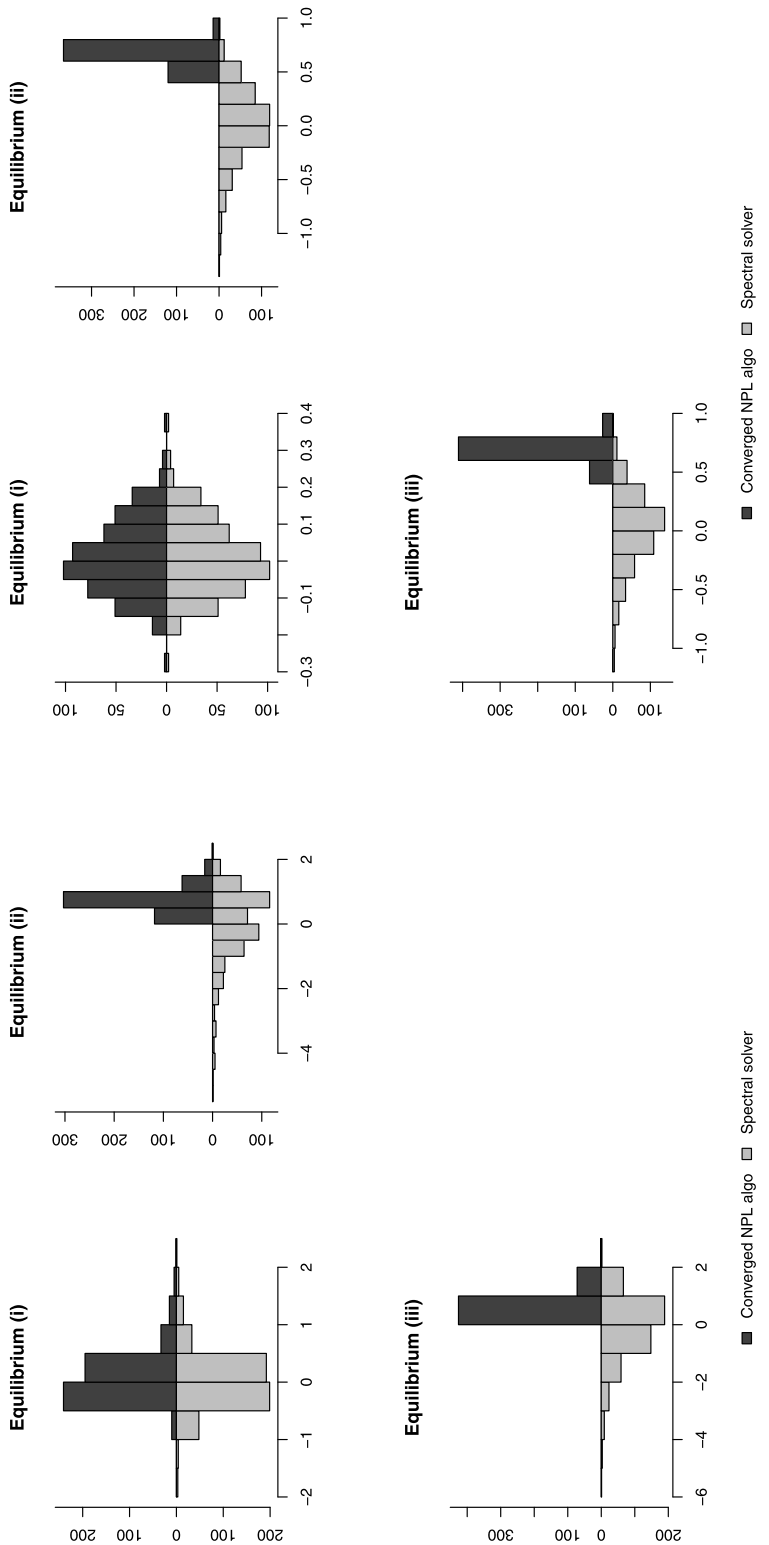
On average, the estimates generated by the spectral approach, especially the ones obtained from the spectral solver, are much closer to the true values of the parameters. This is true, for all equilibria generating the data. The relatively smaller bias for the spectral solver compared to the spectral algorithm is also aligned with the finding that the latter does not reach the NPL estimator as often.

*Distribution of estimates* In Figure 4, we plot the same distributions as in Figure 3 for the strategic interaction parameter $\theta_C$. Once again, the distribution of the spectral solver estimates is roughly normal (perhaps skewed in smaller samples) and centered near the true value. While the distribution of the NPL algorithm estimates is very similar to the spectral solver's in equilibrium (i) it is strikingly different for equilibria (ii) and (iii). Very interestingly, the distributions are quite different from the ones obtained in Experiments I. There is no truncation due to convergence selection: all Monte Carlo samples lead to an estimate for both algorithms. However, the distribution of the NPL algorithm estimates is now concentrated around a value that is not the true value of the parameter. This evidence supports that the NPL algorithm converges to an inconsistent NPL fixed point in equilibria (ii) and (iii).

Figure C2, in Appendix C.4, shows that the Monte Carlo samples with $\hat{\rho}_{\text{NPL}} < 1$ are associated with estimates of the strategic interaction parameter that are different from the estimates in samples with $\hat{\rho}_{\text{NPL}} \geq 1$. Provided that the NPL algorithm fails to deliver the NPL estimator when $\hat{\rho}_{\text{NPL}} \geq 1$, it follows that the NPL algorithm's estimates may also lead to an attenuation bias in the current set of data generating processes.

*Computational cost* Table 9 presents results on the computational cost of the different algorithms in these experiments.[22] Once again, the computation time per iteration is very similar for the NPL, the relaxation, and the spectral algorithms, and we do find that the number of iterations required to reach convergence is, in many cases, smaller for the spectral algorithm. The spectral algorithm's total time until convergence is often smaller

[22]Overall, the computational burden of all the algorithms considered in the analysis is much smaller in Experiments II than in Experiments I. This difference is due to the drastic difference in the size of the state spaces (4 vs. 160) and the number of parameters to be estimated (4 vs. 8).




(a) $M = 100$ (b) $M = 1000$

FIGURE 4. Histograms of $\hat{\theta}_{\mathrm{C}} - \theta_{\mathrm{C}}^{0}$ (Converged NPL algorithm vs. spectral solver.)

TABLE 9. Computational cost—Experiments II.

| | Equilibrium (i) | | Equilibrium (ii) | | Equilibrium (iii) | |
|---|---|---|---|---|---|---|
| | 100 | 1K | 100 | 1K | 100 | 1K |
| | | | *NPL algorithm* | | | |
| Number of iterations | 52.13 | 56.52 | 33.73 | 24.35 | 32.06 | 22.77 |
| | (22.79) | (10.86) | (36.83) | (9.39) | (31.88) | (3.98) |
| Time per iteration | | | | | | |
| Full iteration | 0.008411 | 0.018849 | 0.009433 | 0.022783 | 0.009343 | 0.022809 |
| | (0.000670) | (0.001157) | (0.000816) | (0.001353) | (0.000863) | (0.001125) |
| Update parameters | 0.008322 | 0.018764 | 0.009349 | 0.022696 | 0.009259 | 0.022749 |
| | (0.000696) | (0.001155) | (0.000850) | (0.001379) | (0.000869) | (0.001139) |
| Update CCPs | 0.000065 | 0.000063 | 0.000066 | 0.000063 | 0.000072 | 0.000048 |
| | (0.000149) | (0.000135) | (0.000190) | (0.000202) | (0.000213) | (0.000175) |
| Time until convergence | 0.437 | 1.064 | 0.320 | 0.554 | 0.302 | 0.519 |
| | (0.200) | (0.209) | (0.351) | (0.215) | (0.310) | (0.084) |
| | | | *Relaxation algorithm* | | | |
| Number of iterations | 397.35 | 395.01 | 330.08 | 293.25 | 327.49 | 296.82 |
| | (70.14) | (75.61) | (74.20) | (63.78) | (65.94) | (54.85) |
| Time per iteration | | | | | | |
| Full iteration | 0.008452 | 0.018948 | 0.009383 | 0.022740 | 0.009294 | 0.022641 |
| | (0.000541) | (0.001125) | (0.000705) | (0.001129) | (0.000715) | (0.001078) |
| Update parameters | 0.008365 | 0.018861 | 0.009300 | 0.022658 | 0.009211 | 0.022558 |
| | (0.000545) | (0.001126) | (0.000705) | (0.001129) | (0.000714) | (0.001078) |
| Update CCPs | 0.000068 | 0.000068 | 0.000064 | 0.000066 | 0.000065 | 0.000065 |
| | (0.000052) | (0.000056) | (0.000055) | (0.000059) | (0.000056) | (0.000059) |
| Time until convergence | 3.308 | 7.412 | 3.091 | 6.687 | 3.039 | 6.732 |
| | (0.586) | (1.495) | (0.721) | (1.570) | (0.651) | (1.313) |
| | | | *Spectral algorithm* | | | |
| Number of iterations | 26.58 | 26.43 | 23.87 | 34.07 | 24.08 | 31.49 |
| | (15.08) | (7.80) | (15.68) | (22.63) | (16.93) | (22.30) |
| Time per iteration | | | | | | |
| Full iteration | 0.008509 | 0.019258 | 0.009294 | 0.021727 | 0.009273 | 0.021451 |
| | (0.000714) | (0.001214) | (0.001005) | (0.001667) | (0.000943) | (0.001428) |
| Update parameters | 0.008383 | 0.019138 | 0.009159 | 0.021601 | 0.009158 | 0.021342 |
| | (0.000763) | (0.001233) | (0.001059) | (0.001654) | (0.000978) | (0.001447) |
| Update CCPs | 0.000094 | 0.000099 | 0.000109 | 0.000101 | 0.000091 | 0.000091 |
| | (0.000240) | (0.000242) | (0.000291) | (0.000247) | (0.000268) | (0.000242) |
| Time until convergence | 0.227 | 0.512 | 0.222 | 0.747 | 0.226 | 0.680 |
| | (0.137) | (0.171) | (0.147) | (0.511) | (0.167) | (0.483) |
| | | | *Spectral solver* | | | |
| Time until convergence | 1.706 | 4.031 | 2.137 | 6.967 | 2.203 | 6.955 |
| | (1.775) | (4.014) | (1.088) | (4.414) | (1.791) | (4.150) |

*Note*: Number of iterations needed to converge (conditional on convergence) and computational time in seconds. Averages and standard deviations (in brackets) computed over 500 Monte Carlo samples. The NPL, relaxation, and spectral algorithms are deemed having converged at $k$th iteration if $\max\{|\hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}|\} < 10^{-5}$. Time to perform a full iteration is divided in two parts: updating the parameters $\boldsymbol{\theta}$ given the CCPs (a pseudo-maximum likelihood problem) and updating the CCPs given the parameters (using the algorithm-specific rule). For the spectral solver, the total computation time to reach convergence (with R's BBsolve default tolerance, i.e., the $L_2$ norm of $\hat{\boldsymbol{\phi}}(\mathbf{P})$ being smaller than $\sqrt{\dim(\mathbf{P})} \times 10^{-7}$) is reported.

than the NPL and the relaxation algorithms. We therefore conclude that the benefit of the spectral approach does not come at a higher computational burden.

An important difference between the results obtained from our second set of experiments compared to the first one is that the smaller computational burden of each estimation has allowed us to considerably increase the number of iterations from $K = 100$ to $K = 500$. By doing so, we have found that the relaxation algorithm did converge for a relatively large number of Monte Carlo samples. However, Table 9 shows that, on average, the relaxation method requires a large number of iterations (typically above 300) to converge. Of course, this large number of iterations may be due to using $\alpha = 0.05$.

As mentioned above, as opposed to the first set of experiments, the spectral algorithm does not always generate the same estimates as the spectral solver. In particular, it may fail to deliver the NPL estimator. In that sense, the nonmonotone line search and the other safeguards built in the spectral solver from R's BBsolve function seem to be more relevant in Pesendorfer and Schmidt-Dengler's (2008) data generating processes than in Aguirregabiria and Mira (2007). It is true that these additions increase the computational cost: There is a significant difference in the time until convergence for the spectral solver compared to the spectral algorithm's. However, it is worth emphasizing that the spectral solver's computational burden remains small.

### 6.3 *Summary and practical implications*

Our local asymptotic analysis and our simulation exercises shed new light on the properties of the estimator defined upon convergence of the NPL algorithm. We confirm existing convergence issues of this fixed-point algorithm, and characterize the condition under which this simple iterative approach delivers the NPL estimator. This stability condition depends on the NPL estimator itself, such that it is not feasible in practice to check for this condition before applying the fixed-point algorithm. For this reason, we see the spectral algorithm as a better approach to compute the NPL estimator. We also show that the additional computational burden of using spectral algorithms is small.

The spectral approach typically succeeds in finding NPL fixed points without having to take a stand on their stability in the sample nor in the population. Given its better convergence properties, we recommend implementing the nonmonotone line search, which is available in off-the-shelf software packages such as R's BBsolve function. A very small coding effort is required beyond the code needed to perform the NPL fixed-point algorithm. The code of a $K = 1$ step iteration is used to construct the system of nonlinear equations to be solved by the spectral solver. We also emphasize that one must use multiple starting values and pick the fixed point maximizing the log-likelihood function in case where the sample NPL mapping admits several fixed points.

## 7. Application

In this section, we present an application to a dynamic game of restaurant location between McDonalds (MD) and Burger King (BK) using data for the UK during the period 1991–1995. The dataset was collected by Toivanen and Waterson (2005), and has also

been used in Aguirregabiria and Magesan (2020). We refer the reader to these papers for details about the data. The main purpose of this section is to illustrate the implementation of the algorithms using actual data. In particular, the dimensionality of the vector of equilibrium CCPs in many empirical applications may be much larger than in our Monte Carlo simulations. This section highlights that the spectral approach finds the NPL estimator in an application with a vector of CCPs that contains almost 190,000 elements.

### 7.1 *Payoff function and state space*

In each market $m$ at time $t$, chain $i \in \{\text{BK}, \text{MD}\}$ decides whether it adds a new restaurant ($y_{imt} = 1$) or not ($y_{imt} = 0$) to its existing stock of restaurants ($r_{imt} = \sum_{s<t} y_{ims}$). The game is dynamic because opening a new restaurant is an irreversible decision. Chain $i$'s payoff depends on its total number of restaurants ($n_{imt} = r_{imt} + y_{imt}$), its competitor's ($n_{jmt} = r_{jmt} + y_{jmt}$), and exogenous market characteristics assumed to be constant over time such as market size ($s_m$) and a vector $\mathbf{w}_m$ containing population density, income, average rent, and retail taxes. Let $\mathbf{y}_{mt} = [y_{\text{BK},mt}, y_{\text{MD},mt}]'$ and $\mathbf{x}_{mt} = [r_{\text{BK},mt}, r_{\text{MD},mt}, s_m, \mathbf{w}_m']'$. The current payoff function depends on variable profits (VP), fixed costs (FC), and a standard normal private information shock $\varepsilon_{imt}$ such that $U_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}, \varepsilon_{imt})$ is equal to

$$\text{VP}_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}) - \text{FC}_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}) + y_{imt}\varepsilon_{imt}, \tag{28}$$

with

$$\begin{aligned}&\text{VP}_i(\mathbf{y}_{mt}, \mathbf{x}_{mt})\\ &\quad = s_m\big[\theta^0_{\text{VP},i}\mathbb{1}\{n_{imt} > 0\} + \theta^1_{\text{VP},i}(n_{imt} - n_{jmt}) + \theta^2_{\text{VP},i}(n_{imt} - n_{jmt})^2\big], \quad \text{and} \qquad (29)\\ &\text{FC}_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}) = \theta^0_{\text{FC},i}\mathbb{1}\{n_{imt} > 0\} + \theta^1_{\text{FC},i}n_{imt} + \theta^2_{\text{FC},i}n^2_{imt} + n_{imt}\mathbf{w}_m'\boldsymbol{\theta}_{\text{MS}}. \qquad (30)\end{aligned}$$

This payoff function is linear in parameters such that $U_i(\mathbf{y}_{mt}, \mathbf{x}_{mt}, \varepsilon_{imt}) = \mathbf{z}_{imt}(y_{imt}, y_{jmt})'\boldsymbol{\theta}_i + y_{imt}\varepsilon_{imt}$ where $\boldsymbol{\theta}_i = [\theta^0_{\text{VP},i}, \theta^1_{\text{VP},i}, \theta^2_{\text{VP},i}, \theta^0_{\text{FC},i}, \theta^1_{\text{FC},i}, \theta^2_{\text{FC},i}, \boldsymbol{\theta}_{\text{MS}}']'$, and $\mathbf{z}_{imt}(y_{imt}, y_{jmt})'$ is

$$\big[s_m\mathbb{1}\{n_{imt} > 0\}, s_m(n_{imt} - n_{jmt}), s_m(n_{imt} - n_{jmt})^2, \mathbb{1}\{n_{imt} > 0\}, n_{imt}, n^2_{imt}, n_{imt}\mathbf{w}_m'\big]. \tag{31}$$

Since $s_m$ and $\mathbf{w}_m$ are constant over time, we consider that each market $m$ corresponds to a particular state. The joint support of $r_{\text{BK},mt}$ and $r_{\text{MD},mt}$ determines the size of the state space per market. There are at most 13 restaurants owned by a single chain in a given market in the data. We therefore consider the support of $r_{imt}$ to be $\{0, 1, \ldots, 14\}$ such that there are $15 \times 15 = 225$ possible states per market. Since there are 422 markets and two players in the data, the NPL mapping is defined over a total of $225 \times 422 \times 2 = 189{,}900$ CCPs.

### 7.2 *Step-by-step implementation*

First, we describe in five steps the construction of the NPL mapping. Then we explain how each algorithm—NPL fixed point, relaxation, and spectral—uses this mapping. We drop the market index $m$ to alleviate the notation.

*Step 1: Compute expected current payoffs for every action and state* The expected current payoff of chain $i$ choosing $y_{it}$ given $\mathbf{x}_t$ is

$$U_i^{\mathbf{P}}(y_{it}, \mathbf{x}_t, \varepsilon_{it}) = \mathbf{z}_{it}^{\mathbf{P}}(y_{it})'\boldsymbol{\theta}_i + y_{it}\varepsilon_{it}, \tag{32}$$

where $\mathbf{z}_{it}^{\mathbf{P}}(y_{it}) \equiv P_j(\mathbf{x}_t)\mathbf{z}_{it}(y_{it}, 1) + (1 - P_j(\mathbf{x}_t))\mathbf{z}_{it}(y_{it}, 0)$ and $P_j(\mathbf{x}_t) = \Pr(y_{jt} = 1|\mathbf{x}_t)$.

*Step 2: Transition probabilities* Compute the transition from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$ conditional on $y_{it}$ in all possible states for both chains. For chain $i$, these transition probabilities are

$$\Gamma_i^{\mathbf{P}}(\mathbf{x}_{t+1}|\mathbf{x}_t, y_{it}) = \mathbb{1}\{r_{i,t+1} = r_{it} + y_{it}\}P_j(\mathbf{x}_t)^{\mathbb{1}\{r_{j,t+1}=r_{jt}+1\}}\big(1 - P_j(\mathbf{x}_t)\big)^{\mathbb{1}\{r_{j,t+1}=r_{jt}\}}. \tag{33}$$

Let $\boldsymbol{\Gamma}_{\mathbf{x}}^{\mathbf{P}}$ be the transition matrix after integrating out both chains' decisions. The elements of this matrix are: $(1 - P_i(\mathbf{x}_t))\Gamma_i^{\mathbf{P}}(\mathbf{x}_{t+1}|\mathbf{x}_t, 0) + P_i(\mathbf{x}_t)\Gamma_i^{\mathbf{P}}(\mathbf{x}_{t+1}|\mathbf{x}_t, 1)$.

*Step 3: Discounted payoffs* Chain $i$'s expected and discounted sum of current and future payoffs when it chooses $y_{it}$ given $\mathbf{x}_t$ can be written as $\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(y_{it})'\boldsymbol{\theta}_i + \tilde{e}_{it}^{\mathbf{P}}(y_{it}) + y_{it}\varepsilon_{it}$ where

$$\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(y_{it}) \equiv \mathbf{z}_{it}^{\mathbf{P}}(y_{it}) + \beta \sum_{\mathbf{x}_{t+1}\in\mathcal{X}} \Gamma_i^{\mathbf{P}}(\mathbf{x}_{t+1}|\mathbf{x}_t, y_{it})\mathbf{v}_{\mathbf{z}i}^{\mathbf{P}}(\mathbf{x}_{t+1}), \tag{34}$$

$$\tilde{e}_{it}^{\mathbf{P}}(y_{it}) \equiv \beta \sum_{\mathbf{x}_{t+1}\in\mathcal{X}} \Gamma_i^{\mathbf{P}}(\mathbf{x}_{t+1}|\mathbf{x}_t, y_{it})v_{ei}^{\mathbf{P}}(\mathbf{x}_{t+1}) \tag{35}$$

with $\mathbf{v}_{\mathbf{z}i}^{\mathbf{P}}(\mathbf{x}_t) = (1 - P_i(\mathbf{x}_t))\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(0) + P_i(\mathbf{x}_t)\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(1)$ being a vector of the same dimension as $\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(y_{it})$; $v_{ei}^{\mathbf{P}}(\mathbf{x}_t) = (1 - P_i(\mathbf{x}_t))\tilde{e}_{it}^{\mathbf{P}}(0) + P_i(\mathbf{x}_t)\tilde{e}_{it}^{\mathbf{P}}(1) + e_{it}^{\mathbf{P}}$ being a scalar, and $e_{it}^{\mathbf{P}} = \mathrm{E}[y_{it}\varepsilon_{it}|\mathbf{x}_t, y_{it} \text{ is optimal}] = \phi(\Phi^{-1}(P_i(\mathbf{x}_t)))$, with $\phi(\cdot)$ and $\Phi(\cdot)$ being the pdf and the cdf of the standard normal distribution, respectively. The closed-form expressions for the matrix $\mathbf{V}_{\mathbf{z}i}^{\mathbf{P}} \equiv \{\mathbf{v}_{\mathbf{z}i}^{\mathbf{P}}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ and the vector $\mathbf{V}_{\mathbf{e}i}^{\mathbf{P}} \equiv \{v_{ei}^{\mathbf{P}}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ are given by

$$\mathbf{V}_{\mathbf{z}i}^{\mathbf{P}} = \big[\mathbf{I} - \beta\boldsymbol{\Gamma}_{\mathbf{x}}^{\mathbf{P}}\big]^{-1}\big[(1 - \mathbf{P}_i) * \mathbf{Z}_i^{\mathbf{P}}(0) + \mathbf{P}_i * \mathbf{Z}_i(1)\big], \tag{36}$$

$$\mathbf{V}_{\mathbf{e}i}^{\mathbf{P}} = \big[\mathbf{I} - \beta\boldsymbol{\Gamma}_{\mathbf{x}}^{\mathbf{P}}\big]^{-1}\mathbf{e}_i^{\mathbf{P}}, \tag{37}$$

where $*$ denotes the Hadamard product, $\mathbf{I}$ is the identity matrix, $\mathbf{P}_i$ is the vector containing $P_i(\mathbf{x})$ for all possible values of $\mathbf{x}$, $\mathbf{Z}_i^{\mathbf{P}}(y)$ is a matrix with rows corresponding to $\mathbf{z}_{it}^{\mathbf{P}}(y)'$ for all possible values of $\mathbf{x}$, and $\mathbf{e}_i^{\mathbf{P}}$ is a vector containing $e_{it}^{\mathbf{P}}$ for all possible values of $\mathbf{x}$.

*Step 4: Best response mapping* The values of $\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(y_{it})$ and $\tilde{e}_{it}^{\mathbf{P}}(y_{it})$ are used to compute the best response mapping $\Phi([\tilde{\mathbf{z}}_{it}^{\mathbf{P}}(1) - \tilde{\mathbf{z}}_{it}^{\mathbf{P}}(0)]'\boldsymbol{\theta}_i + \tilde{e}_{it}^{\mathbf{P}}(1) - \tilde{e}_{it}^{\mathbf{P}}(0))$.

*Step 5: NPL mapping* The best response mapping is used to construct the log-likelihood function to be maximized over $\boldsymbol{\theta}$ for a given vector $\mathbf{P}$ to obtain $\hat{\boldsymbol{\vartheta}}(\mathbf{P})$. The sample NPL mapping $\hat{\boldsymbol{\varphi}}(\mathbf{P})$ is constructed by evaluating the best response mapping at $\mathbf{P}$ and $\hat{\boldsymbol{\vartheta}}(\mathbf{P})$.

*NPL algorithm* The NPL algorithm uses the NPL mapping to update the CCPs. For an initial $\hat{\mathbf{P}}_0$, one obtains $\hat{\mathbf{P}}_1 = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_0)$, then $\hat{\mathbf{P}}_2 = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_1)$, etc., and iterates until the maximum number of iterations is reached or until the maximum absolute value between two successive vectors $\hat{\mathbf{P}}_k$ and $\hat{\mathbf{P}}_{k-1}$ is smaller than some tolerance level.

*Relaxation algorithm* For an initial $\hat{\mathbf{P}}_0$ and a fixed value of the relaxation parameter $\alpha$, the relaxation algorithm generates a new vector of CCPs as $\hat{\mathbf{P}}_1 = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_0)^{\alpha} * \hat{\mathbf{P}}_0^{1-\alpha}$, then $\hat{\mathbf{P}}_2 = \hat{\boldsymbol{\varphi}}(\hat{\mathbf{P}}_1)^{\alpha} * \hat{\mathbf{P}}_1^{1-\alpha}$, etc., and iterates until the maximum number of iterations is reached or until the maximum absolute value between two successive vectors $\hat{\mathbf{P}}_k$ and $\hat{\mathbf{P}}_{k-1}$ is smaller than some tolerance level. We use $\alpha = 0.05$ as in our simulations.

*Spectral solver* One constructs the function $\mathbf{P} - \hat{\boldsymbol{\varphi}}(\mathbf{P})$ and uses R's BBsolve to find a solution to $\mathbf{P} - \hat{\boldsymbol{\varphi}}(\mathbf{P}) = \mathbf{0}$. We use the default settings, with the following exceptions. The parameters "maxit" and "tol" are set to the same maximum number of iterations and the tolerance level, respectively, used in the other two algorithms. We set the parameter "method" to $c(2, 3, 1)$ to use the same consecutive trials of spectral step lengths as in our simulations.

*Other details* All three methods are initiated at the same set of 100 vectors of starting values for the $\mathbf{P}$. The first vector of starting values correspond to the frequency count estimator. The other 99 starting values are independent draws from the uniform distribution. We restrict choice probabilities to be between $10^{-12}$ and $1 - 10^{-12}$ both for the starting values and the updated vectors of CCPs. The tolerance level used to declare convergence is $10^{-6}$ for all methods and the maximum number of iterations is set to 1000.

### 7.3 *Estimates and computational time*

Except for one vector of starting values for the relaxation algorithm,[23] we find that all three methods converge to the same NPL estimator across all starting values. The estimates associated with the starting values leading to the best value of the log-likelihood are reported in Table D1 in Appendix D. They are almost exactly the same for all three methods. For this data set, neither the relaxation algorithm nor the spectral solver finds any unstable fixed point in the sample NPL mapping.

We have also computed the total time until convergence for each vector of starting values and each method. The results are aligned with our findings in Section 6. Despite the dimensionality of the vector of equilibrium CCPs, we confirm that the spectral solver is an appealing alternative to computing the NPL estimator. Its computational cost is even slightly smaller than the NPL algorithm's. Once again, the relaxation algorithm is slower to converge, at least when the value of the relaxation parameter is fixed to 0.05.

## 8. Concluding remarks

There are multiple reasons why researchers may want to impose the equilibrium restrictions of their models when estimating structural parameters: efficiency, reduction of finite sample bias, allowing for rich forms of unobserved heterogeneity, or using estimated choice probabilities that are comparable to those obtained in counterfactual

[23]The only exception is the vector of starting values corresponding to the frequency count estimator. For this vector of starting values, the relaxation algorithm eventually leads to two successive vectors of CCPs that satisfy the convergence criterion, but do not correspond to an NPL fixed point. This issue is likely a consequence of the vector of starting values containing many numerical zeros.

experiments. Imposing equilibrium restrictions requires researchers to go beyond simple two-step estimators.

In the estimation of dynamic discrete games, the existing algorithms that impose the model equilibrium restrictions—NFXP, NPL, MPEC, and their variations—have different merits and limitations. As a quasi-Newton algorithm, MPEC guarantees local convergence, but it requires the repeated computation of high-dimensional Jacobian matrices at each iteration of the algorithm. In contrast, the NPL algorithm does not involve the computation of these Jacobian matrices. However, since the NPL algorithm uses fixed-point iterations, it requires some stability conditions to converge. Simulation evidence has shown that convergence may fail even if the stability conditions are satisfied at the population level.

This paper has two main contributions. First, we investigate the reasons and the implications of the lack of convergence of the NPL algorithm. We show that the randomness in the sample NPL mapping—used to update the CCPs—is the culprit. While typically ignored in existing asymptotic studies of the properties of the NPL algorithm, the role of this sample randomness is made obvious when considering that the NPL algorithm is, in practice, iterated in fixed samples. We show that any data generating process in a neighborhood of the NPL fixed-point stability threshold always has a strictly positive probability to generate samples where the algorithm does not converge, as well as samples where the algorithm does converge. We show that the samples for which the algorithm converges are different in a particular and substantial way. This difference generates a convergence selection bias in this type of estimation algorithms. We characterize the nature of this bias and show in Monte Carlo experiments that it can be substantial. This selection bias has a particularly important impact on the structural parameters that capture strategic interactions or competition effects: the estimates of these parameters are biased toward zero. Intuitively, the algorithm converges only for samples consistent with weaker strategic interactions, and this introduces an attenuation bias in the estimation of these strategic parameters.

Second, we propose and implement a spectral algorithm to obtain the NPL estimator. This algorithm has advantages relative to the NPL and the MPEC algorithms. Compared to NPL, the spectral algorithm satisfies local convergence to the consistent estimator. And in contrast to the MPEC algorithm, it does not require the calculation of high-dimensional Jacobian matrices. Furthermore, the dimensionality of the spectral approach's optimization problem is smaller than the MPEC method, which searches over the joint spaces of CCPs, Lagrange multipliers, and structural parameters.

## References

Aguirregabiria, V. and C. Alonso-Borrego (2014), "Labor contracts and flexibility: Evidence from a labor market reform in Spain." *Economic Inquiry*, 52 (2), 930–957. [1228]

Aguirregabiria, V. and C. Ho (2012), "A dynamic oligopoly game of the US airline industry: Estimation and policy experiments." *Journal of Econometrics*, 168 (1), 156–173. [1228]

Aguirregabiria, V. and A. Magesan (2020), "Identification and estimation of dynamic games when players' beliefs are not in equilibrium." *The Review of Economic Studies*, 87 (2), 582–625. [1265]

Aguirregabiria, V. and M. Marcoux (2021), "Supplement to 'Imposing equilibrium restrictions in the estimation of dynamic discrete games'." *Quantitative Economics Supplemental Material*, 12, https://doi.org/10.3982/QE1735. [1241]

Aguirregabiria, V. and P. Mira (2002), "Swapping the nested fixed point algorithm: A class of estimators for discrete Markov decision models." *Econometrica*, 70 (4), 1519–1543. [1224, 1233, 1234, 1246]

Aguirregabiria, V. and P. Mira (2007), "Sequential estimation of dynamic discrete games." *Econometrica*, 75 (1), 1–53. [1224, 1225, 1226, 1227, 1230, 1231, 1232, 1233, 1234, 1245, 1246, 1257, 1264]

Bajari, P., C. Benkard, and J. Levin (2007), "Estimating dynamic model of imperfect competition." *Econometrica*, 75 (1), 1331–1370. [1226]

Barzilai, J. and J. Borwein (1988), "Two-point step size gradient methods." *IMA Journal of Numerical Analysis*, 8 (1), 141–148. [1235]

Bugni, F. and J. Bunting (2021), "On the iterated estimation of dynamic discrete choice games." *The Review of Economic Studies*, 88 (3), 1031–1073. [1227]

Bugni, F. and T. Ura (2019), "Inference in dynamic discrete choice problems under local misspecification." *Quantitative Economics*, 10 (1), 67–103. [1241]

Collard-Wexler, A. (2013), "Demand fluctuations in the ready-mix concrete industry." *Econometrica*, 81 (3), 1003–1037. [1228]

Copeland, A. and C. Monnet (2009), "The welfare effects of incentive schemes." *The Review of Economic Studies*, 76 (1), 93–113. [1228]

De Pinto, A. and G. Nelson (2009), "Land use change with spatially explicit data: A dynamic approach." *Environmental and Resource Economics*, 43 (2), 209–229. [1228]

Dearing, A. and J. Blevins (2021), "Efficient and convergent sequential pseudo-likelihood estimation of dynamic discrete games." arXiv eprint, https://arxiv.org/pdf/1912.10488v2.pdf. [1228, 1258]

Egesdal, M., Z. Lai, and C. Su (2015), "Estimating dynamic discrete-choice games of incomplete information." *Quantitative Economics*, 6 (3), 567–597. [1224, 1227, 1228, 1230, 1238, 1239, 1246]

Ellickson, P. and S. Misra (2008), "Supermarket pricing strategies." *Marketing Science*, 27 (5), 811–828. [1228]

Elliott, G., T. Rothenberg, and J. Stock (1996), "Efficient tests for an autoregressive unit root." *Econometrica*, 64 (4), 813–836. [1241]

Fletcher, R. (1990), "Low storage methods for unconstrained optimization." *Lectures in Applied Mathematics*, 26 (1), 165–179. [1236]

Gayle, P. and X. Xie (2018), "Entry deterrence and strategic alliances." *Economic Inquiry*, 56 (3), 1898–1924. [1228]

Han, L. and S. Hong (2011), "Testing cost inefficieny under free entry in the real estate brokerage industry." *Journal of Business and Economic Statistics*, 29 (4), 564–578. [1228]

Hausman, J. (1978), "Specification tests in econometrics." *Econometrica*, 46 (6), 1251–1271. [1241]

Hotz, V. and R. Miller (1993), "Conditional choice probabilities and the estimation of dynamic models." *The Review of Economic Studies*, 60 (3), 497–529. [1226]

Huang, L. and M. Smith (2014), "The dynamic efficiency costs of common-pool resource exploitation." *American Economic Review*, 104 (12), 4071–4103. [1228]

Iskhakov, F., J. Lee, J. Rust, B. Schjerning, and K. Seo (2016), "Comment on 'Constrained optimization approaches to estimation of structural models'." *Econometrica*, 84 (1), 365–370. [1228]

Kano, K. (2013), "Menu costs and dynamic duopoly." *International Journal of Industrial Organization*, 31 (1), 102–118. [1228]

Kasahara, H. and K. Shimotsu (2012), "Sequential estimation of structural models with a fixed point constraint." *Econometrica*, 80 (5), 2303–2319. [1227, 1228, 1230, 1234, 1240, 1246, 1248, 1250]

La Cruz, W., J. Martinez, and M. Raydan (2006), "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations." *Mathematics of Computation*, 75 (255), 1429–1448. [1234, 1235, 1236, 1237, 1248]

Lin, H. (2015), "Quality choice and market structure: A dynamic analysis of nursing home oligopolies." *International Economic Review*, 56 (4), 1261–1290. [1228]

Lin, Z. and H. Xu (2017), "Estimation of social-influence-dependent peer pressure in a large network game." *The Econometrics Journal*, 20 (3), S86–S102. [1228]

Liu, X. and J. Zhou (2017), "A social interaction model with ordered choices." *Economics Letters*, 161 (1), 86–89. [1228]

Mogensen, P. (2015), "Monte Carlo studies and the convergence of NPL for games." Working paper. [1227, 1253]

Pakes, A., M. Ostrovsky, and S. Berry (2007), "Simple estimators for the parameters of discrete dynamic games (with entry/exit examples)." *RAND Journal of Economics*, 38 (2), 373–399. [1226]

Pesendorfer, M. and P. Schmidt-Dengler (2008), "Asymptotic least squares estimators for dynamic games." *The Review of Economic Studies*, 75 (3), 901–928. [1224, 1225, 1226, 1227, 1230, 1244, 1245, 1257, 1258, 1264]

Pesendorfer, M. and P. Schmidt-Dengler (2010), "Sequential estimation of dynamic discrete games: A comment." *Econometrica*, 78 (2), 833–842. [1224, 1227, 1240, 1245]

Phillips, P. (1987), "Towards a unified asymptotic theory for autoregression." *Biometrika*, 74 (3), 535–547. [1241]

Phillips, P. and P. Perron (1988), "Testing for a unit root in time series regression." *Biometrika*, 75 (2), 335–346. [1241]

Rust, J. (1987), "Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher." *Econometrica*, 55 (5), 999–1033. [1226, 1230]

Staiger, D. and J. Stock (1997), "Instrumental variables regression with weak instruments." *Econometrica*, 65 (3), 557–586. [1241]

Su, C. and K. Judd (2012), "Constrained optimization approaches to estimation of structural models." *Econometrica*, 80 (5), 2213–2230. [1228]

Sweeting, A. (2013), "Dynamic product positioning in differentiated product industries: The effect of fees for musical performance rights on the commercial radio industry." *Econometrica*, 81 (5), 1763–1803. [1228]

Toivanen, O. and M. Waterson (2005), "Market structure and entry: Where's the beef?" *The RAND Journal of Economics*, 36 (3), 680–699. [1264]

Tomlin, B. (2014), "Exchange rate fluctuations, plant turnover and productivity." *International Journal of Industrial Organization*, 35 (1), 12–28. [1228]

Varadhan, R. and P. Gilbert (2009), "BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function." *Journal of Statistical Software*, 32 (4), 1–26. [1248]

Varadhan, R. and C. Roland (2008), "Simple and globally convergent methods for accelerating the convergence of any EM algorithm." *Scandinavian Journal of Statistics*, 35 (2), 335–353. [1236]

---

Co-editor Andres Santos handled this manuscript.

Manuscript received 27 September, 2020; final version accepted 22 June, 2021; available online 1 July, 2021.