

Avoiding the curse of dimensionality in dynamic stochastic games

ULRICH DORASZELSKI

University of Pennsylvania and CEPR

KENNETH L. JUDD

Hoover Institution and NBER

Discrete-time stochastic games with a finite number of states have been widely applied to study the strategic interactions among forward-looking players in dynamic environments. These games suffer from a “curse of dimensionality” when the cost of computing players’ expectations over all possible future states increases exponentially in the number of state variables. We explore the alternative of continuous-time stochastic games with a finite number of states and argue that continuous time may have substantial advantages. In particular, under widely used laws of motion, continuous time avoids the curse of dimensionality in computing expectations, thereby speeding up the computations by orders of magnitude in games with more than a few state variables. This much smaller computational burden greatly extends the range and richness of applications of stochastic games.

KEYWORDS. Dynamic stochastic games, continuous time, Markov perfect equilibrium, numerical methods.

JEL CLASSIFICATION. C63, C73, L13.

1. INTRODUCTION

Discrete-time stochastic games with a finite number of states are central to the analysis of strategic interactions among forward-looking players in dynamic environments. The usefulness of discrete-time games, however, is limited by their computational burden; in particular, under standard assumptions there is a “curse of dimensionality,” since the cost of computing players’ expectations over all possible future states increases exponentially in the number of state variables. In this paper, we examine the alternative of continuous-time games with a finite number of states and show that under comparable assumptions they avoid this curse of dimensionality. As a consequence, continuous-

Ulrich Doraszelski: doraszelski@wharton.upenn.edu

Kenneth L. Judd: judd@hoover.stanford.edu

We thank Victor Aguirregabiria, Ken Arrow, Lanier Benkard, Michaela Draganska, Gautam Gowrisankaran, Rustam Ibragimov, Sarit Markovich, Ariel Pakes, Katja Seim, Gabriel Weintraub, a co-editor, two anonymous referees, and the participants of SITE 2004, the CRES IO Conference in 2005, and the Informs Annual Meeting in 2006 for their comments. Doraszelski gratefully acknowledges the hospitality of the Hoover Institution during the academic year 2006–2007.

Copyright © 2012 Ulrich Doraszelski and Kenneth L. Judd. Licensed under the [Creative Commons Attribution-NonCommercial License 3.0](http://creativecommons.org/licenses/by-nc/3.0/). Available at <http://www.qeconomics.org>.

DOI: [10.3982/QE153](https://doi.org/10.3982/QE153)

time games with more than a few state variables are orders of magnitude faster to solve than their discrete-time counterparts. Whether an economic problem is best modeled in continuous or discrete time depends not only on the computational burden but also on the details of the institutional and technological setting. We argue that continuous-time formulations of games are often natural. Overall, the continuous-time approach offers a computationally and conceptually promising alternative to modeling dynamic strategic interactions.

Dating back to [Shapley \(1953\)](#), discrete-time stochastic games with a finite number of states have a long tradition in economics. A well known example is the [Ericson and Pakes \(1995\)](#) (hereafter, EP) model of dynamic competition in an oligopolistic industry with investment, entry, and exit. The EP model has triggered a large and active literature in industrial organization and other fields (see [Doraszelski and Pakes \(2007\)](#) for a survey). Because such models are generally too complex to be solved analytically, [Pakes and McGuire \(1994\)](#) (hereafter, PM1) presented an algorithm to solve numerically for a Markov perfect equilibrium.

The range of applications of finite-state stochastic games is limited by their high computational cost. The first source of burden is the large size of the state space. Indeed, there can be a curse of dimensionality in that the number of states increases exponentially in the number of state variables, that is, the dimension of the state vector. To avoid this curse, applications of EP's framework routinely restrict attention to symmetric and anonymous equilibria (see Section 3.3 for details).

While the number of states reflects the richness of the economic environment and is therefore independent of the concept of time, the second source of computational burden is not: As [Pakes and McGuire \(2001\)](#) (hereafter, PM2) pointed out, in discrete-time stochastic games, computing players' expectations over all possible future states can be subject to another curse of dimensionality. Suppose that a player can move to one of K states from one period to the next and that these transitions are independent across players. Given that there are K possibilities for each of N players, there are K^N possibilities for the future state of the game, and computing the expectation over all these successor states therefore involves summing over K^N terms. Because of this exponential increase of the computational burden, applications of discrete-time games are constrained to a handful of players and restrict heterogeneity among players.

In this paper, we develop continuous-time stochastic games with a finite number of states.¹ We show that specifying stochastic games in continuous time has computational advantages because under widely used laws of motion (see Section 2.3 for details), it avoids the curse of dimensionality in computing expectations. In contrast to a discrete-time game, the possibility of two or more players' states changing simultaneously disappears in a continuous-time game. This is not a restriction on the behavior of players; rather it reflects the fact that under these laws of motion, changes happen one by one as time passes. The absence of simultaneous changes implies that the expectation over successor states in the discrete-time game is replaced by a much smaller sum

¹Our approach differs from continuous-time games with a continuum of states which date back to [Isaacs \(1954\)](#) (zero-sum games) and [Starr and Ho \(1969\)](#) (nonzero-sum games); see [Basar and Olsder \(1999\)](#) for a standard presentation of differential games and [Dockner, Jorgensen, Van Long, and Sorger \(2000\)](#) for a survey of applications.

in the continuous-time game and results in a simpler, and computationally much more tractable, model: while computing the expectation over successor states in the discrete-time game involves summing over K^N terms, it merely requires adding up $(K - 1)N$ terms in the continuous-time game. This eliminates the curse of dimensionality in computing expectations.

The third source of computational burden of finite-state stochastic games is the number of iterations. There are reasons to think that an iterative algorithm along the lines of PM1 needs more iterations to converge for continuous-time games than for comparable discrete-time games (see Section 5.2 for details). This “iteration penalty” partly offsets the gain from avoiding the curse of dimensionality in computing expectations.

Throughout this paper, we compare our continuous-time model to the discrete-time model in EP, PM1, PM2, and the subsequent literature (e.g., [Gowrisankaran \(1999a\)](#), [Fershtman and Pakes \(2000\)](#), [Benkard \(2004\)](#)). We therefore restrict attention to games with simultaneous moves, meaning that players choose their actions simultaneously in each period (in the discrete-time model) or at each point in time (in the continuous-time model). We show that under the laws of motion that are commonly assumed in the existing literature, the discrete-time model suffers from a curse of dimensionality in computing the expectation over successor states, whereas the continuous-time model avoids this curse. Despite the iteration penalty, avoiding the curse of dimensionality in computing expectations accelerates the computations by orders of magnitude for games with more than a few state variables.

The computational advantages of continuous time stem from the fact that under widely used laws of motion, the possibility of two or more players’ states changing simultaneously disappears. In discrete time, the curse of dimensionality in computing the expectation over successor states can simply be assumed away by ruling out simultaneous changes in the coordinates of the state vector. Consider a model where each period one player is picked at random to choose an action. The state of the player with the move then changes in response to his/her chosen action. Then another random draw is taken to pick a player, and so on. Because actions are chosen one at a time, this game is one of sequential moves and may be thought of as a “random-leadership Stackelberg game,” whereas the discrete-time model in EP, PM1, PM2, and the subsequent literature as well as our continuous-time model are “Nash games” in which players’ actions are chosen simultaneously. The random-leadership Stackelberg game does not suffer from the curse of dimensionality in computing the expectation over successor states, and in ongoing research ([Doraszelski and Judd \(2007\)](#)), we show that it is almost as fast to solve as our continuous-time model. However, the underlying game-theoretic assumptions (and the institutional realities that justify them) are very different, in much the same way that a static Cournot quantity-setting game differs from a Stackelberg game.

The curse of dimensionality in computing expectations can also be assumed away in discrete time while retaining the assumption of simultaneous moves in the literature following EP. Consider a model where each period each player decides on an action. Then one player is picked at random and his/her state changes in response to the chosen action. Then again each player decides on an action, and so on. This game again

does not suffer from the curse of dimensionality in computing the expectation over successor states. However, because the law of motion must account for the random variable that picks the player whose state is allowed to change, it is not comparable to the ones typically used in the literature following EP. It is also not obvious what institutional realities justify a law of motion where the success of the investment project of one firm implies the failure of the investment project of another firm.

In this paper, we maintain the underlying game-theoretic assumptions and laws of motion in the literature following EP. Even so, there may be substantial differences between a discrete- and a continuous-time formulation of an economic problem and sometimes one or the other approach is preferable. The period length in a discrete-time model is implicitly determined by the discount factor. Moreover, the larger the discount factor, the slower is the convergence of the discrete-time algorithm (see Section 5.2 for details). This is why, in practice, discrete-time models often work with small discount factors such as $\beta = 0.925$ in EP, PM1, and PM2 that imply long periods.² Given that many economic processes unfold in close to continuous time, shorter periods are often desirable.

The so-called embedding problem (Elfvig (1937)) is another source of differences between discrete- and continuous-time games: The discrete-time Markov chains underlying many applications of EP's framework cannot be exactly matched to continuous-time Markov chains in the sense that it may not be possible to construct continuous-time Markov chains that induce the same probability distribution over states at all discrete times $t = 0, 1, 2, \dots$. In this sense, discrete-time Markov chains are richer than continuous-time Markov chains.³ On the other hand, some discrete-time Markov chains can be embedded into more than one continuous-time Markov chain (or even a continuum of continuous-time Markov chains, see Examples 12 and 13 in Singer and Spilerman (1976)). There is thus no easy way to align discrete- and continuous-time Markov chains.

Taken together the long periods in discrete-time models and the embedding problem mean that several issues have to be considered in deciding between a continuous- and a discrete-time formulation of an economic problem. First, in contrast to a continuous-time model, a discrete-time model limits how often and typically also how much a state variable can change over a finite interval of time. Second, in a discrete-time model, a player may react to a change in a rival's state by changing his/her action, but in contrast to a continuous-time model, the player must wait at least a period before this brings about a change in his/her own state. Third, some dynamic phenomena such as predictable seasonal fluctuations in demand or cost and, more generally, calendar time are more easily modeled in discrete time. Our continuous-time approach also rules out

²Mehra (2003) reported that the average real return on a relatively riskless security was about 1% during the twentieth century. If a firm can borrow at a real interest rate of 2%, then $\beta = 0.925$ implies a period length of almost 4 years.

³It is worth noting that applications of EP's framework preclude embeddability by restricting players' transitions to immediately adjacent states. This assumption is often made more to control the computational burden of discrete-time models than for substantive reasons and, in fact, may be undesirable in some settings.

deterministic transitions from one state to another. The more general point here is that continuous- and discrete-time models are different, and both are limited in their ability to accurately represent the real world. The central question is whether a continuous- or a discrete-time model is a better approximation of the economic process under study. The correct answer depends on the details of the institutional and technological setting, and must thus be determined on a case-by-case basis.

None of these differences between discrete- and continuous-time models should distract from the fact that in many cases there are no compelling economic reasons for either discrete or continuous time. A case in point is the quality ladder model developed by PM1 that we use as a running example in this paper. In a case like this, the computational advantages of continuous time may be decisive on their own right. In fact, most existing applications of EP's framework could have been formulated in continuous instead of discrete time with substantial computational savings. To give the reader a sense of the magnitude of these savings, we compare the performance of an algorithm that is closely related to PM1. In discrete time, the algorithm uses over 84 hours per iteration in a model with $N = 14$ players and $K = 3$ possible transitions per player, while in continuous time, the algorithm uses 2.93 seconds per iteration, over 100,000 times faster. Partly offsetting this gain is the fact that for comparable continuous-time games, the algorithm needs more iterations to converge to an equilibrium. This loss, however, is small relative to the gain from avoiding the curse of dimensionality in computing expectations. In the example with $N = 14$ players, continuous time beats discrete time by a factor of almost 30,000. To put this number in perspective, while it takes about 20 minutes to compute an equilibrium of the continuous-time game, it would take over 1 year to compute an equilibrium of the discrete-time game!

Overall, we believe that the advantages of continuous time are often substantial, and open the way to study more complex and realistic stochastic games than currently feasible. Continuous time may also be useful in empirical work on stochastic games; indeed, the subsequent literature ([Kryukov \(2008\)](#), [Arcidiacono, Bayer, Blevins, and Ellickson \(2010\)](#)) has estimated continuous-time games similar to ours.⁴

The remainder of the paper is organized as follows. Section 2 describes the basic elements of discrete- and continuous-time stochastic games with a finite number of states and shows that continuous time avoids the curse of dimensionality under widely used laws of motion. Section 3 presents our computational strategies for both models. Section 4 formulates discrete- and continuous-time versions of the quality ladder model of PM1. Section 5 compares the performance of the discrete- and continuous-time algorithms, and Section 6 discusses a number of conceptual differences between continuous- and discrete-time models. Section 7 concludes.

2. MODELS

In this section, we first describe the discrete- and continuous-time approaches to finite-state stochastic games. Then we show that, under widely used laws of motion, the

⁴Note that we do not have to estimate a discrete-time model just because observations take place periodically. If continuous time more accurately captures the economic process under study, then it is preferable to "aggregate up" the predictions of the continuous-time model to the periodicity of the observations.

continuous-time model avoids the curse of dimensionality in computing the expectation over successor states that the discrete-time model suffers.

2.1 Discrete-time model

A discrete-time stochastic game with a finite number of states is often just called a stochastic game (Filar and Vrieze (1997), Basar and Olsder (1999)). Time is discrete and the horizon is infinite. There are N players. We let Ω denote the finite set of possible states; the state of the game in period t is $\omega_t \in \Omega$. In applications such as EP, ω_t is a vector partitioned into $(\omega_t^1, \dots, \omega_t^N)$, where ω_t^i denotes the (one or more) coordinates of the state that describe player i (e.g., the player's production capacity and/or product quality). We refer to ω_t^i as the state of player i and to ω_t as the state of the game.

Player i 's action (also called his/her control or policy) in period t is $x_t^i \in \mathbb{X}^i(\omega_t)$, where $\mathbb{X}^i(\omega_t)$ is the set of feasible actions for player i in state ω_t . We make no specific assumptions about $\mathbb{X}^i(\omega_t)$, which may be one- or multidimensional, discrete or continuous. The collection of players' actions in period t is $x_t = (x_t^1, \dots, x_t^N)$. We follow the usual convention of letting x_t^{-i} denote $(x_t^1, \dots, x_t^{i-1}, x_t^{i+1}, \dots, x_t^N)$. We assume that, in each period, players choose their actions simultaneously. Our game is therefore one of simultaneous moves.

The state follows a controlled discrete-time, finite-state, first-order Markov process. Specifically, if the state in period t is ω_t and the players choose actions x_t , then the probability that the state in period $t + 1$ is ω' is $\Pr(\omega' | \omega_t, x_t)$.

We decompose payoffs into two components. First, in period t , player i receives a payoff equal to $\pi^i(x_t, \omega_t)$ when players' actions are x_t and the state is ω_t . For example, if ω_t is a list of firms' capacities and x_t lists their output and investment decisions, then $\pi^i(x_t, \omega_t)$ represents firm i 's profit from product market competition net of investment expenses. Second, at the end of period t player i receives a payoff if there is a change in the state. Specifically, $\Phi^i(x_t, \omega_t, \omega_{t+1})$ is the change in the wealth of player i at the end of period t if the state moves from ω_t to $\omega_{t+1} \neq \omega_t$ (think of the transition as occurring at the end of the period) and players' actions were x_t .⁵ For example, if a firm searches for a buyer of a piece of equipment it wants to sell and sets a reservation price, both the search effort and the reservation price are coded in x_t^i . If the firm succeeds in finding an acceptable buyer, the state changes and the firm receives a payment equal to $\Phi^i(x_t, \omega_t, \omega_{t+1})$. In general, $\Phi^i(x_t, \omega_t, \omega_{t+1})$ depends on the nature of the transition (e.g., selling some or all equipment) and may be affected by the search effort of the firm prior to the sale as well as its reservation price. While $\pi^i(x_t, \omega_t)$ is paid out at the beginning of the period, we assume that $\Phi^i(x_t, \omega_t, \omega_{t+1})$ accrues at the end. This representation of payoffs allows us to capture many features of models of industry dynamics, including entry and exit.

Players discount future payoffs using a discount factor $\beta \in [0, 1)$. The objective of player i is to maximize the expected net present value of his/her future cash flows

$$\mathbb{E} \left\{ \sum_{t=0}^{\infty} \beta^t (\pi^i(x_t, \omega_t) + \beta \Phi^i(x_t, \omega_t, \omega_{t+1})) \right\},$$

⁵We set $\Phi^i(x_t, \omega_t, \omega_t) = 0$ without loss of generality.

where $\Phi^i(x_t, \omega_t, \omega_{t+1})$ is discounted (relative to $\pi^i(x_t, \omega_t)$) due to our assumption that it accrues at the end of the period after a change in the state has occurred.⁶

As is done in many applications of dynamic stochastic games, we focus on Markov perfect (a.k.a. feedback) equilibria. In period t , player i chooses an action x_t^i that depends solely on the current state ω_t . Formally, a Markovian strategy for player i maps the set of possible states Ω into his/her set of feasible actions $\mathbb{X}^i(\omega_t)$. Our solution concept is motivated by Bellman's (1957) principle of optimality: Given that all his/her rivals adopt a Markovian strategy, a player faces a dynamic programming problem and can do no better than to also adopt a Markovian strategy. Thus, a Markov perfect equilibrium remains a subgame perfect equilibrium even if the restriction to Markovian strategies is relaxed.

Let $V^i(\omega)$ denote the expected net present value of future cash flows to player i if the current state is ω . Suppose that the other players use strategies $X^{-i}(\omega)$. Then the Bellman equation for player i is

$$\begin{aligned} V^i(\omega) = & \max_{x^i \in \mathbb{X}^i(\omega)} \pi^i(x^i, X^{-i}(\omega), \omega) \\ & + \beta E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \}. \end{aligned} \quad (1)$$

The Bellman equation adds the current cash flow of player i , $\pi^i(x^i, X^{-i}(\omega), \omega)$, to the appropriately discounted expected future cash flow,

$$E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \},$$

where the expectation is taken over the successor states ω' . Player i 's strategy is given by

$$\begin{aligned} X^i(\omega) = & \arg \max_{x^i \in \mathbb{X}^i(\omega)} \pi^i(x^i, X^{-i}(\omega), \omega) \\ & + \beta E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \}. \end{aligned} \quad (2)$$

Each player has his/her own version of equations (1) and (3). The system of equations defined by the collection of (1) and (3) for each player $i = 1, \dots, N$ and each state $\omega \in \Omega$ defines a Markov perfect equilibrium in pure strategies.

Existence The extant literature provides a number of existence theorems for discrete-time stochastic games with either discrete (e.g., Fink (1964), Sobel (1971), Maskin and Tirole (2001)) or continuous actions (e.g., Federgruen (1978), Whitt (1980)). It invariably relies on mixed strategies to guarantee existence.

Computing mixed strategies over discrete actions (such as entry and exit in EP's framework) is challenging⁷ and computing mixed strategies over continuous actions (such as investment) is presently infeasible. Similar to PM1 and PM2, we therefore re-

⁶Discounting $\Phi^i(\cdot)$ is without loss of generality because it can always be replaced by $\tilde{\Phi}^i(\cdot) = \beta \Phi^i(\cdot)$, the net present value of $\Phi^i(\cdot)$ at the beginning of the period.

⁷In a survey of the literature, Breton (1991, p. 56) lamented, "In the zero-sum case, there exist reasonably efficient algorithms, but such is not the case in the general sum N -player case." Using a mathematical programming approach, he reported being able to solve, with considerable difficulty, discrete-time stochastic games with up to 3 players, 5 states, and 5 actions per player and state. Most recently, Herings and Peeters

strict attention to Markov perfect equilibria in pure strategies. There are very few general theorems that guarantee existence of such computationally tractable equilibria. Doraszelski and Satterthwaite (2010) showed how to reformulate EP's framework to guarantee existence. Subsequent work by Escobar (2008) covered general games with continuous actions.

In general, there may be multiple Markov perfect equilibria in pure strategies even after further restrictions such as symmetry and anonymity are imposed; see the examples in Doraszelski and Satterthwaite (2010).

2.2 Continuous-time model

We next describe the continuous-time stochastic game with a finite number of states.⁸ As in the discrete-time model, the horizon is infinite, the state of the game at time t is $\omega_t \in \Omega$, there are N players, and player i 's action at time t is denoted by $x_t^i \in \mathbb{X}^i(\omega_t)$. We further retain the assumption of simultaneous moves. In our continuous-time model, this means that players choose their actions simultaneously at each point in time.

The key difference is that the state in the continuous-time model follows a controlled continuous-time, finite-state Markov process. In discrete time, the time path of the state is a sequence, but in continuous time, the path is a piecewise-constant, right-continuous function of time. Jumps occur at random times according to a controlled Poisson process. At time t , the hazard rate of a jump occurring is $\phi(x_t, \omega_t) < \infty$.⁹ If a jump occurs at time t , then the probability that the state moves to ω' is $f(\omega' | \omega_{t-}, x_{t-})$, where $\omega_{t-} = \lim_{s \rightarrow t-} \omega_s$ is the state just before the jump and $x_{t-} = \lim_{s \rightarrow t-} x_s$ are players' actions just before the jump. That is, $f(\omega' | \omega_{t-}, x_{t-})$ characterizes the transitions of the induced first-order Markov process. Since a jump from a state to itself does not change the game, we simply ignore it and instead adjust, without loss of generality, the hazard rate of a jump occurring so that $f(\omega_{t-} | \omega_{t-}, x_{t-}) = 0$.

The payoff of player i consists of two components. First, player i receives a payoff flow equal to $\pi^i(x_t, \omega_t)$ when players' actions are x_t and the state is ω_t . Second, $\Phi^i(x_{t-}, \omega_{t-}, \omega_t)$ is the instantaneous change in the wealth of player i at time t if the state moves from ω_{t-} to $\omega_t \neq \omega_{t-}$ and players' actions just before the jump were x_{t-} . Like the discrete-time model, $\pi^i(x_t, \omega_t)$ may capture firm i 's profit from product market competition net of investment expenses and $\Phi^i(x_{t-}, \omega_{t-}, \omega_t)$ the scrap value that the firm receives upon exiting the industry or the setup cost that it incurs upon entering the industry. Unlike the discrete-time model, there is a clear-cut distinction between $\pi^i(x_t, \omega_t)$ and $\Phi^i(x_{t-}, \omega_{t-}, \omega_t)$ in the continuous-time model: $\pi^i(x_t, \omega_t)$ represents a flow of money, expressed in dollars per unit of time, whereas $\Phi^i(x_{t-}, \omega_{t-}, \omega_t)$ represents a change in the stock of wealth, expressed in dollars. As in the discrete-time

(2004) solved games with up to 5 players, 5 states, and 5 actions per player and state. The smallest applications of EP's framework have hundreds and the largest ones have millions of states. The sheer size of the state space alone thus makes them orders of magnitude too large for computing mixed-strategy equilibria.

⁸While less widely used in economics, continuous-time Markov decision problems are as tractable as their discrete-time counterparts and, dating back to Chapter 11 of Bellman (1957) and Chapter 8 of Howard (1960), have a common history in the mathematics and the operations research literatures.

⁹The assumption of a finite hazard rate rules out deterministic state-to-state transitions (see Section 6.4 for details).

game, this representation of payoffs can represent many dynamic phenomena; for example, the Online Appendix (available in a supplementary file on the journal website, <http://qeconomics.org/supp/153/supplement.pdf>) gives details on modeling entry and exit in our continuous-time game.

Players discount future payoffs using a discount rate $\rho > 0$. The objective of player i is to maximize the expected net present value of his/her future cash flows,

$$E \left\{ \int_0^\infty e^{-\rho t} \pi^i(x_t, \omega_t) dt + \sum_{m=1}^\infty e^{-\rho T_m} \Phi^i(x_{T_m^-}, \omega_{T_m^-}, \omega_{T_m}) \right\},$$

where T_m is the random time of the m th jump in the state, $x_{T_m^-}$ are players' actions just before the m th jump, $\omega_{T_m^-}$ is the state just before the m th jump, and ω_{T_m} is the state just after the m th jump.

The remaining aspects of the continuous-time model are similar to the discrete-time model. We again focus on Markov perfect equilibria. Thus, at time t , player i chooses an action x_t^i that depends solely on the current state ω_t .¹⁰ As in the discrete-time model, given that all his/her rivals adopt a Markovian strategy, a player can do no better than to also adopt a Markovian strategy. Furthermore, although the player gets to pick his/her action from scratch at each point in time, his/her optimal action changes only when the state of the game changes (under weak compactness and continuity assumptions; see Theorem 6.1 of [Feinberg \(2004\)](#)).

The Bellman equation for player i is similar to the one in discrete time (see [Bellman \(1957, pp. 83, 86–87\)](#)) for the statement and a formalization of the principle of optimality in a continuous-time setting). To see this, note that over a short interval of time of length $\Delta > 0$, the law of motion is

$$\Pr(\omega_{t+\Delta} \neq \omega_t | \omega_t, x_t) = \phi(x_t, \omega_t) \Delta + O(\Delta^2),$$

$$\Pr(\omega_{t+\Delta} = \omega' | \omega_t, x_t, \omega_{t+\Delta} \neq \omega_t) = f(\omega' | \omega_t, x_t) + O(\Delta).$$

Player i thus solves the dynamic programming problem given by

$$\begin{aligned} V^i(\omega) = & \max_{x^i \in X^i(\omega)} \pi^i(x^i, X^{-i}(\omega), \omega) \Delta \\ & + (1 - \rho \Delta) \{ (1 - \phi(x^i, X^{-i}(\omega), \omega) \Delta - O(\Delta^2)) V^i(\omega) \\ & + (\phi(x^i, X^{-i}(\omega), \omega) \Delta + O(\Delta^2)) \\ & \times (E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \} + O(\Delta)) \}, \end{aligned} \quad (3)$$

¹⁰As is well known, in continuous time there is no natural notion of “a last time before time t ,” thus rendering induction inapplicable. Because induction is fundamental to defining decision trees, strategies, and outcomes in discrete-time games, in general these notions do not have direct continuous-time analogs and numerous technical difficulties ensue (see, e.g., [Simon and Stinchcombe \(1989\)](#)). To avoid them, we focus on Markov perfect equilibria and assume a finite hazard rate. Consider playing tit-for-tat in a prisoner's dilemma. Due to the latter assumption, one cannot construct a state variable that indicates whether a player has cooperated at all times before time t ; the most one can do is to loosely track past behavior by having the state variable change with a finite hazard rate as the player switches from cooperation to defection.

which, as $\Delta \rightarrow 0$, simplifies to the Bellman equation

$$\begin{aligned} \rho V^i(\omega) = & \max_{x^i \in \mathbb{X}^i(\omega)} \pi^i(x^i, X^{-i}(\omega), \omega) - \phi(x^i, X^{-i}(\omega), \omega) V^i(\omega) \\ & + \phi(x^i, X^{-i}(\omega), \omega) \\ & \times E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \}. \end{aligned} \quad (4)$$

Hence, $V^i(\omega)$ can be interpreted as the asset value to player i of participating in the game. This asset is priced by requiring that the opportunity cost of holding it, $\rho V^i(\omega)$, equals the current cash flow, $\pi^i(x^i, X^{-i}(\omega), \omega)$, plus the expected capital gain or loss conditional on a jump occurring,

$$E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \} - V^i(\omega),$$

times the hazard rate of a jump occurring, $\phi(x^i, X^{-i}(\omega), \omega)$. Similar to the discrete-time model, player i 's strategy is found by carrying out the maximization on the right-hand side of the Bellman equation (4).

Existence For the same reason as in the discrete-time model, computational tractability requires the existence of a Markov perfect equilibrium in pure strategies. In what follows, we provide sufficient conditions for the existence of such an equilibrium.

We focus our attention on games with continuous actions.

ASSUMPTION 1. $\mathbb{X}^i(\omega)$ is nonempty, compact, and convex for all ω and i .

Next we assume that players discount future payoffs.

ASSUMPTION 2. $\rho > 0$.

We further assume that the model's primitives are continuous.

ASSUMPTION 3. $\pi^i(x, \omega)$, $\Phi^i(x, \omega, \omega')$, $\phi(x, \omega)$, and $f(x, \omega)$ are continuous in x for all ω , ω' , and i .

Similar continuity assumptions are commonplace in the literature on discrete-time stochastic games (see [Mertens \(2002\)](#) for a survey).

Let $V^i(\cdot)$ denote a $(|\Omega| \times 1)$ vector of values of player i in the various possible states and let $h^i(x^i, X^{-i}(\omega), \omega, V^i(\cdot))$ denote the maximand in the Bellman equation (4) for player i . To guarantee existence in pure strategies, we finally assume that player i 's maximization problem always has a unique solution.

ASSUMPTION 4. $\arg \max_{x^i \in \mathbb{X}^i(\omega)} h^i(x^i, X^{-i}(\omega), \omega, V^i(\cdot))$ is single-valued for all $X^{-i}(\omega)$, ω , $V^i(\cdot)$, and i .

Note that we require that the best reply is unique for arbitrary policies $X^{-i}(\omega)$ of the rivals and for arbitrary values $V^i(\cdot)$ of the player, both in and out of equilibrium. A sufficient condition for Assumption 4 to hold is that $h^i(\cdot)$ is strictly quasiconcave in x^i for all $X^{-i}(\omega)$, ω , $V^i(\cdot)$, and i .

Our Assumption 4 is the exact analog of an assumption that Doraszelski and Satterthwaite (2010) make in the context of discrete-time stochastic games. The main work there is to provide sufficient conditions in terms of the model's primitives for Assumption 4 to hold. While this is beyond the scope of the present paper, we note that in concrete examples Assumption 4 is often easily verified. In particular, it holds for the quality ladder model that we use in Section 4 to illustrate the computational advantages of continuous time as well as for the variants of the model with entry and exit that we describe in the Online Appendix.

The above assumptions ensure the existence of a computationally tractable equilibrium.

PROPOSITION 1. *Under Assumptions 1, 2, 3, and 4, there exists a Markov perfect equilibrium in pure strategies.*

The proof is provided in the Appendix.

The fact that the continuous-time Bellman equation (4) is the limit of equation (3) does not imply that the equilibria of a sequence of discrete-time games converge to the equilibria of the continuous-time game.¹¹

2.3 Avoiding the curse of dimensionality

Computing a Markov perfect equilibrium requires computing the expectation over successor states. Setting $\Phi^i(X(\omega), \omega, \omega') = 0$ and $x^i = X^i(\omega)$ to simplify the notation, in the discrete-time Bellman equation (1) this expectation is

$$E_{\omega'}\{V^i(\omega')|\omega, X(\omega)\} = \sum_{\{\omega': \Pr(\omega'|\omega, X(\omega)) > 0\}} V^i(\omega') \Pr(\omega'|\omega, X(\omega)). \quad (5)$$

In the continuous-time Bellman equation (4) we have an analogous expression with $f(\omega'|\omega, X(\omega))$ replacing $\Pr(\omega'|\omega, X(\omega))$. Computing the expectation over successor states therefore involves summing over all states ω' such that $\Pr(\omega'|\omega, X(\omega)) > 0$ in the discrete-time model or $f(\omega'|\omega, X(\omega)) > 0$ in the continuous-time model. Without additional structure, there is clearly no reason for either model to have any computational advantages. In particular, continuous time by itself is not sufficient to avoid the curse of dimensionality in computing expectations.

Under widely used laws of motion, however, the expectation over successor states is substantially less burdensome to compute in continuous than in discrete time. This is easiest to see—but extends beyond—the special case of independent transitions. In the

¹¹This observation cautions against estimating the primitives from a discrete-time model and then plugging the estimated primitives into the continuous-time model to compute equilibria.

literature following EP, it is commonly specified that firm i 's state evolves as

$$(\omega')^i = \omega^i + \tau^i - \eta^i, \quad (6)$$

where the discrete random variables $\{\tau^i\}_{i=1}^N$ and $\{\eta^i\}_{i=1}^N$ are mutually independent. Typically $\tau^i \in \{0, 1\}$ is governed by firm i 's investment decision and $\eta^i \in \{0, 1\}$ is a firm-specific depreciation shock (e.g., Besanko and Doraszelski (2004), Chen (2009), Doraszelski and Markovich (2007)). This specification may be appropriate in modeling capacity, advertising, or research and development, where investment successes and setbacks are idiosyncratic.

In the special case of independent transitions, the transition probability of the controlled discrete-time Markov process can be written as

$$\Pr(\omega'|\omega, x) = \prod_{i=1}^N \Pr^i((\omega')^i|\omega, x), \quad (7)$$

where $\Pr^i((\omega')^i|\omega, x)$ is the transition probability for player i 's state and may depend on the states and actions of all players, including those of player i 's rivals. Substituting equation (7) into equation (5) shows that if each player can move to one of K states, then the expectation over successor states involves summing over K^N terms and grows exponentially in N . Hence, in the special case of independent transitions, the discrete-time model suffers from a curse of dimensionality.

The continuous-time model avoids this curse in computing the expectation over successor states. In the special case of independent transitions, the hazard rate of a jump in the state of player i occurring is $\phi^i(x, \omega) < \infty$, and if a jump occurs, then the probability that the state of player i moves to $(\omega')^i$ is $f^i((\omega')^i|\omega, x)$. Over a short interval of time of length $\Delta > 0$, player i 's state evolves according to

$$\begin{aligned} \Pr^i(\omega_{t+\Delta}^i \neq \omega_t^i | \omega_t, x_t) &= \phi^i(x_t, \omega_t) \Delta + O(\Delta^2), \\ \Pr^i(\omega_{t+\Delta}^i = (\omega')^i | \omega_t, x_t, \omega_{t+\Delta}^i \neq \omega_t^i) &= f^i((\omega')^i | \omega_t, x_t) + O(\Delta), \end{aligned}$$

and the hazard rate of a change in the state of the game at time t is therefore $\phi(x_t, \omega_t) = \sum_{i=1}^N \phi^i(x_t, \omega_t)$. This last equality reveals a critical fact about continuous-time Markov processes: in the special case of independent transitions, during a short interval of time, there will be (with probability infinitesimally close to 1) a jump in the state of at most one player. In the discrete-time model, we must keep track of all possible combinations of players' transitions between time t and time $t+1$. The possibility of two or more players' states changing simultaneously disappears in the continuous-time model; this results in a simpler and computationally much more tractable model.

Indeed, in the special case of independent transitions, the Bellman equation (4) of player i can be written as

$$\begin{aligned} \rho V^i(\omega) &= \max_{x^i \in \mathbb{X}^i(\omega)} \pi^i(x^i, X^{-i}(\omega), \omega) - \phi(x^i, X^{-i}(\omega), \omega) V^i(\omega) \\ &\quad + \sum_{j=1}^N \phi^j(x^i, X^{-i}(\omega), \omega) \end{aligned}$$

$$\begin{aligned} & \times E_{(\omega')^j} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, (\omega')^j, \omega^{-j}) \\ & + V^i((\omega')^j, \omega^{-j}) | \omega, x^i, X^{-i}(\omega) \}. \end{aligned}$$

The N -dimensional expectation over successor states in equation (4) decomposes into N one-dimensional expectations given by

$$\begin{aligned} & E_{(\omega')^j} \{ V^i((\omega')^j, \omega^{-j}) | \omega, X(\omega) \} \\ & = \sum_{\{(\omega')^j: f^j((\omega')^j | \omega, X(\omega)) > 0\}} V^i((\omega')^j, \omega^{-j}) f^j((\omega')^j | \omega, X(\omega)), \end{aligned}$$

where again we set $\Phi^i(X(\omega), \omega, \omega') = 0$ and $x^i = X^i(\omega)$ to simplify the notation. Hence, if each player can move to one of K states, then computing the expectation over successor states involves summing over $(K - 1)N$ terms in the continuous-time model but K^N terms in the discrete-time model.¹² Since $(K - 1)N$ grows linearly rather than exponentially with N , computing the expectation over successor states in the special case of independent transitions is no longer subject to the curse of dimensionality.

The computational advantages of continuous time stem from the fact that the possibility of two or more players' states changing simultaneously disappears. While this possibility can be reintroduced into the continuous-time model, the resulting stochastic process is not comparable to the one underlying EP's framework. An example makes this point. There are two firms and two states per firm, so the state space is $\Omega = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. A firm may move up one level but never back, so $\tau^i \in \{0, 1\}$ and $\eta^i = 0$ in the law of motion in equation (6).

Letting $\Pr(\tau^i = 1) = p$, the transition probability matrix of the discrete-time Markov chain is

$$P = \begin{pmatrix} (1-p)^2 & p(1-p) & p(1-p) & p^2 \\ 0 & 1-p & 0 & p \\ 0 & 0 & 1-p & p \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

It follows that ω_t^1 and ω_t^2 are uncorrelated in period t . At $t = 1$, for example, we have

$$\text{Cov}(\omega_1^1, \omega_1^2 | \omega_0^1 = 0, \omega_0^2 = 0) = P[1, 4] - (P[1, 2] + P[1, 4])(P[1, 3] + P[1, 4]) = 0,$$

where $P[R, C]$ denotes the element in row R and column C of the matrix P .

Turning to continuous time, let λ be the hazard rate of one firm moving up. In addition, let ζ be the hazard rate of both firms moving up. The infinitesimal generator (rate matrix) of the continuous-time Markov chain is

$$Q = \begin{pmatrix} -2\lambda - \zeta & \lambda & \lambda & \zeta \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

¹²Here we exploit the fact that, unlike in the discrete-time model, there is no need to explicitly consider the possibility of remaining in the same state.

and the one-period-ahead transition probability matrix is

$$P = \exp(Q)$$

$$= \begin{pmatrix} e^{-\zeta-2\lambda} & \frac{e^{-\zeta-2\lambda}(-1+e^{\zeta+\lambda})\lambda}{\zeta+\lambda} & \frac{e^{-\zeta-2\lambda}(-1+e^{\zeta+\lambda})\lambda}{\zeta+\lambda} & \frac{-e^{-\zeta-2\lambda}\zeta+\zeta+e^{-\zeta-2\lambda}\lambda-2e^{-\lambda}\lambda+\lambda}{\zeta+\lambda} \\ 0 & e^{-\lambda} & 0 & 1-e^{-\lambda} \\ 0 & 0 & e^{-\lambda} & 1-e^{-\lambda} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

As long as $\zeta \neq 0$ and players' states can change simultaneously, ω_t^1 and ω_t^2 are correlated at time t . At $t = 1$, for example, we have, in contrast to the discrete-time stochastic process,

$$\begin{aligned} \text{Cov}(\omega_1^1, \omega_1^2 | \omega_0^1 = 0, \omega_0^2 = 0) \\ &= P[1, 4] - (P[1, 2] + P[1, 4])(P[1, 3] + P[1, 4]) \\ &= \frac{e^{-2(\zeta+2\lambda)}((-1+e^{\zeta+2\lambda})\zeta^2 + 2e^{\zeta+\lambda}(-1+e^\lambda)\lambda\zeta - e^{\zeta+2\lambda}(-1+e^\lambda)\lambda^2)}{(\zeta+\lambda)^2} \neq 0 \end{aligned}$$

unless $\zeta = 0$. Hence, as long as $\zeta \neq 0$ and players' states can change simultaneously, the continuous-time stochastic process is not comparable to EP's framework.

Common shocks The computational advantages of continuous time extend beyond the special case of independent transitions. The other widely used law of motion in the literature following EP holds that firm i 's state evolves according to

$$(\omega')^i = \omega^i + \tau^i - \eta, \quad (8)$$

where the discrete random variables $\{\tau^i\}_{i=1}^N$ and η are mutually independent. The common shock η induces correlation in players' states beyond that induced by strategic interactions. Typically $\tau^i \in \{0, 1\}$ is governed by firm i 's investment decision and $\eta \in \{0, 1\}$ is an industrywide depreciation shock. In the quality ladder model of PM1, $\eta = 1$ represents an increase in the quality of the outside good that, given the functional form of demand, is equivalent to a decrease in the qualities of all inside goods. In the model of dynamic cost competition in EP, $\eta = 1$ represents an increase in factor prices. [Berry and Pakes \(1993\)](#), [Gowrisankaran \(1999a\)](#), [Fershtman and Pakes \(2000\)](#), [de Roos \(2004\)](#), and [Markovich \(2008\)](#), among others, also use the law of motion in equation (8).

The discrete-time model again suffers from a curse of dimensionality that the continuous-time model avoids. Suppose the common shock η has L possible levels and, conditional on η , each player can move to one of K states. Computing the expectation over successor states in the discrete-time model involves summing over LK^N terms (conditional on η , the expectation has K^N terms) compared to $L - 1 + (K - 1)N$ terms in the continuous-time model ($L - 1$ terms for the common shock plus $(K - 1)N$ terms as above). In the Online Appendix, we provide further details on modeling an industrywide depreciation shock.

Common states Similarly to common shocks, common states that affect the current payoffs of all players are computationally more burdensome in the discrete- than in the continuous-time model. Suppose, for example, that in addition to players' states that describe firm-specific production capacities, there is a common state such as industry demand that evolves independently of production capacities (e.g., [Besanko and Doraszelski \(2004\)](#), [Besanko, Doraszelski, Lu, and Satterthwaite \(2010b\)](#)). If the common state can move to L possible levels and each player can move to one of K states, then the summation is over LK^N terms in discrete time but $L - 1 + (K - 1)N$ terms in continuous time.

Multiple states per player The curse of dimensionality becomes even more severe in applications where each player is described by $D > 1$ coordinates of the state (e.g., [Benkard \(2004\)](#), [Langohr \(2003\)](#)). Returning to the special case of independent transitions, computing the expectation over successor states in the discrete-time model involves summing over K^{ND} terms compared to $(K - 1)ND$ terms in the continuous-time model. What matters is the total number of coordinates of the state vector. The curse of dimensionality is just as severe in a single-agent dynamic programming problem with an ND -dimensional state vector as in an N -player discrete-time stochastic game with an ND -dimensional state vector.

3. COMPUTATIONAL STRATEGIES

Next we present our computational strategies for the discrete- and continuous-time models. Our approach is similar to PM1, the most often used algorithm in the literature following EP to solve numerically for a Markov perfect equilibrium. The more general observation is that computing an equilibrium is just the problem of solving a large system of nonlinear equations. If the size of the problem is very large, then a direct application of Newton's method or other solution methods for nonlinear equations is typically impractical, and some type of Gaussian method is necessary. The idea behind Gaussian methods is that it is harder to solve a large system of equations once than to solve smaller systems many times. Thus it may be advantageous to break up the large system into small pieces.

As we showed in Section 2.3, the fact that the possibility of two or more players' states changing simultaneously disappears in continuous time under widely used laws of motion results in a simpler model. Since the equations that characterize the equilibrium are simpler, the computational advantages of continuous time are not tied to a particular algorithm. Any algorithm that uses these equations stands to benefit. This includes the direct application of solution methods for nonlinear equations in [Ferris, Judd, and Schmedders \(2007\)](#) and the path-following or homotopy methods in [Besanko, Doraszelski, Kryukov, and Satterthwaite \(2010\)](#) (see also [Borkovsky, Doraszelski, and Kryukov \(2010\)](#)).

3.1 Discrete-time algorithm

The algorithm is iterative. First we order the states in Ω and make initial guesses for the value $V^i(\omega)$ and the policy $X^i(\omega)$ of each player $i = 1, \dots, N$ in each state $\omega \in \Omega$. Then

we update these guesses as we proceed through the state space in the prespecified order. Specifically, in state $\omega \in \Omega$, given old guesses $V^i(\omega)$ and $X^i(\omega)$, we compute new guesses $\hat{V}^i(\omega)$ and $\hat{X}^i(\omega)$ for each player $i = 1, \dots, N$ as

$$\begin{aligned} \hat{X}^i(\omega) \leftarrow & \arg \max_{x^i} \pi^i(x^i, X^{-i}(\omega), \omega) \\ & + \beta E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \}, \end{aligned} \quad (9)$$

$$\begin{aligned} \hat{V}^i(\omega) \leftarrow & \pi^i(\hat{X}^i(\omega), X^{-i}(\omega), \omega) \\ & + \beta E_{\omega'} \{ \Phi^i(\hat{X}^i(\omega), X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, \hat{X}^i(\omega), X^{-i}(\omega) \}. \end{aligned} \quad (10)$$

Note that the old guesses for the policies of player i 's opponents, $X^{-i}(\omega)$, and the old guess for player i 's value, $V^i(\omega)$, are used when computing the new guesses $\hat{V}^i(\omega)$ and $\hat{X}^i(\omega)$. This procedure is, therefore, a Gauss–Jacobi scheme at each state $\omega \in \Omega$.

There are two ways to update the old guesses $V^i(\omega)$ and $X^i(\omega)$. PM1 suggest a Gauss–Jacobi scheme that computes the new guesses $\hat{V}^i(\omega)$ and $\hat{X}^i(\omega)$ for all players $i = 1, \dots, N$ and *all* states $\omega \in \Omega$ before replacing the old guesses with the new guesses. Their value function iteration approach is also called a pre-Gauss–Jacobi method in the literature on nonlinear equations (see Judd (1998) for an extensive discussion of Gauss–Jacobi and Gauss–Seidel methods). In contrast to PM1, we employ the block Gauss–Seidel scheme that is typically used for discrete-time stochastic games with a finite number of states (e.g., Benkard (2004)). In our block Gauss–Seidel scheme, immediately after computing $\hat{V}^i(\omega)$ and $\hat{X}^i(\omega)$ for all players $i = 1, \dots, N$ and a *given* state $\omega \in \Omega$, we replace the old guesses with the new guesses for all players in that state. This has the advantage that “information” is used as soon as it becomes available. The algorithm cycles through the state space until the changes in the value and policy functions are small (see Section 5.4 for details).

3.2 Continuous-time algorithm

In its basic form, our computational strategy adapts the block Gauss–Seidel scheme to the continuous-time model. The sole change is that to update players' values and policies in state $\omega \in \Omega$, we replace equations (9) and (10) by

$$\begin{aligned} \hat{X}^i(\omega) \leftarrow & \arg \max_{x^i} \pi^i(x^i, X^{-i}(\omega), \omega) - \phi(x^i, X^{-i}(\omega), \omega) V^i(x^i, X^{-i}(\omega), \omega) \\ & + \phi(x^i, X^{-i}(\omega), \omega) \\ & \times E_{\omega'} \{ \Phi^i(x^i, X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, x^i, X^{-i}(\omega) \}, \end{aligned} \quad (11)$$

$$\begin{aligned} \hat{V}^i(\omega) \leftarrow & \frac{1}{\rho + \phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)} \pi^i(\hat{X}^i(\omega), X^{-i}(\omega), \omega) \\ & + \frac{\phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)}{\rho + \phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)} \\ & \times E_{\omega'} \{ \Phi^i(\hat{X}^i(\omega), X^{-i}(\omega), \omega, \omega') + V^i(\omega') | \omega, \hat{X}^i(\omega), X^{-i}(\omega) \}. \end{aligned} \quad (12)$$

The remainder of the algorithm proceeds as before. Note that by dividing through by $\rho + \phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)$, we ensure that equation (12) is contractive for a given player (holding fixed the policies of all players) since

$$\frac{\phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)}{\rho + \phi(\hat{X}^i(\omega), X^{-i}(\omega), \omega)} < 1$$

as long as the hazard rate is bounded above. Note that the contraction factor varies with players' policies. In the discrete-time model, by contrast, the contraction factor equals the discount factor β . Unfortunately, the system of equations that defines the equilibrium is not contractive, and hence neither our continuous- nor our discrete-time algorithm is guaranteed to converge.

3.3 Precomputed addresses, symmetry, and anonymity

The first advantage of continuous time is that under widely used laws of motion, it avoids the curse of dimensionality in computing the expectation over successor states. We next describe a way to further speed up this computation. To understand this suggestion, we need to briefly discuss the nuts and bolts of computer storage. Any algorithm must store the value and policy functions in some table that we denote $\mathbb{M}$. Each row of this table corresponds to a state $\omega \in \Omega$ and contains the vector $(V^1(\omega), \dots, V^N(\omega), X^1(\omega), \dots, X^N(\omega))$ of values and policies for all players in that state. Consider the expectation over successor states in the discrete-time model as given by equation (5). To compute this sum, the algorithm must find the rows and columns with the relevant information in table $\mathbb{M}$, implying that the sum is really

$$\begin{aligned} & \sum_{\{\omega': \Pr(\omega'|\omega, \mathbb{M}[R(\omega), (N+1, \dots, 2N)]) > 0\}} \mathbb{M}[R(\omega'), C(\omega', i)] \\ & \times \Pr(\omega'|\omega, \mathbb{M}[R(\omega), (N+1, \dots, 2N)]), \end{aligned} \quad (13)$$

where $C(\omega', i)$ is the column in row $R(\omega')$ that contains the value for player i in state ω' and $N+1, \dots, 2N$ are the columns in row $R(\omega)$ that contain the policies for players $j = 1, \dots, N$ in state ω . Equation (13) displays all the computations that must occur in evaluating $E_{\omega'}\{V^i(\omega')|\omega, X(\omega)\}$ and emphasizes that there are two kinds of costs involved: The first is the summation over all states ω' such that $\Pr(\omega'|\omega, X(\omega)) > 0$; the second is the computation of the address, $R(\omega')$ and $C(\omega', i)$, of the value of player i at each of them. One way to reduce running times is to precompute these addresses and store them along with the values and policies for state ω . More precisely, for each successor state ω' of state ω , we append a vector $(R(\omega'), C(\omega', 1), \dots, C(\omega', N))$ of precomputed addresses to the vector $(V^1(\omega), \dots, V^N(\omega), X^1(\omega), \dots, X^N(\omega))$ of values and policies.

Precomputed addresses decrease running times but increase memory requirements since $N+1$ numbers need to be stored for each successor state. The practicality of this computational strategy thus hinges on the number of successor states. As we showed in Section 2.3, under widely used laws of motion, this number is much smaller in the

continuous- than in the discrete-time model. Precomputed addresses are therefore essentially only practical in continuous time.

The usefulness of precomputed addresses further depends on how hard it is to evaluate $R(\omega)$ and $C(\omega, i)$. Evaluating $R(\omega)$ and $C(\omega, i)$ is complicated when attention is restricted to symmetric and anonymous equilibria, as is often done in applications of EP's framework to slow down the growth of the state space in the number of players N and the number of states per player M . Symmetry allows us to focus on the problem of player 1 and anonymity (also called exchangeability) says that player 1 does not care about the identity of his/her rivals, only about the distribution of their states (see, e.g., [Doraszelski and Satterthwaite \(2010\)](#) for a formal definition). In practice, symmetry and anonymity are imposed by limiting the computation of players' values and policies to states in the set $\tilde{\Omega} = \{(\omega^1, \omega^2, \dots, \omega^N) \in \Omega : \omega^1 \leq \omega^2 \leq \dots \leq \omega^N\}$.¹³ Whereas Ω grows exponentially in N , $\tilde{\Omega}$ grows polynomially. More specifically, the number of states to be examined is reduced from $|\Omega| = M^N$ to $|\tilde{\Omega}| = \frac{(N+M-1)!}{N!(M-1)!}$.¹⁴ Pakes, Gowrisankaran, and McGuire (1993) and Gowrisankaran (1999b) proposed slightly different methods for mapping the elements of $\tilde{\Omega}$ into consecutive integers. These methods form the basis for computing $R(\omega)$, but require that $\omega \in \tilde{\Omega}$. While this is achieved by sorting the coordinates of the vector ω , sorting implies that $C(\omega, i)$ is no longer always equal to i : Suppose that the state of the game is $(1, 1, 3)$ and that firm 1 moves to state 2. Hence, the state becomes $(2, 1, 3)$ or, after sorting, $(1, 2, 3)$, so that $C((2, 1, 3), 1) = 2$, $C((2, 1, 3), 2) = 1$, and $C((2, 1, 3), 3) = 3$. Since evaluating $R(\omega)$ and $C(\omega, i)$ is rather involved, there is a lot to be gained from precomputed addresses; see Section 5.

4. EXAMPLE: THE PAKES AND MCGUIRE (1994) QUALITY LADDER MODEL

We use the quality ladder model developed by PM1 to demonstrate the computational advantages of continuous time in Section 5 and to illustrate the conceptual differences between discrete and continuous time in Section 6. Below we first describe their model and then reformulate it in continuous time. To focus on the key issue related to the curse of dimensionality in discrete-time models, we abstract from entry and exit in what follows and set $\Phi^i(x, \omega, \omega') = 0$. We also differ from PM1 in that our depreciation shocks are independent across firms as in the law of motion in equation (6), whereas PM1 assumed an industrywide depreciation shock as in equation (8). In the Online Appendix, we describe how to add entry and exit either by way of an entry/exit intensity or by way of randomly drawn, privately observed setup costs/scrap values (as in [Doraszelski and Satterthwaite \(2010\)](#)), and how to model an industrywide depreciation shock in continuous time.

¹³Some additional restrictions are needed to obtain a symmetric and anonymous equilibrium. If $N = 2$, for example, symmetry requires that $V^1(1, 1) = V^2(1, 1)$.

¹⁴Symmetry and anonymity can in some cases also reduce the number of terms involved in computing the expectation over successor states in the discrete-time model. For example, if $\omega = (2, 3)$ and a player can move down one level, stay the same, or move up one level, then there are 8 successor states $\omega' \in \{(1, 2), (2, 2), (1, 3), (2, 3), (3, 3), (1, 4), (2, 4), (3, 4)\}$ with two distinct ways to reach state $\omega' = (2, 3)$.

The existence of a Markov perfect equilibrium in pure strategies follows from Doraszelski and Satterthwaite (2010) for the discrete-time model. For the continuous-time model, we demonstrate in the Online Appendix that Proposition 1 applies to the model with entry and exit; a similar argument ensures existence for the simpler model that we use below.

4.1 Discrete-time model

The quality ladder model assumes that there are N firms with vertically differentiated products engaged in price competition. Firm i produces a product of quality $\omega^i \in \{1, \dots, M\}$. The state space is $\Omega = \{1, \dots, M\}^N$. We first describe price competition and then turn to quality dynamics.

Demand Each consumer purchases at most one unit of one product. The utility consumer k derives from purchasing product i is $g(\omega^i) - p^i + \varepsilon^{ik}$, where

$$g(\omega^i) = \begin{cases} 3\omega^i - 4, & \omega^i \leq 5, \\ 12 + \ln(2 - \exp(16 - 3\omega^i)), & \omega^i > 5, \end{cases}$$

maps the quality of the product into the consumer's valuation for it and ε^{ik} represents taste differences among consumers. There is a no-purchase alternative, product 0, which has utility ε^{0k} . We assume that the idiosyncratic shocks $\varepsilon^{0k}, \varepsilon^{1k}, \dots, \varepsilon^{Nk}$ are independently and identically extreme value distributed across products and consumers; therefore, the demand for firm i 's product is

$$q^i(p^1, \dots, p^N; \omega) = m \frac{\exp(g(\omega^i) - p^i)}{1 + \sum_{j=1}^N \exp(g(\omega^j) - p^j)},$$

where $m > 0$ is the size of the market (the measure of consumers).

Price competition In each period, firm i observes the quality of its and its rivals' products, and chooses the price p^i of product i to maximize profits, thereby solving

$$\max_{p^i \geq 0} q^i(p^1, \dots, p^N; \omega)(p^i - c),$$

where $c \geq 0$ is the marginal cost of production. Given a state $\omega \in \Omega$, there exists a unique Nash equilibrium $(p^1(\omega), \dots, p^N(\omega))$ of the product market game (Caplin and Nalebuff (1991)). It is found easily by numerically solving the system of first-order conditions corresponding to firms' profit-maximization problems.

Law of motion Firm i 's state ω^i represents the quality of its product in the present period. The quality of firm i 's product in the subsequent period is governed by its investment $x^i \geq 0$ in quality improvements $\tau^i \in \{0, 1\}$ and by depreciation $\eta^i \in \{0, 1\}$. If the investment is successful, then quality increases by one level. The probability of success is $\frac{\alpha x^i}{1 + \alpha x^i}$, where $\alpha > 0$ is a measure of the effectiveness of investment. With probability $\delta \in [0, 1]$, the firm is hit by a depreciation shock and quality decreases by one level.

Combining the investment and depreciation processes, if $\omega^i \in \{2, \dots, M-1\}$, then the quality of firm i 's product changes according to the transition probability

$$\Pr^i((\omega')^i | \omega, x) = \begin{cases} \frac{(1-\delta)\alpha x^i}{1+\alpha x^i}, & (\omega')^i = \omega^i + 1, \\ \frac{1-\delta+\delta\alpha x^i}{1+\alpha x^i}, & (\omega')^i = \omega^i, \\ \frac{\delta}{1+\alpha x^i}, & (\omega')^i = \omega^i - 1. \end{cases}$$

Since firm i cannot move further down (up) from the lowest (highest) product quality, we set

$$\Pr^i((\omega')^i | 1, \omega^{-i}, x) = \begin{cases} \frac{(1-\delta)\alpha x^i}{1+\alpha x^i}, & (\omega')^i = 2, \\ \frac{1+\delta\alpha x^i}{1+\alpha x^i}, & (\omega')^i = 1, \end{cases}$$

$$\Pr^i((\omega')^i | M, \omega^{-i}, x) = \begin{cases} \frac{1-\delta+\alpha x^i}{1+\alpha x^i}, & (\omega')^i = M, \\ \frac{\delta}{1+\alpha x^i}, & (\omega')^i = M-1. \end{cases}$$

Payoff function The per-period payoff of firm i is derived from the Nash equilibrium of the product market game and is given by

$$\pi^i(x, \omega) \equiv q^i(p^1(\omega), \dots, p^N(\omega); \omega)(p^i(\omega) - c) - x^i,$$

where we have subtracted investment x^i from the profit from price competition.

Parameterization As in PM1, the size of the market is $m = 5$, the marginal cost of production is $c = 5$, the effectiveness of investment is $\alpha = 3$, and the depreciation probability is $\delta = 0.7$. We again follow PM1 in first assuming that the discount factor is $\beta = 0.925$, which corresponds to a yearly interest rate of 8.1%, and that the number of quality levels per firm is $M = 18$, but we also examine other values for β and M in Section 5.

4.2 Continuous-time model

In the interest of brevity, we start by noting that the details of price competition remain unchanged. In the continuous-time model, we can thus reinterpret $\pi^i(x, \omega)$ as the payoff flow of firm i .

Law of motion To make the continuous- and discrete-time models easily comparable, we take the hazard rate for the investment project of firm i being successful in the continuous-time model to be $\frac{\alpha x^i}{1+\alpha x^i}$, the same choice as for the success probability in the discrete-time model. This is appropriate since the expected time to the first success is $\frac{1+\alpha x^i}{\alpha x^i}$ in both models (although the variance is generally higher in the continuous-time model). Moreover, our choice of functional form for the success hazard ensures

that the marginal incentive to invest in quality improvements—and therefore the level of investment—is similar in the continuous- and discrete-time models. Similarly, the depreciation hazard in the continuous-time model equals the depreciation probability, δ , in the discrete-time model.¹⁵

Jumps in firm i 's state thus occur according to a Poisson process with hazard rate

$$\phi^i(x, \omega) = \frac{\alpha x^i}{1 + \alpha x^i} + \delta,$$

and when a jump occurs, firm i 's state changes according to the transition probability

$$f^i((\omega')^i | \omega, x) = \begin{cases} \frac{\alpha x^i}{(1 + \alpha x^i)\phi^i(x, \omega)}, & (\omega')^i = \omega^i + 1, \\ \frac{\delta}{\phi^i(x, \omega)}, & (\omega')^i = \omega^i - 1, \end{cases}$$

if $\omega^i \in \{2, \dots, M - 1\}$. Since firm i cannot move further down (up) from the lowest (highest) product quality, we set

$$\begin{aligned} \phi^i(x, 1, \omega^{-i}) &= \frac{\alpha x^i}{1 + \alpha x^i}, & f^i(2 | 1, \omega^{-i}, x) &= 1, \\ \phi^i(x, M, \omega^{-i}) &= \delta, & f^i(M - 1 | M, \omega^{-i}, x) &= 1. \end{aligned}$$

Parameterization Whenever possible, we use the same parameter values in the continuous- as in the discrete-time model. Moreover, we can easily match the discrete-time discount factor β to the continuous-time discount rate ρ : if Δ is the unit of time in the discrete-time model, then β and ρ are related by $\beta = e^{-\rho\Delta}$ or, equivalently, by $\rho = -\frac{\ln \beta}{\Delta}$. We take $\Delta = 1$ to obtain $\rho = -\ln \beta$.

5. COMPUTATIONAL ADVANTAGES OF CONTINUOUS TIME

This section illustrates the computational advantages of continuous time using the quality ladder model of Section 4 as an example. Even though this is one specific example, it is useful for many purposes. First, the results pertaining to the curse of dimensionality are clearly robust since they simply involve the floating point operations in computing the expectation over successor states. The burden of such computations depends on neither functional forms nor parameter values. Also, as we have pointed out in Section 2.3, what matters is the total number of coordinates of the state vector. Hence, the

¹⁵Another possibility is to take the success and depreciation hazards to be $\ln(1 + \alpha x^i)$ and $-\ln(1 - \delta)$, respectively. This choice ensures that the probability of an investment success between time t and time $t + 1$ is the same as in the discrete-time model. However, because $\frac{\alpha x^i}{1 + \alpha x^i}$ but not $\ln(1 + \alpha x^i)$ is bounded above, the marginal incentive to invest—and therefore the level of investment—is higher in the continuous-time model. In practice, these comparability considerations play little role and continuous time provides greater freedom in choosing functional forms. For example, $(x^i)^\gamma$ is a familiar constant elasticity form that can be used in a continuous- but not in a discrete-time model. The parameter γ has a clear-cut interpretation as the elasticity of the success hazard with respect to investment expenditures or, equivalently, (the negative of) the elasticity of the expected time to an investment success.

N -firm quality ladder model should be viewed as representative of dynamic stochastic games with N -dimensional state vectors. Second, the results related to the rate of convergence may depend on functional forms and parameter values but there is no reason to believe that our example is atypical. Third, we use our example to illustrate a strategy for diagnosing convergence.

5.1 Time per iteration

As we showed in Section 2.3, under widely used laws of motion, continuous time avoids the curse of dimensionality in the expectation over successor states. Since the algorithms for both discrete and continuous time perform this computation once for each state and each firm in each iteration, we divide the time it takes to complete one iteration by the number of states and the number of firms. Tables 1 and 2 summarize the results for the three algorithms presented in Section 3—the discrete-time algorithm, the continuous-time algorithm without precomputed addresses, and the continuous-time algorithm with precomputed addresses.¹⁶ Table 1 assumes $M = 18$ quality levels per firm and up to $N = 8$ firms just as PM1 did; Table 2 reduces M to 9 so as to accommodate a larger number of firms. Both tables also report the number of states after symmetry and anonymity are invoked, $\frac{(N+M-1)!}{N!(M-1)!}$, and the number of unknowns, which equals one value and one policy per state and firm, along with the ratio of discrete to continuous time without precomputed addresses, the ratio of continuous time without to with precomputed addresses, and the ratio of discrete time to continuous time with precomputed addresses.

Avoiding the curse of dimensionality in the expectation over successor states yields a significant advantage only if this particular computation takes up a large fraction of the running time. Tables 1 and 2 show that this is the case: the discrete-time algorithm spends more than 50% of its time on it if $N = 2$, about 90% if $N = 4$, and essentially 100% if $N \geq 6$. Hence, computing the expectation over successor states is indeed the bottleneck of the discrete-time algorithm. The continuous-time algorithms, in contrast, spend between 33% and 72% of their time on it.

Even in its basic form, the continuous-time algorithm is far faster than the discrete-time algorithm. The gain from continuous time increases from 50% if $N = 2$ to a factor of 200 if $N = 8$ in the case of $M = 18$ (Table 1) and from 42% if $N = 2$ to a factor of 70,947 if $N = 14$ in the case of $M = 9$ (Table 2). In line with theory, the computational burden grows exponentially in N in discrete time, but approximately linearly in continuous time. Consequently, the gain from continuous time explodes in the dimension of the state vector.

Precomputed addresses yield further gains: the continuous-time algorithm without precomputed addresses takes about 20–50% more time per iteration than the continuous-time algorithm with precomputed addresses. Compounding the gains from continuous time and precomputed addresses yields a total gain over discrete time that

¹⁶The programs are written in ANSI C and compiled with Microsoft Visual C++ .NET 2003 (code optimization enabled). All computations are carried out on an IBM ThinkPad T40 with a 1.6 GHz Intel Pentium M processor and 1.5 GB memory running Microsoft Windows XP Professional.

TABLE 1. Time per iteration per state per firm and percentage of time spent on computing the expectation.^a

Number of Firms	Number of States	Number of Unknowns	Discrete Time		Continuous Time Without Precomputed Addresses		Continuous Time With Precomputed Addresses		Ratio		
									Discrete to Continuous Time Without Precomputed Addresses	Continuous Time Without to With Precomputed Addresses	Discrete to Continuous Time With Precomputed Addresses
			sec	%	sec	%	sec	%			
2	171	684	1.07 (−6)	55	7.13 (−7)	41	5.85 (−7)	36	1.50	1.22	1.83
3	1140	6840	1.61 (−6)	76	6.67 (−7)	44	5.26 (−7)	38	2.41	1.27	3.06
4	5985	47,880	3.30 (−6)	87	6.68 (−7)	49	5.10 (−7)	41	4.94	1.31	6.48
5	26,334	263,340	8.05 (−6)	98	7.06 (−7)	49	5.24 (−7)	43	11.40	1.35	15.36
6	100,947	1,211,364	2.15 (−5)	97	7.51 (−7)	52	5.37 (−7)	46	28.57	1.40	40.00
7	346,104	4,845,456	6.19 (−5)	100	7.74 (−7)	56	5.47 (−7)	49	80.00	1.42	113.21
8	1,081,575	17,305,200	1.65 (−4)	100	8.23 (−7)	58	5.92 (−7)	56	200.28	1.39	278.44

^aQuality ladder model with $M = 18$ quality levels per firm and a discount factor of 0.925. (k) is shorthand for $\times 10^k$.

TABLE 2. Time per iteration per state per firm and percentage of time spent on computing the expectation.^a

Number of Firms	Number of States	Number of Unknowns	Discrete Time		Continuous Time Without Precomputed Addresses		Continuous Time With Precomputed Addresses		Ratio		
			sec	%	sec	%	sec	%	Disc. to Cont. Time Without Precomp. Addresses	Cont. Time Without to With Precomp. Addresses	Disc. to Cont. Time With Precomp. Addresses
2	45	180	9.78 (−7)	52	6.89 (−7)	42	5.67 (−7)	33	1.42	1.22	1.73
3	165	990	1.45 (−6)	74	6.36 (−7)	44	5.05 (−7)	38	2.29	1.26	2.88
4	495	3960	2.90 (−6)	88	6.36 (−7)	48	4.75 (−7)	43	4.55	1.34	6.10
5	1287	12,870	6.94 (−6)	96	6.42 (−7)	53	4.77 (−7)	46	10.81	1.35	14.57
6	3003	36,036	1.81 (−5)	98	6.88 (−7)	55	4.88 (−7)	45	26.34	1.41	37.12
7	6435	90,090	5.02 (−5)	100	7.33 (−7)	53	5.11 (−7)	48	68.48	1.43	98.26
8	12,870	205,920	1.31 (−4)	100	7.77 (−7)	55	5.24 (−7)	50	168.33	1.48	249.38
9	24,310	437,580	3.82 (−4)	100	7.77 (−7)	62	5.39 (−7)	53	492.16	1.44	709.04
10	43,758	875,160	1.07 (−3)	100	8.34 (−7)	64	5.94 (−7)	44	1282.19	1.40	1800.00
11	75,582	1,662,804	2.99 (−3)	100	8.42 (−7)	67	5.77 (−7)	56	3557.14	1.46	5187.50
12	125,970	3,023,280	8.20 (−3)	100	8.60 (−7)	68	5.95 (−7)	60	9533.08	1.44	13,770.00
13	203,490	5,290,740	2.42 (−2)	100	9.22 (−7)	69	6.20 (−7)	61	26,235.65	1.49	39,033.56
14	319,770	8,953,560	6.76 (−2)	100	9.53 (−7)	72	6.55 (−7)	59	70,946.70	1.45	103,195.27

^aQuality ladder model with $M = 9$ quality levels per firm and a discount factor of 0.925. (k) is shorthand for $\times 10^k$.

ranges from 83% if $N = 2$ to a factor of 278 if $N = 8$ in the case of $M = 18$ (Table 1) and from 73% if $N = 2$ to a factor of 103,195 if $N = 14$ in the case of $M = 9$ (Table 2).

In sum, an iteration of the continuous-time algorithms is orders of magnitude faster than its discrete-time counterpart for games with more than a few state variables. Most of the gain is from avoiding the curse of dimensionality, but the precomputed addresses, a computational strategy that is effectively constrained to continuous time, also make a significant contribution.

5.2 Number of iterations

While an iteration is far faster in the continuous- than in the discrete-time algorithm, this does not prove that the equilibrium of a continuous-time model is faster to compute, since the model is not solved until the iterations of the algorithm have converged. There are reasons to think that the continuous-time algorithm needs more iterations to converge. Suppose that the strategic elements in the stochastic game were eliminated, so that it reduces to a disjoint set of single-agent dynamic programming problems. In discrete time, a value function iteration approach (also called a pre-Gauss–Jacobi method) would now converge at rate β . As we pointed out in Section 3.2, the continuous-time contraction factor

$$\eta(X(\omega), \omega) = \frac{\phi(X(\omega), \omega)}{\rho + \phi(X(\omega), \omega)}$$

is not constant, but varies with players' policies from state to state. It has a simple interpretation: $\eta(X(\omega), \omega)$ is the expected net present value of a dollar delivered at the next time the state changes if the current state is ω and players' policies are $X(\omega)$. This is easily seen in the special case of $\rho \ll \phi(X(\omega), \omega) = 1$, since

$$\eta(X(\omega), \omega) = \frac{1}{\rho + 1} \approx 1 - \rho = 1 + \ln \beta \approx \beta.$$

In general, if the discount rate ρ is large or if the hazard rate $\phi(X(\omega), \omega)$ is small, then $\eta(X(\omega), \omega)$ is small and there is a strong contraction aspect to a value function iteration approach. However, $\eta(X(\omega), \omega)$ could be close to 1 if the discount rate is small or if the hazard rate is large, in which case a value function iteration approach would converge slowly. Since $\phi(X(\omega), \omega) = \sum_{i=1}^N \phi^i(X(\omega), \omega)$ in the special case of independent transitions, this in particular suggests that convergence could be slow if the number of players N is large.

To further explore this issue, we require a measure of the distance between two sets of value and policy functions. We want our distance measure to be unit-free and to describe the relative difference. Therefore, we define the L_∞ -relative difference between $\hat{V} = (\hat{V}^1, \dots, \hat{V}^N)$ and $\check{V} = (\check{V}^1, \dots, \check{V}^N)$ to be

$$E(\hat{V}, \check{V}) = \left\| \frac{\hat{V} - \check{V}}{1 + |\hat{V}|} \right\| = \max_{i=1, \dots, N} \max_{\omega \in \Omega} \left| \frac{\hat{V}^i(\omega) - \check{V}^i(\omega)}{1 + |\hat{V}^i(\omega)|} \right|.$$

We similarly define $E(\hat{X}, \check{X})$.

TABLE 3. Number of iterations to convergence.^a

Number of Firms	Discount Factor	Discrete Time		Continuous Time		Ratio	
		<10 ⁻⁴	<10 ⁻⁸	<10 ⁻⁴	<10 ⁻⁸	<10 ⁻⁴	<10 ⁻⁸
3	0.925	118	201	212	446	0.56	0.45
3	0.98	412	702	776	1699	0.53	0.41
3	0.99	782	1367	1531	3393	0.51	0.40
3	0.995	1543	2719	3042	6779	0.51	0.40
6	0.925	118	201	364	725	0.32	0.28
6	0.98	494	780	1674	3324	0.30	0.23
6	0.99	983	1525	3379	6761	0.29	0.23
6	0.995	1900	2945	6797	13,637	0.28	0.22
9	0.925	119	201	404	818	0.29	0.25
9	0.98	492	775	2363	4493	0.21	0.17
9	0.99	988	1526	4973	9469	0.20	0.16
9	0.995	2003	3042	10,148	19,365	0.20	0.16
12	0.925			412	854		
12	0.98			2721	5106		
12	0.99			6023	11,181		
12	0.995			12,580	23,304		

^aThe stopping rule is either “distance to truth <10⁻⁴” or “distance to truth <10⁻⁸.” Quality ladder model with $M = 9$ quality levels per firm.

Table 3 compares the discrete- and continuous-time algorithms.^{17,18} It presents the number of iterations until the distance between the current iterate $\hat{V}$ and $\hat{X}$, and the “true” solution V_∞ and X_∞ as measured by $E(\hat{V}, V_\infty)$ and $E(\hat{X}, X_\infty)$ is below a prespecified tolerance of either 10⁻⁴ or 10⁻⁸. To obtain V_∞ and X_∞ , we ran the algorithm until the distance between subsequent iterates as measured by $E(\hat{V}, V)$ and $E(\hat{X}, X)$ failed to decrease any further. The iterations continued until this distance was less than 10⁻¹³ and, in some cases, less than 10⁻¹⁵. The final iterates were considered the true solution since they satisfied the equilibrium conditions essentially up to machine precision.

In light of our previous discussion, we expect the number of iterations to be sensitive to the number of firms and the discount factor. Hence, Table 3 assumes $N \in \{3, 6, 9, 12\}$ and $\beta = e^{-\rho} \in \{0.925, 0.98, 0.99, 0.995\}$. We omit the cases with $N = 12$ in discrete time because one iteration takes more than 3 hours, thus making it impractical to compute the true solution. We see that the continuous-time algorithm needs more iterations to converge than its discrete-time counterpart, and that this gap widens very slightly as we increase β (decrease ρ). On the other hand, the number of iterations needed by the discrete-time algorithm remains more or less constant as we increase the number of firms, whereas the number of iterations needed by the continuous-time algorithm increases rapidly as we go from $N = 3$ to $N = 6$. Fortunately, the number of iterations in-

¹⁷Whether we use precomputed addresses in continuous time is immaterial for the number of iterations to convergence.

¹⁸The starting values are $V^i(\omega) = \frac{\pi^i(\omega)}{1-\beta}$ and $X^i(\omega) = 0$ in discrete time, and $V^i(\omega) = \frac{\pi^i(\omega)}{\rho}$ and $X^i(\omega) = 0$ in continuous time.

creases slowly as we go from $N = 6$ to $N = 9$ and remains more or less constant thereafter, so that the gap between the algorithms stabilizes.

We last note that both the discrete- and the continuous-time algorithms always converged in the case of the quality ladder model as specified in Section 4. Our experience with other models is that sometimes either one or both algorithms fail to converge and that the number of convergence failures is about the same for the two algorithms.¹⁹

5.3 Time to convergence

The continuous-time algorithm suffers an iteration penalty because $\eta(X(\omega), \omega)$ substantially exceeds the discrete-time discount factor β . Even though the continuous-time algorithm needs more iterations, the loss in the number of iterations is small when compared to the gain from avoiding the curse of dimensionality in computing the expectation over successor states. Table 4 illustrates this comparison and the total gain from continuous time. Continuous time beats discrete time by 60% if $N = 3$, a factor of 12 if $N = 6$, a factor of 209 if $N = 9$, a factor of 3977 if $N = 12$, and a factor of 29,734 if $N = 14$. To put these numbers in perspective, in the case of the 14-firm quality ladder model, it

TABLE 4. Time to convergence.^a

Number of Firms	Discrete Time (min)	Continuous Time (min)	Ratio		
			Time per Iteration	Number of Iterations	Time to Convergence
2	1.80 (−4)	1.12 (−4)	1.73	0.93	1.61
3	1.42 (−3)	8.83 (−4)	2.88	0.56	1.60
4	1.13 (−2)	4.43 (−3)	6.10	0.42	2.54
5	8.78 (−2)	1.70 (−2)	14.57	0.36	5.18
6	6.42 (−1)	5.34 (−2)	37.12	0.32	12.03
7	4.44 (0)	1.47 (−1)	98.26	0.31	30.19
8	2.67 (1)	3.56 (−1)	249.38	0.30	74.94
9	1.66 (2)	7.95 (−1)	709.04	0.29	208.85
10	9.28 (2)	1.77 (0)	1800.00	0.29	523.72
11	4.94 (3)	3.30 (0)	5187.50	0.29	1498.33
12	2.46 (4)	6.18 (0)	13,770.00	0.29	3977.26
13	1.27 (5)	1.13 (1)	39,033.56	0.29	11,246.96
14	6.00 (5)	2.02 (1)	103,195.27	0.29	29,734.23

^aThe stopping rule is “distance to truth $< 10^{-4}$.” Entries in italics are based on an estimated 119 iterations to convergence in discrete time. Quality ladder model with $M = 9$ quality levels per firm and a discount factor of 0.925. (k) is shorthand for $\times 10^k$.

¹⁹There are a number of things one can try to facilitate convergence. First, dampening may help to “smooth out” the path that the algorithm takes; see footnote 18 of PM1 and Chapter 3 of Judd (1998) for details. Second, the Stein–Rosenberg theorem asserts, at least for certain systems of linear equations, that if the Gauss–Jacobi algorithm fails to converge, then so does the Gauss–Seidel algorithm; (see, e.g., Proposition 6.9 in Section 2.6 of Bertsekas and Tsitsiklis (1997)). This suggests that a Gauss–Jacobi scheme such as PM1 may be less prone to convergence failures than our Gauss–Seidel scheme. Third, one may solve out for the Nash equilibrium in each state rather than rely on the iterative best reply approach of our algorithm (see Chen, Doraszelski, and Harrington (2009)).

takes about 20 minutes to compute the equilibrium of the continuous-time game, but it would take over 1 year to do the same in discrete time!

The stochastic approximation algorithm suggested by PM2 is another effort to alleviate the burden of computing equilibria of dynamic stochastic games. To break the curse of dimensionality in computing expectations in discrete-time games, they created approximations to players' expectations over all possible future states and updated them each time a state was visited by a random draw from the set of successor states. Similar to Monte Carlo integration, many visits to a state are required to reduce the approximation error to an acceptable level and obtain useful estimates of these expectations. In addition, PM2 addressed a separate issue in computing equilibria, namely the large size of the state space, by tracking the states that appear to be visited frequently in equilibrium, that is, are in the ergodic set, and ignoring the rest. Since the number of states is independent of the concept of time, we do not pursue this idea and instead simply note that our continuous-time algorithm can be extended to target the ergodic set and that this may result in further speedups in some applications.²⁰

Since PM2 exploited this additional idea in addition to stochastic approximation, whereas we focus on the problem of computing the expectation over successor states, it is difficult to compare their algorithm with our continuous-time approach. However, to give the reader some basis for comparison, we note that PM2 reported that their algorithm cuts running time roughly in half (relative to PM1) in a 6-firm quality ladder model where the ergodic set comprises about 3.3% of all states. They also project that it reduces running time by a factor of 250 in a 10-firm quality ladder model where the ergodic set contains 0.4% of all states. In contrast, our continuous-time approach avoids approximations altogether, computes the equilibrium on the entire state space, and yet reduces running time by a factor of 12 and 524, respectively.

5.4 Stopping rules

It is rarely feasible to compute the true solution V_∞ and X_∞ . Rather, the algorithm is terminated once the distance between subsequent iterates, $E(\hat{V}, V)$ and $E(\hat{X}, X)$, is deemed sufficiently small. The problem is that the distance to the true solution, $E(\hat{V}, V_\infty)$ and $E(\hat{X}, X_\infty)$, may be much larger than the distance between subsequent iterates. This makes it hard to tell whether the algorithm has provided us with a reasonable approximation to an equilibrium. Below we describe a stopping rule that uses a careful examination of the iteration history to assess the accuracy of the computations.

²⁰Many applications require the equilibrium to be known at states outside the ergodic set. For example, in studying the effect of a change in antitrust policy, the initial state generated by the old regime may not be in the ergodic set induced by the equilibrium under the new regime, so that the transition from the old to the new regime cannot be captured accurately unless the equilibrium is computed at the transient states. In practice, this can be done via multiple restarts of the PM2 algorithm, but at a cost. Moreover, the ergodic set is large in many dynamic stochastic games and there is thus little gain from restricting attention to it. In Doraszelski and Markovich (2007), for example, the ergodic set consists of the entire state space. Finally, as PM2 acknowledged, their algorithm needs to be significantly altered to solve models in which behavior depends on players' values and policies "off the equilibrium path," as is typically the case in models of collusion, since off-path states are by definition never visited in equilibrium (PM2, p. 1278).

As we pointed out in Section 3.2, neither the discrete- nor the continuous-time algorithm is guaranteed to converge. However, if the algorithm does converge, then convergence is linear as in all Gaussian schemes (Ortega and Rheinboldt (1970, p. 301)). Consider therefore a sequence $\{z_l\}_{l=0}^{\infty}$ that converges linearly to the limit z_{∞} , so that

$$\lim_{l \rightarrow \infty} \frac{\|z_{l+1} - z_{\infty}\|}{\|z_l - z_{\infty}\|} \leq \theta < 1.$$

Suppose the first inequality can be strengthened to hold along the entire sequence of iterates, that is, $\|z_{l+1} - z_{\infty}\| \leq \theta \|z_l - z_{\infty}\|$ for all l ; this contraction property is similar to dynamic programming except that we do not a priori know the convergence factor θ . Then the distance to the limit is related to the distance between subsequent iterates by $\|z_l - z_{\infty}\| \leq \frac{\|z_{l+1} - z_l\|}{1 - \theta}$. Hence, to ensure that the current iterate is within a prespecified tolerance ε of the limit, we can stop once

$$\|z_{l+1} - z_l\| \leq \varepsilon(1 - \theta). \quad (14)$$

The key is to estimate the convergence factor from past iterates. We let k be the first iteration such that $\|z_k - z_{k-1}\| < 10\varepsilon$ and let l be the first iteration such that $\|z_l - z_{l-1}\| < \varepsilon$ to produce the estimate

$$\hat{\theta} = \left(\frac{\|z_l - z_{l-1}\|}{\|z_k - z_{k-1}\|} \right)^{1/(l-k)}. \quad (15)$$

Equations (14) and (15) together comprise the *adaptive* stopping rule. It contrasts with the widely used ad hoc rule of stopping once

$$\|z_{l+1} - z_l\| \leq \varepsilon.$$

Table 5 compares the two stopping rules with $\varepsilon = 10^{-4}$. For the sake of brevity, we focus on the continuous-time quality ladder model; the results for discrete time are similar and can be found in the Online Appendix. In all cases, the adaptive rule outperforms the ad hoc rule. The ad hoc rule prematurely terminates the algorithm, although the distance to the true solution exceeds the prespecified tolerance by up to 3 orders of magnitude. In contrast, the adaptive rule usually terminates the algorithm once the distance to the true solution is smaller than the prespecified tolerance, and it is always within an order of magnitude. Because we do not know the exact value of the convergence factor, there are cases in which our adaptive rule stops early (e.g., $N = 3$ and $\beta = 0.995$). We note, however, that our estimator in equation (15) is quite crude and can be improved at little cost (Judd (1998, pp. 42–44)). Overall, Table 5 clearly shows the importance of having a reliable stopping rule.

6. CONCEPTUAL DIFFERENCES BETWEEN CONTINUOUS AND DISCRETE TIME

In Section 5, we emphasized the computational advantages of continuous time. However, a continuous- and a discrete-time formulation of the same economic problem may differ and sometimes one or the other approach is preferable. Below we first note some

TABLE 5. Stopping rules.^a

Number of Firms	Discount Factor	Ad Hoc Rule		Adaptive Rule		
		Terminal Iteration	Distance to Truth	Terminal Iteration	Distance to Truth	Convergence Factor
3	0.925	131	2.43 (−3)	218	7.84 (−5)	0.9676
3	0.98	313	9.93 (−3)	775	1.00 (−4)	0.9899
3	0.99	455	2.01 (−2)	1483	1.27 (−4)	0.9937
3	0.995	589	4.05 (−2)	2778	1.92 (−4)	0.9953
6	0.925	220	3.84 (−3)	370	8.42 (−5)	0.9777
6	0.98	742	1.78 (−2)	1689	9.17 (−5)	0.9948
6	0.99	1198	3.66 (−2)	3454	8.15 (−5)	0.9978
6	0.995	1832	7.41 (−2)	6766	1.04 (−4)	0.9986
9	0.925	232	4.45 (−3)	407	9.18 (−5)	0.9791
9	0.98	1100	2.30 (−2)	2387	9.01 (−5)	0.9961
9	0.99	1927	4.87 (−2)	5091	7.84 (−5)	0.9984
9	0.995	3129	1.00 (−1)	10,358	8.10 (−5)	0.9992
12	0.925	227	4.73 (−3)	411	1.02 (−4)	0.9781
12	0.98	1276	2.58 (−2)	2751	8.89 (−5)	0.9966
12	0.99	2447	5.59 (−2)	6185	7.48 (−5)	0.9987
12	0.995	4217	1.16 (−1)	12,994	7.01 (−5)	0.9994

^aTerminal iteration, distance to truth at terminal iteration, and estimated convergence factor are shown. Prespecified tolerance is 10^{-4} . Continuous-time quality ladder model with $M = 9$ quality levels per firm. (k) is shorthand for $\times 10^k$.

of the conceptual differences between discrete- and continuous-time models. Next we compare the equilibrium behavior of players and the dynamics implied by that behavior using the quality ladder model of Section 4 as an example. We finally note some limitations of continuous-time models.

6.1 State changes

Discrete- and continuous-time models differ in *how often* and *how much* a state variable can change over a finite interval of time. In discrete-time models, a state variable can change at most once per period, so the number of changes is bounded.²¹ In continuous-time models, by contrast, the number of changes over a finite interval of time is not bounded. This may or may not be appropriate, depending on the institutional and technological details of the economic problem under study. Rigidities in the decision-making process could put a limit on change, for example, if decisions are made by a board of directors that meets at fixed times or if there are contractual obligations that lock a firm into its decision for a period of time.

In addition to restricting how often a state variable can change over a finite interval of time, discrete-time models also force the changes to take place at regular intervals, a sometimes useful feature. Consider the automobile industry. Automobile manufacturers

²¹To allow for a larger number of changes over a finite interval of time in discrete-time models, one could think about shortening the length of a period by taking the discount factor close to 1. However, as Table 3 shows, the number of iterations to convergence increases with β .

launch new models at more or less the same time in early fall of each year, a fact that can be easily captured in a discrete-time model, but, without explicitly accounting for the reasons behind it, not in a continuous-time model. On the other hand, this feature of discrete-time models makes it harder to interpret data that do not arrive at regular intervals. For example, plant openings and closings do not all take place on the same day, but instead are spread out over the year.

Turning from the number of changes to their size, many dynamic stochastic games such as the quality ladder model of Section 4 restrict players' transitions to immediately adjacent states. This assumption imposes a sense of continuity—a player cannot go from state 3 to state 5 without passing through state 4—although the number of states is finite. The “continuity” assumption has different consequences for discrete- and continuous-time models. In discrete-time models it implies that a state variable changes by at most 1 unit in any given period. Hence, a minimum of n periods is required for a change of n units. In continuous-time models, by contrast, the “continuity” assumption just says that the state variable changes by 1 unit at a time, but these changes “add up.” Continuous time thus allows for a richer range of outcomes over a finite interval of time.

6.2 Strategic interactions

The nature of strategic interactions may also be different in discrete- and continuous-time models and has to be considered in choosing an appropriate formulation of an economic problem. Suppose that two firms are both trying to expand their capacity. In a discrete-time model there is some chance that both firms succeed in a given period. This may result in excess capacity, make both firms regret their investments, and perhaps even spur some efforts to disinvest. “Mistakes” like this cannot happen in a continuous-time model, since at most one firm succeeds at a given point in time and the other promptly adjusts by stopping its investment.

In what follows, we assume that players make decisions in a discrete-time model at the beginning of the period, whereas state-to-state transitions take place at the end, say because it takes a period for an investment project to come to fruition.²² If so, then a player may also react right away to a change in a rival's state at the end of period $t - 1$ by changing his/her action at the beginning of period t , but the player must wait at least a period before this brings about a change in his/her own state. In many applications of EP's framework such as the quality ladder model of Section 4, the state space is fairly coarse. Thus, a change in the state has a large effect on the strategic situation, and while a lag of a few days, weeks, or even months may be plausible, a lag of 1 or more years is often not. In a continuous-time model, by contrast, a player may adjust his/her action to bring about a change in his/her own state much more quickly. This difference in the ability of players to respond may substantially affect the equilibrium, since a player's actions are contingent on his/her rivals' reactions. Whether the rapid response of a continuous-time model or the delayed response of a discrete-time model is deemed a better approximation depends on the economic problem at hand.

²²While other assumptions may be made, ours appears to be the most widely used one in the literature following EP (see, e.g., [Fershtman and Pakes \(2000, p. 210\)](#)).

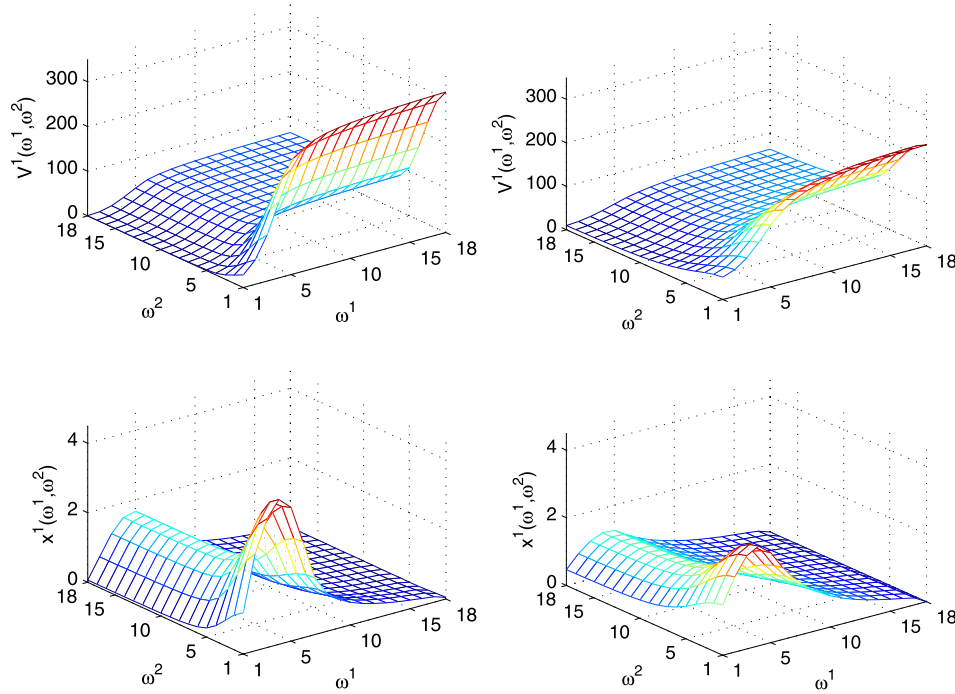


FIGURE 1. Equilibrium value (upper panels) and policy functions (lower panels) for the discrete- (left panels) and the continuous-time (right panels) quality ladder model with $N = 2$ firms, $M = 18$ quality levels per firm, and a discount factor of 0.925.

6.3 Equilibrium and dynamics

Given the conceptual differences between discrete- and continuous-time models noted above, we now ask how the equilibrium behavior of firms and the dynamics implied by that behavior change as the discrete-time quality ladder model is recast in continuous time.

Figure 1 compares the equilibrium value and policy functions for $N = 2$ firms and $M = 18$ quality levels per firm. Overall, the shapes of these functions are similar. Perhaps the most obvious difference is that a low-quality firm has a lower value and invests less in the discrete-time model (left panels) than in the continuous-time model (right panels). For example, we have $V^1(1, 9) = 0.43$ and $x^1(1, 9) = 0.09$ in discrete time, and $V^1(1, 9) = 16.44$ and $x^1(1, 9) = 0.58$ in continuous time. The reason is that in the discrete-time model, a minimum of n periods is required for a change of n units in a state variable. Hence, the firm is stuck in states with low quality and thus low profit from product market competition for a long time. In contrast, the continuous-time model does not limit how often and how much a state variable can change over a finite interval of time, so that, by investing more heavily, the firm is able to more quickly reach states with high quality.

The second difference is that the peak of investment around state $(4, 1)$ is lower in the continuous-time model. In fact, the policy functions differ most in state $(4, 1)$ and

the value functions in state $(6, 1)$ with $x^1(4, 1) = 4.16$ and $V^1(6, 1) = 272.91$ in discrete time, and $x^1(4, 1) = 2.94$ and $V^1(6, 1) = 168.83$ in continuous time. As firm 1 enjoys an advantage over firm 2 in state $(4, 1)$, it has an incentive to further invest in quality improvements so as to cement its leadership position. But in the continuous-time model, the follower is able to more quickly catch up to the leader. This renders the leadership position more tentative and consequently less valuable, and, in turn, diminishes the leader's incentive to invest.

From the equilibrium policy function, we construct the probability distribution over the next period's state $((\omega')^1, (\omega')^2)$ given this period's state (ω^1, ω^2) , that is, the transition probability matrix that characterizes the discrete-time Markov process of industry dynamics. Similarly, we construct the infinitesimal generator (rate matrix) of the continuous-time Markov process of industry dynamics. Rather than rely on simulation, we apply stochastic process theory to analyze these Markov processes. More specifically, we compute the transient distribution over states at time t , μ^t , starting from state $(1, 1)$. This tells us how likely each possible industry structure is at time t , given that both firms began the game at the minimal quality level. In addition, we compute the limiting (or ergodic) distribution over states, μ^∞ .²³ The transient distribution captures short-run dynamics and the limiting distribution captures long-run (or steady-state) dynamics.

Figure 2 compares the transient distribution at time $t = 5, 10, 25$ and the limiting distribution. The probability mass is apparently more concentrated in the discrete-time model (left panels) than in the continuous-time model (right panels), in line with the fact that continuous time allows for a richer range of outcomes over a finite interval of time. In Table 6, we list the most likely industry structure (modal state) and its probability at various points in time. In the discrete-time model, in the short run the industry evolves either in a symmetric or in an asymmetric fashion. However, even if a firm is able to gain the upper hand over its rival in the short run, in the long run the most likely industry structure is symmetric and the limiting distribution leaves little probability mass on asymmetric industry structures (see again Figure 2). In the continuous-time model, the dynamics of the industry exhibit greater variability, thus making it less likely that a firm can sustain a pronounced advantage over its rival for an extended period of time. We finally report in Table 6 a firm's expected profit from product market competition and its expected investment in quality improvements along with their standard deviations. These statistics are mostly similar, except that early on expected profits are higher in the continuous-time model due to the fact that a firm is able to more quickly reach states with high quality.

²³Let P be the $M^2 \times M^2$ transition probability matrix in the discrete-time model. The $1 \times M^2$ transient distribution in period t is given by $\mu^t = \mu^0 P^t$, where μ^0 is the $1 \times M^2$ initial distribution and P^t is the t^{th} matrix power of P . The Markov process turns out to be irreducible. That is, all its states belong to a single closed communicating class and the $1 \times M^2$ limiting distribution μ^∞ solves the system of linear equations $\mu^\infty = \mu^\infty P$. In the continuous-time model, let Q be the infinitesimal generator. The transient distribution at time t is given by $\mu^t = \mu^0 \exp(Qt)$ and the limiting distribution μ^∞ solves the system of linear equations $0 = \mu^\infty Q$.

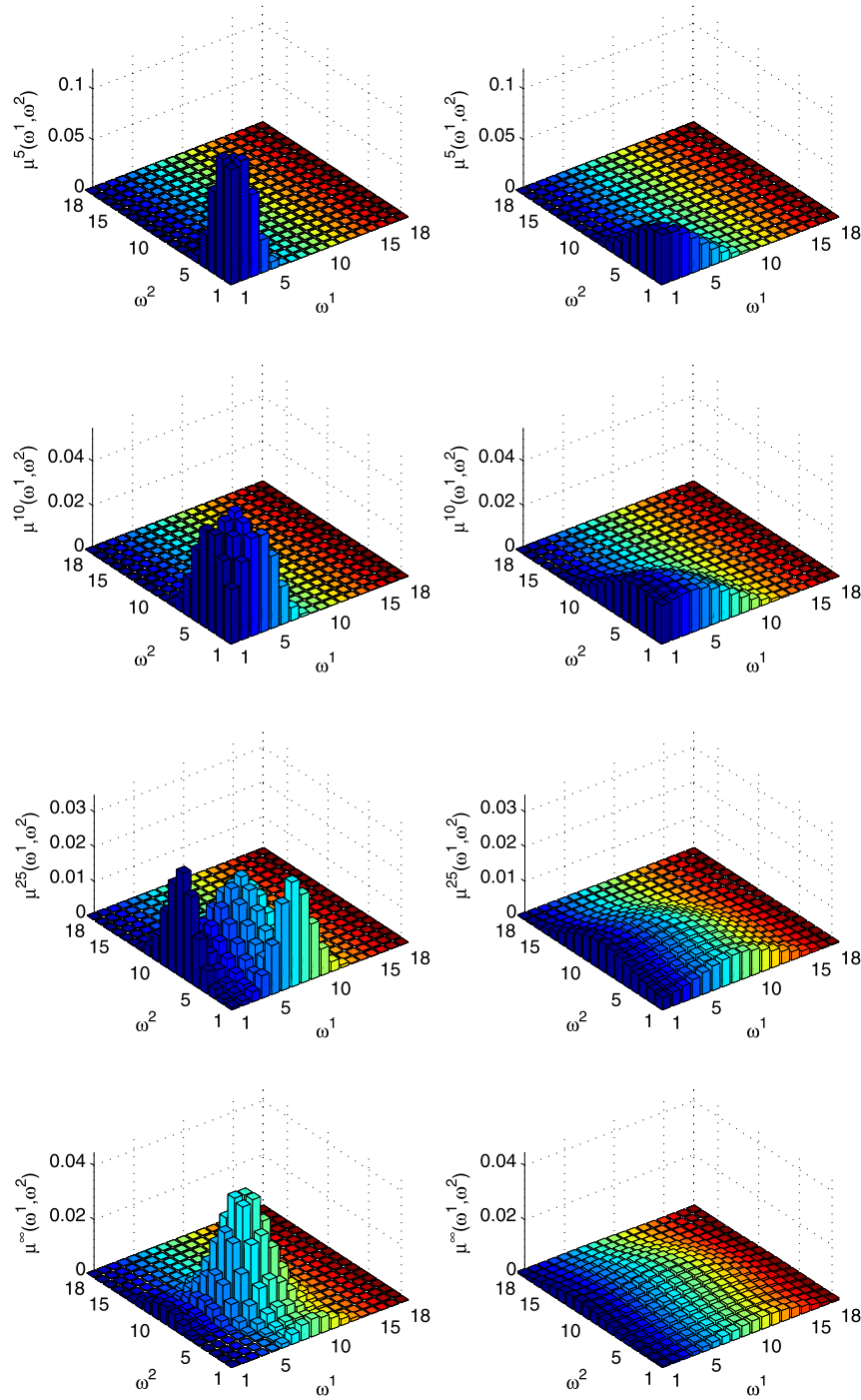


FIGURE 2. Transient distribution at time $t = 5, 10, 25$ (upper panels) and limiting distribution (lower panels) for the discrete- (left panels) and continuous-time (right panels) quality ladder model with $N = 2$ firms, $M = 18$ quality levels per firm, and a discount factor of 0.925.

TABLE 6. Most likely industry structure and its probability, expected profit, and investment, and their standard deviations.^a

t	Most Likely Industry Structure	Prob.	Profit		Investment	
			Mean	Std. Dev.	Mean	Std. Dev.
Discrete time						
5	(2, 1), (1, 2)	0.1177	1.20	3.02	2.20	1.12
10	(3, 3)	0.0510	4.57	7.12	2.32	1.19
25	(7, 1), (1, 7)	0.0301	8.60	9.32	1.46	0.93
50	(7, 7)	0.0307	7.78	8.18	0.97	0.61
100	(8, 7), (7, 8)	0.0360	6.74	6.56	0.90	0.53
∞	(8, 7), (7, 8)	0.0404	6.14	5.27	0.91	0.51
Continuous time						
5	(1, 1)	0.0489	4.98	7.91	1.94	0.61
10	(3, 3)	0.0193	7.24	9.07	1.66	0.64
25	(7, 6), (6, 7)	0.0088	7.89	8.63	1.17	0.59
50	(9, 8), (8, 9)	0.0077	7.44	7.78	0.92	0.51
100	(9, 9)	0.0080	7.16	7.31	0.84	0.48
∞	(9, 9)	0.0080	7.13	7.26	0.84	0.48

^aDiscrete- (upper panel) and continuous-time (lower panel) quality ladder model with $N = 2$ firms, $M = 18$ quality levels per firm, and a discount factor of 0.925.

6.4 Limitations: Calendar time and deterministic transitions

Suppose that there are predictable seasonal fluctuations in demand or cost. In the automobile industry, for example, the model year begins in the early fall with strong demand, after which demand gradually weakens. Because the time within the year determines demand, it is a state variable. Another example is the collusion model of [Fershtman and Pakes \(2011\)](#). In their model, a firm's cost is privately known. The cartel meets when one of its members calls for a meeting, whereupon each firm discloses its cost and output is allocated. Between meetings, each firm invests to reduce its cost, but this is not observed by other firms. The elapsed time since the last meeting is a state variable because a firm's uncertainty about its rivals' costs rises with it. Calendar time is easily handled in a discrete-time model because discrete time adds just another discrete state variable. In contrast, continuous time adds a continuous state variable. Since our continuous-time approach is based on a finite number of states, it cannot directly model calendar time.²⁴

The continuous-time approach is also limited to stochastic state-to-state transitions. Suppose a firm is guaranteed an investment success at a cost of $\bar{x}$. Given the "lumpy" nature of investment, the firm spends either zero and stays put or spends $\bar{x}$ and moves up one state. Such a deterministic transition corresponds to a transition probability of 1 and can thus in principle be modeled in discrete time, whereas in continuous time it requires an infinite hazard rate.²⁵

²⁴Apart from [Fershtman and Pakes \(2011\)](#), we are not aware of any applications of EP's framework in which calendar time plays a role.

²⁵Whether an investment success can indeed be guaranteed or whether there always remains some uncertainty as in the quality ladder model in Section 4 undoubtedly depends on the type of investment one

From a practical point of view, however, lumpy investment also poses difficulties for discrete time because the existence of an equilibrium cannot generally be ensured without allowing for computationally burdensome mixed strategies. The same issue arises if it is assumed that exit (entry) takes place for sure upon receiving (paying) a certain scrap value (setup cost) because this entails that an incumbent firm (potential entrant) transits deterministically to an “inactive” state (initial state). To avoid the need for mixed entry/exit strategies in the context of EP’s framework, [Doraszelski and Satterthwaite \(2010\)](#) assumed that an incumbent firm (potential entrant) draws a random scrap value (setup cost) in each period. Draws are independent across firms and periods. Its scrap value (setup cost) is private information that is known to the firm but unknown to its rivals. Its rivals thus perceive the firm *as if* it were following a mixed entry/exit strategy. The same idea has been applied to construct computationally tractable models with lumpy investment ([Ryan \(2006\)](#), [Besanko, Doraszelski, Lu, and Satterthwaite \(2010a\)](#), [Besanko et al. \(2010b\)](#)). It carries over to continuous time if it is assumed that an incumbent firm (potential entrant) draws a random scrap value (setup cost) at random times instead of in each period. In the Online Appendix, we show how to model entry and exit in continuous time along these lines.

7. CONCLUDING REMARKS

Under standard assumptions, discrete-time stochastic games with a finite number of states suffer from a curse of dimensionality in computing players’ expectations over all possible future states in that their computational burden increases exponentially in the number of state variables. We develop the alternative of continuous-time stochastic games with a finite number of states and demonstrate that they avoid the curse of dimensionality under comparable assumptions, thereby speeding up the computations by orders of magnitude for games with more than a few state variables. We further speed up the computations with precomputed addresses, a computational strategy that is effectively constrained to continuous time. Extending our continuous-time algorithms to focus on the ergodic set as suggested by PM2 may lead to further gains in some applications. There are also many variations of the block Gauss–Seidel scheme in [Section 3](#), and it is highly likely that there are some superior approaches available.

In addition to the computational burden, whether an economic problem is best modeled in continuous or discrete time depends on the details of the institutional and technological setting. We argue that continuous-time formulations have a number of features that may be useful in modeling dynamic strategic interactions. Overall, the advantages of continuous-time games are often substantial and open the way to study more complex and realistic stochastic games than currently feasible.

has in mind. [Pakes \(1994, p. 177\)](#), for example, contended, “One might argue the relevance of the special deterministic case for investment in physical capital, but it seems much less appropriate for the accumulation of ‘intangible’ capital stocks that emanate from a firm’s investment in research and exploration, or in advertising and goodwill. Here the randomness in the outcome from the investment activities often seem... to have strikingly large variances.”

APPENDIX

Proof of Proposition 1

We show that Brouwer's fixed point theorem applies to a suitably defined mapping from the set of values and policies into itself.

Our first task is to show that we can restrict attention to a nonempty, compact, and convex subset of values. By the extreme value theorem, Assumptions 2 and 3 imply that the model's primitives are bounded. Let

$$\begin{aligned}\underline{\pi} &= \min_{x, \omega, i} \pi^i(x, \omega), & \bar{\pi} &= \max_{x, \omega, i} \pi^i(x, \omega), \\ \underline{\Phi} &= \min_{x, \omega, \omega', i} \Phi^i(x, \omega, \omega'), & \bar{\Phi} &= \max_{x, \omega, \omega', i} \Phi^i(x, \omega, \omega'), \\ \bar{\phi} &= \min_{x, \omega} \phi(x, \omega)\end{aligned}$$

be the lower and upper bounds on the model's primitives. In equilibrium, $V^i(\omega)$ is bounded by the (loose) lower and upper bounds

$$\underline{V} = \frac{1}{\rho}(\underline{\pi} + \bar{\phi} \min\{0, \underline{\Phi}\}), \quad \bar{V} = \frac{1}{\rho}(\bar{\pi} + \bar{\phi} \max\{0, \bar{\Phi}\}).$$

$\underline{V}$ is constructed by assuming that player i receives the lowest possible payoff flow $\underline{\pi}$ forever and, in addition, is subjected to the lowest possible change in his/her stock of wealth $\underline{\Phi}$ (or zero, if the change in wealth is positive) at the highest possible hazard rate $\bar{\phi}$. Similarly, $\bar{V}$ is constructed by assuming that player i receives the highest possible payoff flow $\bar{\pi}$ forever and, in addition, is subjected to the highest possible change in his/her stock of wealth $\bar{\Phi}$ (or zero, if the change in wealth is negative) at the highest possible hazard rate $\bar{\phi}$. Thus, in equilibrium, the value of player i in state ω must be an element of the nonempty, compact, and convex set $[\underline{V}, \bar{V}]$.

Our second task is to define a mapping from the set of values and policies into itself. To simplify the notation, recall that $V^i(\cdot)$ is a $(|\Omega| \times 1)$ vector of values of player i in the various possible states. Define $V(\cdot) = (V^1(\cdot), \dots, V^N(\cdot))$ to be a $(|\Omega| \times N)$ matrix of players' values. Analogously define $X^i(\cdot)$ and $X(\cdot) = (X^1(\cdot), \dots, X^N(\cdot))$. Pointwise define the mapping $Y(\cdot) = \begin{pmatrix} Y^V(\cdot) \\ Y^X(\cdot) \end{pmatrix}$ from the set of values and policies by

$$\begin{aligned}Y^{V,i,\omega}(V(\cdot), X(\cdot)) &= \left\{ \max \left\{ \underline{V}, \min \left\{ \bar{V}, \frac{1}{\rho} \max_{x^i \in \mathbb{X}^i(\omega)} h^i(x^i, X^{-i}(\omega), \omega, V^i(\cdot)) \right\} \right\} \right\}, \\ Y^{X,i,\omega}(V(\cdot), X(\cdot)) &= \left\{ \arg \max_{x^i \in \mathbb{X}^i(\omega)} h^i(x^i, X^{-i}(\omega), \omega, V^i(\cdot)) \right\}\end{aligned}$$

for each player $i = 1, \dots, N$ in each state $\omega \in \Omega$. To apply Brouwer's fixed point theorem, we have to show that $Y(\cdot)$ is a continuous function from a nonempty, compact, and convex set into itself. If so, then there must exist $(V(\cdot), X(\cdot))$ such that $(V(\cdot), X(\cdot)) \in Y((V(\cdot), X(\cdot)))$.

Starting with $Y^V(\cdot)$, fix ω and i . Because of Assumptions 1 and 3, the theorem of the maximum yields that $\max_{x^i \in \mathbb{X}^i(\omega)} h^i(x^i, X^{-i}(\omega), \omega, V^i(\cdot))$ is a continuous function of

$X^{-i}(\omega)$ and $V^i(\cdot)$. By Assumption 2, $Y^{V,i,\omega}(\cdot)$ is thus a continuous function of $V(\cdot)$ and $X(\cdot)$ that maps into $[\underline{V}, \bar{V}]$ by construction.

Turning from $Y^V(\cdot)$ to $Y^X(\cdot)$, fix ω and i . The theorem of the maximum also yields that $Y^{X,i,\omega}(\cdot)$ is a nonempty, compact-valued, and upper hemicontinuous correspondence. Moreover, $Y^{X,i,\omega}(\cdot)$ is single-valued by Assumption 4. Thus $Y^{X,i,\omega}(\cdot)$ is a continuous function of $V(\cdot)$ and $X(\cdot)$ that maps into $\mathbb{X}^i(\omega)$.

REFERENCES

- Arcidiacono, P., P. Bayer, J. Blevins, and P. Ellickson (2010), "Estimation of dynamic discrete choice models in continuous time." Working paper, Duke University, Durham. [57]
- Basar, T. and J. Olsder (1999), *Dynamic Noncooperative Game Theory*, second edition. Society for Industrial and Applied Mathematics, Philadelphia. [54, 58]
- Bellman, R. (1957), *Dynamic Programming*. Princeton University Press, Princeton. [59, 60, 61]
- Benkard, L. (2004), "A dynamic analysis of the market for wide-bodied commercial aircraft." *Review of Economic Studies*, 71 (3), 581–611. [55, 67, 68]
- Berry, S. and A. Pakes (1993), "Some applications and limitations of recent advances in empirical industrial organization: Merger analysis." *American Economic Review*, 83 (2), 247–252. [66]
- Bertsekas, D. and J. Tsitsiklis (1997), *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont. [79]
- Besanko, D. and U. Doraszelski (2004), "Capacity dynamics and endogenous asymmetries in firm size." *Rand Journal of Economics*, 35 (1), 23–49. [64, 67]
- Besanko, D., U. Doraszelski, Y. Kryukov, and M. Satterthwaite (2010), "Learning-by-doing, organizational forgetting, and industry dynamics." *Econometrica*, 78 (2), 453–508. [67]
- Besanko, D., U. Doraszelski, L. Lu, and M. Satterthwaite (2010a), "Lumpy capacity investment and disinvestment dynamics." *Operations Research*, 58 (4), 1178–1193. [88]
- Besanko, D., U. Doraszelski, L. Lu, and M. Satterthwaite (2010b), "On the role of demand and strategic uncertainty in capacity investment and disinvestment dynamics." *International Journal of Industrial Organization*, 28 (4), 383–389. [67, 88]
- Borkovsky, R., U. Doraszelski, and Y. Kryukov (2010), "A user's guide to solving dynamic stochastic games using the homotopy method." *Operations Research*, 58 (4), 1116–1132. [67]
- Breton, M. (1991), "Algorithms for stochastic games." In *Stochastic Games and Related Topics: In Honor of Professor L.S. Shapley* (T. Raghavan, T. Ferguson, T. Parthasarathy, and O. Vrieze, eds.), 45–57, Kluwer Academic Publishers, Dordrecht. [59]

Caplin, A. and B. Nalebuff (1991), "Aggregation and imperfect competition: On the existence of equilibrium." *Econometrica*, 59 (1), 25–59. [71]

Chen, J. (2009), "The effects of mergers with dynamic capacity accumulation." *International Journal of Industrial Organization*, 27, 92–109. [64]

Chen, J., U. Doraszelski, and J. Harrington (2009), "Avoiding market dominance: Product compatibility in markets with network effects." *Rand Journal of Economics*, 40 (3), 455–485. [79]

de Roos, N. (2004), "A model of collusion timing." *International Journal of Industrial Organization*, 22, 351–387. [66]

Dockner, E., S. Jorgensen, N. Van Long, and G. Sorger (2000), *Differential Games in Economics and Management Science*. Cambridge University Press, Cambridge. [54]

Doraszelski, U. and K. Judd (2007), "Dynamic stochastic games with sequential state-to-state transitions." Working paper, Harvard University, Cambridge. [55]

Doraszelski, U. and S. Markovich (2007), "Advertising dynamics and competitive advantage." *Rand Journal of Economics*, 38 (3), 557–592. [64, 80]

Doraszelski, U. and A. Pakes (2007), "A framework for applied dynamic analysis in IO." In *Handbook of Industrial Organization*, Vol. 3 (M. Armstrong and R. Porter, eds.), 1887–1966, North-Holland, Amsterdam. [54]

Doraszelski, U. and M. Satterthwaite (2010), "Computable Markov-perfect industry dynamics." *Rand Journal of Economics*, 41 (2), 215–243. [60, 63, 70, 71, 88]

Elfvig, G. (1937), "Zur Theorie der Markoffschen Ketten." *Acta Societatis Scientiarum Fennicae Nova Series A*, 2 (8), 1–17. [56]

Ericson, R. and A. Pakes (1995), "Markov-perfect industry dynamics: A framework for empirical work." *Review of Economic Studies*, 62, 53–82. [54]

Escobar, J. (2008), "Existence of pure and behavior strategy stationary Markov equilibrium in dynamic stochastic games." Working paper, Stanford University, Stanford. [60]

Federgruen, A. (1978), "On N -person stochastic games with denumerable state space." *Advances in Applied Probability*, 10, 452–471. [59]

Feinberg, E. (2004), "Continuous time discounted jump Markov decision processes: A discrete-event approach." *Mathematics of Operations Research*, 29 (3), 492–524. [61]

Ferris, M., K. Judd, and K. Schmedders (2007), "Solving dynamic games with Newton's method." Working paper, University of Wisconsin, Madison. [67]

Fershtman, C. and A. Pakes (2000), "A dynamic oligopoly with collusion and price wars." *Rand Journal of Economics*, 31, 207–236. [55, 66, 83]

Fershtman, C. and A. Pakes (2011), "Dynamic games with asymmetric information: A framework for empirical work." Working paper, Harvard University, Cambridge. [87]

- Filar, J. and K. Vrieze (1997), *Competitive Markov Decision Processes*. Springer, New York. [58]
- Fink, A. (1964), "Equilibrium in a stochastic n -person game." *Journal of Science of the Hiroshima University Series A-I*, 28, 89–93. [59]
- Gowrisankaran, G. (1999a), "A dynamic model of endogenous horizontal mergers." *Rand Journal of Economics*, 30 (1), 56–83. [55, 66]
- Gowrisankaran, G. (1999b), "Efficient representation of state spaces for some dynamic models." *Journal of Economic Dynamics and Control*, 23, 1077–1098. [70]
- Herings, J. and R. Peeters (2004), "Stationary equilibria in stochastic games: Structure, selection, and computation." *Journal of Economic Theory*, 118, 32–60. [59, 60]
- Howard, R. (1960), *Dynamic Programming and Markov Processes*. MIT Press, Cambridge. [60]
- Isaacs, R. (1954), *Differential Games*. John Wiley & Sons, New York. [54]
- Judd, K. (1998), *Numerical Methods in Economics*. MIT Press, Cambridge. [68, 79, 81]
- Kryukov, Y. (2008), "Dynamic R&D and the effectiveness of policy intervention in the pharmaceutical industry." Working paper, Carnegie Mellon University, Pittsburgh. [57]
- Langohr, P. (2003), "Competitive convergence and divergence: Capability and position dynamics." Working paper, Northwestern University, Evanston. [67]
- Markovich, S. (2008), "Snowball: A dynamic oligopoly model with indirect network effects." *Journal of Economic Dynamics and Control*, 32, 909–938. [66]
- Maskin, E. and J. Tirole (2001), "Markov perfect equilibrium, I: Observable actions." *Journal of Economic Theory*, 100, 191–219. [59]
- Mehra, R. (2003), "The equity premium: Why is it a puzzle?" *Financial Analysis Journal*, 59 (1), 54–69. [56]
- Mertens, J. (2002), "Stochastic games." In *Handbook of Game Theory*, Vol. 3 (R. Aumann and S. Hart, eds.), 1810–1832, Elsevier, Amsterdam. [62]
- Ortega, J. and W. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York. [81]
- Pakes, A. (1994), "Dynamic structural models, problems, and prospects: mixed continuous discrete controls and market interactions." In *Advances in Econometrics: Sixth World Congress*, Vol. 2 (C. Sims, ed.), Cambridge University Press, Cambridge. [88]
- Pakes, A., G. Gowrisankaran, and P. McGuire (1993), "Implementing the Pakes–McGuire algorithm for computing Markov perfect equilibria in Gauss." Working paper, Yale University, New Haven. [70]
- Pakes, A. and P. McGuire (1994), "Computing Markov-perfect Nash equilibria: Numerical implications of a dynamic differentiated product model." *Rand Journal of Economics*, 25 (4), 555–589. [54]

- Pakes, A. and P. McGuire (2001), “Stochastic algorithms, symmetric Markov perfect equilibrium, and the “curse” of dimensionality.” *Econometrica*, 69 (5), 1261–1281. [54]
- Ryan, S. (2006), “The costs of environmental regulation in a concentrated industry.” Working paper, MIT, Cambridge. [88]
- Shapley, L. (1953), “Stochastic games.” *Proceedings of the National Academy of Sciences*, 39, 1095–1100. [54]
- Simon, L. and M. Stinchcombe (1989), “Extensive form games in continuous time: Pure strategies.” *Econometrica*, 57 (5), 1171–1214. [61]
- Singer, B. and S. Spilerman (1976), “The representation of social processes by Markov models.” *American Journal of Sociology*, 82 (1), 1–54. [56]
- Sobel, M. (1971), “Noncooperative stochastic games.” *Annals of Mathematical Statistics*, 42 (6), 1930–1935. [59]
- Starr, A. and Y. Ho (1969), “Nonzero-sum differential games.” *Journal of Optimization Theory and Applications*, 3 (3), 184–206. [54]
- Whitt, W. (1980), “Representation and approximation of noncooperative sequential games.” *SIAM Journal of Control and Optimization*, 18 (1), 33–48. [59]