

On Finding Local Nash Equilibria (and Only Local Nash Equilibria) in Zero-Sum Continuous Games

Eric Mazumdar

EMAZUMDAR@EECS.BERKELEY.EDU

*Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720-1776, USA*

Michael I. Jordan

JORDAN@CS.BERKELEY.EDU

*Division of Computer Science and Department of Statistics
University of California, Berkeley, CA 94720-1776, USA*

S. Shankar Sastry

SASTRY@COE.BERKELEY.EDU

*Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720-1776, USA*

Editor:

Abstract

We propose local symplectic surgery, a two-timescale procedure for finding local Nash equilibria in two-player zero-sum games. We first show that previous gradient-based algorithms cannot guarantee convergence to local Nash equilibria due to the existence of non-Nash stationary points. By taking advantage of the differential structure of the game, we construct an algorithm for which the local Nash equilibria are the only attracting fixed points. We also show that the algorithm exhibits no oscillatory behaviors in neighborhoods of equilibria and show that it has the same per-iteration complexity as other recently proposed algorithms. We conclude by validating the algorithm on two numerical examples: a toy example with multiple Nash equilibria and a non-Nash equilibrium, and the training of a small generative adversarial network (GAN).

1. Introduction

The classical problem of finding Nash equilibria in multi-player games has been a focus of intense research in computer science, control theory, economics and mathematics (Basar and Olsder, 1998; Nisan et al., 2007; C. Daskalakis, 2009). Some connections have been made between this extensive literature and machine learning (see, e.g., Cesa-Bianchi and Lugosi, 2006; Banerjee and Peng, 2003; Foerster et al., 2017), but these connections have focused principally on decision-making by single agents and multiple agents, and not on the burgeoning pattern-recognition side of machine learning, with its focus on large data sets and simple gradient-based algorithms for prediction and inference. This gap has begun to close in recent years, due to new formulations of learning problems as involving competition between subsystems that are construed as adversaries (Goodfellow et al., 2014), the need to robustify learning systems with regard to against actual adversaries (Xu et al., 2009) and with regard to mismatch between assumptions and data-generating mechanisms (Yang, 2011; Giordano et al., 2018), and an increasing awareness that real-world machine-learning systems are often embedded in larger economic systems or networks (Jordan, 2018).

These emerging connections bring significant algorithmic and conceptual challenges to the fore. Indeed, while gradient-based learning has been a major success in machine learning, both

in theory and in practice, work on gradient-based algorithms in game theory has often highlighted their limitations. For example, gradient-based approaches are known to be difficult to tune and train (Daskalakis et al., 2017; Mescheder et al., 2017; Hommes and Ochea, 2012; Balduzzi et al., 2018), and recent work has shown that gradient-based learning will almost surely avoid a subset of the local Nash equilibria in general-sum games (Mazumdar and Ratliff). Moreover, there is no shortage of work showing that gradient-based algorithms can converge to limit cycles or even diverge in game-theoretic settings (Benaïm and Hirsch, 1999; Hommes and Ochea, 2012; Daskalakis et al., 2017; Mertikopoulos et al., 2018b).

These drawbacks have led to a renewed interest in approaches to finding the Nash equilibria of zero-sum games, or equivalently, to solving saddle point problems. Recent work has attempted to use second-order information to reduce oscillations around equilibria and speed up convergence to fixed points of the gradient dynamics (Mescheder et al., 2017; Balduzzi et al., 2018). Other recent approaches have attempted to tackle the problem from the variational inequality perspective but also with an eye on reducing oscillatory behaviors (Mertikopoulos et al., 2018a; Gidel et al., 2018).

None of these approaches, however, address a fundamental issue that arises in zero-sum games. As we will discuss, the set of attracting fixed points for the gradient dynamics in zero-sum games can include critical points that are not Nash equilibria. In fact, any saddle point of the underlying function that does not satisfy a particular alignment condition of a Nash equilibrium is a candidate attracting equilibrium for the gradient dynamics. Further, as we show, these points are attracting for a variety of recently proposed adjustments to gradient-based algorithms, including consensus optimization (Mescheder et al., 2017), the symplectic gradient adjustment (Balduzzi et al., 2018), and a two-timescale version of simultaneous gradient descent (Heusel et al., 2017). Moreover, we show by counterexample that these algorithms can all converge to non-Nash stationary points.

We present a new gradient-based algorithm for finding the local Nash equilibria of two-player zero-sum games and prove that the only stationary points to which the algorithm can converge are local Nash equilibria. Our algorithm makes essential use of the underlying structure of zero-sum games. To obtain our theoretical results we work in continuous time—via an ordinary differential equation (ODE)—and our algorithm is obtained via a discretization of the ODE. While a naive discretization would require a matrix inversion and would be computationally burdensome, our discretization is a two-timescale discretization that avoids matrix inversion entirely and is of a similar computational complexity as that of other gradient-based algorithms.

The paper is organized as follows. In Section 2 we define our notation and the problem we address. In Section 3 we define the limiting ODE that we would like our algorithm to follow and show that it has the desirable property that its only limit points are local Nash equilibria of the game. In Section 4 we introduce local symplectic surgery, a two-timescale procedure that asymptotically tracks the limiting ODE and show that it can be implemented efficiently. Finally, in Section 5 we present two numerical examples to validate the algorithm. The first is a toy example with three local Nash equilibria, and one non-Nash fixed point. We show that simultaneous gradient descent and other recently proposed algorithms for zero-sum games can converge to any of the four points while the proposed algorithm only converges to the local Nash equilibria. The second example is a small generative adversarial network (GAN), where we show that the proposed algorithm converges to a suitable solution within a similar number of steps as simultaneous gradient descent.

2. Preliminaries

We consider a two-player game, in which one player tries to minimize a function, $f : \mathbb{R}^d \rightarrow \mathbb{R}$, with respect to their decision variable $x \in \mathbb{R}^{d_x}$, and the other player aims to maximize f with respect to their decision variable $y \in \mathbb{R}^{d_y}$, where $d = d_y + d_x$. We write such a game as $\mathcal{G} = \{(f, -f), \mathbb{R}^d\}$, since the second player can be seen as minimizing $-f$. We assume that neither player knows anything about the critical points of f , but that both players follow the rules of the game. Such a situation arises naturally when training machine learning algorithms (e.g., training generative adversarial networks or in multi-agent reinforcement learning). Without restricting f , and assuming both players are non-cooperative, the best they can hope to achieve is a *local Nash equilibrium*; i.e., a point (x^*, y^*) that satisfies

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*),$$

for all x and y in neighborhoods of x^* and y^* respectively. Such equilibria are locally optimal for both players with respect to their own decision variable, meaning that neither player has an incentive to unilaterally deviate from such a point. As was shown in Ratliff et al. (2013), generically, local Nash equilibria will satisfy slightly stronger conditions, namely they will be differential Nash equilibria (DNE):

Definition 1 A strategy $(x^*, y^*) \in \mathbb{R}^n$ is a differential Nash equilibrium if:

- $D_x f(x^*, y^*) = 0$ and $D_y f(x^*, y^*) = 0$.
- $D_{xx}^2 f(x^*, y^*) \succ 0$, and $D_{yy}^2 f(x^*, y^*) \prec 0$.

Here $D_x f$ and $D_y f$ denote the partial derivatives of f with respect to x and y respectively, and $D_{xx}^2 f$ and $D_{yy}^2 f$ denote the matrices of second derivatives of f with respect to x and y . Both differential and local Nash equilibria in two-player zero-sum games are, by definition, special saddle points of the function f that satisfy a particular *alignment condition* with respect to the player's decision variables. Indeed, the definition of differential Nash equilibria, which holds for almost all local Nash equilibria in a formal mathematical sense, makes this condition clear: the directions of positive and negative curvature of the function f at a local Nash equilibria must be *aligned* with the minimizing and maximizing player's decision variables respectively.

We note that the key difference between local and differential Nash equilibria is that $D_{xx}^2 f(x, y)$, and $D_{yy}^2 f(x, y)$ are required to be definite instead of semidefinite. This distinction simplifies our analysis while still allowing our results to hold for almost all continuous games.

2.1 Issues with gradient-based algorithms in zero-sum games

Having introduced local Nash equilibria as the solution concept of interest, we now consider how to find such solutions, and in particular we highlight some issues with gradient-based algorithms in zero-sum continuous games. The most common method of finding local Nash equilibria in such games is to have both players randomly initialize their variables (x_0, y_0) and then follow their respective gradients. That is, at each step $n = 1, 2, \dots$, each agent updates their variable as follows:

$$\begin{aligned} x_{n+1} &= x_n - \gamma_n D_x f(x_n, y_n) \\ y_{n+1} &= y_n + \gamma_n D_y f(x_n, y_n), \end{aligned}$$

where $\{\gamma_n\}_{n=0}^{\infty}$ is a sequence of step sizes. The minimizing player performs gradient descent on their cost while the maximizing player ascends their gradient. We refer to this algorithm as simultaneous gradient descent (simGD). To simplify the notation, we let $z = (x, y)$, and define the vector-valued function $\omega : \mathbb{R}^d \rightarrow \mathbb{R}^d$ as:

$$\omega(z) = \begin{bmatrix} D_x f(x, y) \\ -D_y f(x, y) \end{bmatrix}.$$

In this notation, the simGD update is given by:

$$z_{n+1} = z_n - \gamma_n \omega(z_n). \quad (1)$$

Since (1) is in the form of a discrete-time dynamical system, it is natural to examine its limiting behavior through the lens of dynamical systems theory. Intuitively, given a properly chosen sequence of step sizes, (1) should have the same limiting behavior as the continuous-time flow:

$$\dot{z} = -\omega(z). \quad (2)$$

We can analyze this flow in neighborhoods of equilibria by studying the Jacobian matrix of ω , denoted $J : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$:

$$J(z) = \begin{bmatrix} D_{xx}^2 f(x, y) & D_{yx}^2 f(x, y) \\ -D_{xy}^2 f(x, y) & -D_{yy}^2 f(x, y) \end{bmatrix}. \quad (3)$$

We remark that the diagonal blocks of $J(z)$ are always symmetric and $D_{xy}^2 f = (D_{yx}^2 f)^T$. Thus $J(z)$ can be written as the sum of a block symmetric matrix $S(z)$ and a block anti-symmetric matrix $A(z)$, where:

$$S(z) = \begin{bmatrix} D_{xx}^2 f(z) & 0 \\ 0 & -D_{yy}^2 f(z) \end{bmatrix}; \quad A(z) = \begin{bmatrix} 0 & D_{yx}^2 f(z) \\ -D_{xy}^2 f(z) & 0 \end{bmatrix}.$$

Given the structure of the Jacobian, we can now draw links between differential Nash equilibria and equilibrium concepts in dynamical systems theory. We focus on hyperbolic critical points of ω .

Definition 2 A strategy $z \in \mathbb{R}^d$ is a critical point of ω if $\omega(z) = 0$. It is a hyperbolic critical point if $Re(\lambda) \neq 0$ for all $\lambda \in \text{spec}(J(z^*))$, where $Re(\lambda)$, denotes the real part of the eigenvalue λ of $J(z^*)$.

It is well known that hyperbolic critical points are generic among critical points of smooth dynamical systems (see e.g. (Sastry, 1999)), meaning that our focus on hyperbolic critical points is not very restrictive. Of particular interest are locally asymptotically stable equilibria of the dynamics.

Definition 3 A strategy $z^* \in \mathbb{R}^d$ is a locally asymptotically stable equilibrium (LASE) of the continuous-time dynamics $\dot{z} = -\omega(z)$ if $\omega(z^*) = 0$ and $Re(\lambda) > 0$ for all $\lambda \in \text{spec}(J(z^*))$.

LASE have the desirable property that they are locally exponentially attracting under the flow of $-\omega$. This implies that a properly discretized version of $\dot{z} = -\omega(z)$ will also converge exponentially fast in a neighborhood of such points. LASE are the only attracting hyperbolic equilibria. Thus,

making statements about all the LASE of a certain continuous-time dynamical system allows us to characterize all attracting hyperbolic equilibria.

As shown in Ratliff et al. (2013) and Nagarajan and Kolter (2017), the fact that all differential Nash equilibria are critical points of ω coupled with the structure of J in zero-sum games guarantees that all differential Nash equilibria of the game are LASE of the gradient dynamics. However the converse is not true. The structure present in zero-sum games is not enough to ensure that the differential Nash equilibria are the only LASE of the gradient dynamics. When either $D_{xx}^2 f$ or $D_{yy}^2 f$ is indefinite at a critical point of $\omega(z)$, the Jacobian can still have eigenvalues with strictly positive real parts.

Example 1 Consider a matrix $M \in \mathbb{R}^{2 \times 2}$ having the form:

$$M = \begin{bmatrix} a & c \\ -c & -b \end{bmatrix},$$

where $a, b \in \mathbb{R}$ and $a, b > 0$. These conditions imply that M cannot be the Jacobian of ω at an local Nash equilibria. However, if $b < a$ and $c^2 > ab$, both of the eigenvalues of M will have strictly positive real parts, and such a point could still be a LASE of the gradient dynamics.

Such points, which we refer to as non-Nash LASE of (2), are what makes having guarantees on the convergence of algorithms in zero-sum games particularly difficult. Non-Nash LASE are not locally optimal for both players, and may not even be optimal for one of the players. By definition, at least one of the two players has a direction in which they would move to unilaterally decrease their cost. Such points arise solely due to the gradient dynamics, and persist even in other gradient-based dynamics suggested in the literature. In Appendix B, we show that three recent algorithms for finding local Nash equilibria in zero-sum continuous games—consensus optimization, symplectic gradient adjustment, and a two-time scale version of simGD—are susceptible to converge to such points and therefore have no guarantees of convergence to local Nash equilibria. We note that such points can be very common since every saddle point of f that is not a local Nash equilibrium is a candidate non-Nash LASE of the gradient dynamics. Further, local minima or maxima of f could also be non-Nash LASE of the gradient dynamics.

To understand how non-Nash equilibria can be attracting under the flow of $-\omega$, we again analyze the Jacobian of ω . At such points, the symmetric matrix $S(z)$ must have both positive and negative eigenvalues. The sum of S with A , however, has eigenvalues with strictly positive real part. Thus, the anti-symmetric matrix $A(z)$ can be seen as stabilizing such points.

Previous gradient-based algorithms for zero-sum games have also pinpointed the matrix A as the source of problems in zero-sum games, however they focus on a different issue. Consensus optimization (Mescheder et al., 2017) and the symplectic gradient adjustment (Balduzzi et al., 2018) both seek to adjust the gradient dynamics to reduce oscillatory behaviors in neighborhoods of stable equilibria. Since the matrix $A(z)$ is anti-symmetric, it has only imaginary eigenvalues. If it dominates S , then the eigenvalues of J can have a large imaginary component. This leads to oscillations around equilibria that have been shown empirically to slow down convergence (Mescheder et al., 2017). Both of these adjustments rely on tunable hyper-parameters to achieve their goals. Their effectiveness is therefore highly reliant on the choice of parameter. Further, as shown in Appendix B neither of the adjustments are able to rule out convergence to non-Nash equilibria.

A second promising line of research into theoretically sound methods of finding the Nash equilibria of zero-sum games has approached the issue from the perspective of variational inequalities

(Mertikopoulos et al., 2018a; Gidel et al., 2018). In Mertikopoulos et al. (2018a) extragradient methods were used to solve coherent saddle point problems and reduce oscillations when converging to saddle points. In such problems, however, all saddle points of the function f are assumed to be local Nash equilibria, and thus the issue of converging to non-Nash equilibria is assumed away. Similarly, by assuming that ω is monotone, as in the theoretical treatment of the averaging scheme proposed in Gidel et al. (2018), the cost function is implicitly assumed to be convex-concave. This in turn implies that the Jacobian satisfies the conditions for a Nash equilibrium everywhere. The behavior of their approaches in more general zero-sum games with less structure (like the training of GANs) is therefore not well known. Moreover, since their approach relies on averaging the gradients, they do not fundamentally change the nature of the critical points of simGD.

In the following sections we propose an algorithm for which the only LASE are the differential Nash equilibria of the game. We also show that, regardless of the choice of hyper-parameter, the Jacobian of the new dynamics at LASE has real eigenvalues, which means that the dynamics cannot exhibit oscillatory behaviors around differential Nash equilibria.

3. Constructing the limiting differential equation

In this section we define the continuous-time flow that our discrete-time algorithm should ideally follow.

Assumption 1 (Lipschitz assumptions on f and J) *Assume that $f \in \mathcal{C}^3(\mathbb{R}^d, \mathbb{R})$ and f and ω are L_f -Lipschitz and L_ω -Lipschitz respectively. Finally assume that all critical points of ω are hyperbolic.*

We do not require $J(z)$ to be invertible everywhere, but only at the critical points of f .

Now, consider the continuous-time flow:

$$\dot{z} = -h(z) = -\frac{1}{2} \left(\omega(z) + J^T(z) (J^T(z)J(z) + \lambda(z)I)^{-1} J^T(z)\omega(z) \right), \quad (4)$$

where $\lambda \in \mathcal{C}^2(\mathbb{R}^d, \mathbb{R})$ is such that $0 \leq \lambda(z) \leq \xi$ for all $z \in \mathbb{R}^d$ and $\xi > 0$ and $\lambda(z) = 0 \iff \omega(z) = 0$.

The function λ ensures that, even when J is not invertible everywhere, the inverse matrix in (4) exists. The vanishing condition $\lambda(z) = 0 \iff \omega(z) = 0$ ensures us that the Jacobian of the adjustment term is exactly $J^T(z)$ at differential Nash equilibria.

The dynamics introduced in (4) can be seen as an adjusted version of the gradient dynamics where the adjustment term only allows trajectories to approach critical points of ω along the players' axes. If a critical point is not locally optimal for one of the players (i.e., it is a non-Nash critical point) then that player can push the dynamics out of a neighborhood of that point. The mechanism is easier to see if we assume J is invertible and set $\lambda(z) \equiv 0$. This results in the following dynamics:

$$\dot{z} = -\frac{1}{2} \left(\omega(z) + J^T(z)J^{-1}(z)\omega(z) \right). \quad (5)$$

In this simplified form we can see that the Jacobian of the adjustment is approximately J^T when $\|\omega(z)\|_2$ is small. This approximation is exact at critical points of ω . Adding this adjustment term to ω exactly cancels out the rotational part of the vector field contributed by the antisymmetric matrix

$A(z)$ in a neighborhood of critical points. Since we identified $A(z)$ as the source of oscillatory behaviors and non-Nash equilibria in Section 2, this adjustment addresses both of these issues. The following theorem establishes this formally.

Theorem 4 *Under Assumption 1 and if $J^T(z) (J^T(z)J(z) + \lambda(z))^{-1} J^T(z)\omega(z) \neq -\omega(z) \forall z \in \mathbb{R}^d$, the continuous-time dynamical system $\dot{z} = -h(z)$ satisfies:*

- z is a LASE of $\dot{z} = -h(z) \iff z$ is a differential Nash equilibrium of the game $\{(f, -f), \mathbb{R}^d\}$.
- If z is a critical point of $\dot{z} = -h(z)$, then the Jacobian of h at z has real eigenvalues.

Proof We first show that:

$$h(z) = 0 \iff \omega(z) = 0.$$

Clearly, $\omega(z) = 0 \implies h(z) = 0$. To show the converse, we assume that $h(z) = 0$ but $\omega(z) \neq 0$. This implies that:

$$J^T(z) (J^T(z)J(z) + \lambda(z)I)^{-1} J^T(z)\omega(z) = -\omega(z).$$

Since we assumed that this cannot be true, we must have that $h(z) = 0 \implies \omega(z) = 0$.

Having shown that under our assumptions, the critical points of h are the same as those of ω , we now note that the Jacobian of $h(z)$ at a critical point must have the form:

$$J_h(z) = \frac{1}{2} (J(z) + J^T(z)(J^T(z)J(z))^{-1}J^T(z)J(z)) = \frac{1}{2} (J(z) + J^T(z)) = S(z).$$

By assumption, at critical points, $J(z)$ is invertible and $\lambda(z) = 0$. Given that $\omega(z) = 0$, terms that include $\omega(z)$ disappear, and the adjustment term contributes only a factor of $J^T(z)$ to the Jacobian of h at a critical point. This exactly cancels out the antisymmetric part of the Jacobian of ω . The Jacobian of h is therefore symmetric at critical points of ω and has positive eigenvalues only when $D_{xx}^2 f(z) \succ 0$ and $D_{yy}^2 f(z) \prec 0$.

Since these are also the conditions for differential Nash equilibria, all differential Nash equilibria of \mathcal{G} must be LASE of $\dot{z} = -h(z)$. Further, non-Nash LASE of $\dot{z} = -\omega(z)$ cannot be LASE of $\dot{z} = -h(z)$, since by definition either $D_{xx}^2 f(z)$ or $D_{yy}^2 f(z)$ is indefinite at such points. To show the second part of the theorem, we simply note that J_h must be symmetric at all critical points which in turn implies that it has only real eigenvalues. \blacksquare

The continuous-time dynamical system therefore solves both of the problems we highlighted in Section 2, for any choice of the function λ that satisfies our assumptions. The assumption that $\omega(z)$ is never an eigenvector of $J^T(z) (J^T(z)J(z) + \lambda(z)I)^{-1} J^T(z)$ with an eigenvalue of -1 ensures that the adjustment does not create new critical points. In high dimensions the assumption is mild since the scenario is extremely specific, but it is also possible to show that this assumption can be removed entirely by adding a time-varying term to (4) while still retaining the theoretical guarantees. We show this in Appendix A.

Theorem 4 shows that the only attracting hyperbolic equilibria of the limiting ordinary differential equation (ODE) are the differential Nash equilibria of the game. Also, since $J_h(z)$ is symmetric at critical points of ω , if either $D_{xx}^2 f(z)$ or $-D_{yy}^2 f(z)$ has at least one negative eigenvalue then such

a point would be a linearly unstable equilibrium of $\dot{z} = -h(z)$. Such points are linearly unstable and are therefore almost surely avoided when the algorithm is randomly initialized (Benaïm and Hirsch, 1995; Sastry, 1999).

Theorem 4 also guarantees that the continuous-time dynamics do not oscillate near critical points. Oscillatory behaviors, as outlined in Mescheder et al. (2017), are known to slow down convergence of the discretized version of the process. Reducing oscillations near critical points is the main goal of consensus optimization (Mescheder et al., 2017) and the symplectic gradient adjustment (Balduzzi et al., 2018). However, for both algorithms, the extent to which they are able to reduce the oscillations depends on the choice of hyperparameter. The proposed dynamics achieves this for any $\lambda(z)$ that satisfies our assumptions.

We close this section by noting that one can pre-multiply the adjustment term by some function $g(z)$ such that $\omega(z) = 0 \implies g(z) = 1$ while still retaining the theoretical properties described in Theorem 4. Such a function can be used to ensure that the dynamics closely track a trajectory of simGD except in neighborhoods of critical points. For example, if the matrix J is ill-conditioned, such a term could be used to ensure that the adjustment does not dominate the underlying gradient dynamics. In Section 5 we give an example of such a damping function.

4. Two-timescale approximation

Given the limiting ODE, we could perform a straightforward Euler discretization to obtain a discrete-time update having the form:

$$z_{n+1} = z_n - \gamma h(z_n).$$

However, due to the matrix inversion, such a discrete-time update would be prohibitively expensive to implement in high-dimensional parameter spaces like those encountered when training GANs. To solve this problem, we now introduce a two-timescale approximation to the continuous-time dynamics that has the same limiting behavior, but is much faster to compute at each iteration, than the simple discretization. Since this procedure serves to exactly remove the symplectic part, $A(z)$ of Jacobian in neighborhoods of hyperbolic critical points, we refer to this two-timescale procedure as local symplectic surgery (LSS). In Appendix A we derive the two-timescale update rule for the time-varying version of the limiting ODE and show that it also has the same properties.

The two-timescale approximation to (4) is given by:

$$\begin{aligned} z_{n+1} &= z_n - a_n h_1(z_n, v_n) \\ v_{n+1} &= v_n - b_n h_2(z_n, v_n), \end{aligned} \tag{6}$$

where h_1 and h_2 are defined as:

$$\begin{aligned} h_1(z, v) &= \frac{1}{2} (\omega(z) + J^T(z)v) \\ h_2(z, v) &= J^T(z)J(z)v - J^T(z)\omega(z) + \lambda(z)v, \end{aligned}$$

and the sequences of step sizes $\{a_n\}_{n=0}^\infty, \{b_n\}_{n=0}^\infty$ satisfy the following assumptions:

Assumption 2 (Assumptions on the step sizes) *The sequences $\{a_n\}_{n=0}^\infty$ and $\{b_n\}_{n=0}^\infty$ satisfy:*

- $\sum_{i=1}^\infty a_i = \infty$, and $\sum_{i=1}^\infty b_i = \infty$;

Algorithm 1 Local Symplectic Surgery

Input Functions f, ω, J, λ ; Step sizes a_n, b_n ; Initial values (x_0, y_0, v_0)

Initialize $(x, y, v, n) \leftarrow (x_0, y_0, v_0, 0)$

while not converged **do**

$g_x \leftarrow D_x [f(x, y) + \omega^T(x, y)v]$
 $g_y \leftarrow D_y [-f(x, y) + \omega^T(x, y)v]$
 $g_v \leftarrow D_v [\|J(x, y)v - \omega(x, y)\|_2^2 + \lambda(x, y)\|v\|_2^2]$
 $x \leftarrow x - a_n g_x$
 $y \leftarrow y - a_n g_y$
 $v \leftarrow v - b_n g_v$
 $n \leftarrow n + 1$

end while

Output $(x^*, y^*) \leftarrow (x, y)$

- $\sum_{i=1}^{\infty} a_i^2 < \infty$, and $\sum_{i=1}^{\infty} b_i^2 < \infty$;
- $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = 0$.

We note that h_2 is Lipschitz continuous in v uniformly in z under Assumption 1.

The v process performs gradient descent on a regularized version of least squares, where the regularization is governed by $\lambda(z)$. If the v_n process is on a faster time scale, the intuition is that it will first converge to $(J^T(z_n)J(z_n) + \lambda(n)I)^{-1}\omega(z_n)$, and then z_n will track the limiting ODE in (4). In the next section we show that this behavior holds even in the presence of noise.

The key benefit to the two-timescale process is that z_{n+1} and v_{n+1} can be computed efficiently since neither require a matrix inversion. In fact, as we show in Appendix C, the computation can be done with Jacobian-vector products with the same order of complexity as that of simGD, consensus optimization, and the symplectic gradient adjustment. This insight gives rise to the procedure outlined in Algorithm 1.

4.1 Long-term behavior of the two-timescale approximation

We now show that LSS asymptotically tracks the limiting ODE even in the presence of noise. This implies that the algorithm has the same limiting behavior as (4). In particular, our setup allows us to treat the case where one only has access to unbiased estimates of h_1 and h_2 at each iteration. This is the setting most likely to be encountered in practice, for example in the case of training GANs in a mini-batch setting.

We assume that we have access to estimators \hat{h}_1 and \hat{h}_2 such that:

$$\begin{aligned} \mathbb{E} [\hat{h}_1(z, v)] &= \omega(z) + J^T(z)v \\ \mathbb{E} [\hat{h}_2(z, v)] &= J^T(z)J(z)v + J^T(z)\omega(z). \end{aligned}$$

To place this in the form of classical two-timescale stochastic approximation processes, we write each estimator \hat{h}_1 and \hat{h}_2 as the sum of its mean and zero-mean noise processes M^z and M^v re-

spectively. This results in the following two timescale process:

$$\begin{aligned} z_{n+1} &= z_n - a_n[\omega(z_n) + J^T(z_n)v_n + M_{n+1}^z] \\ v_{n+1} &= v_n - b_n[J^T(z_n)J(z_n)v_n + J^T(z_n)\omega(z_n) + \lambda(z_n)v_n + M_{n+1}^v]. \end{aligned} \quad (7)$$

We assume that the noise processes satisfy the following standard conditions (Benaïm, 1999; Borkar, 2008):

Assumption 3 *Assumptions on the noise: Define the filtration \mathcal{F}_n :*

$$\mathcal{F}_n = \sigma(z_0, v_0, M_1^v, M_1^z, \dots, M_n^z, M_n^v),$$

for $n \geq 0$. Given \mathcal{F}_n , we assume that:

- M_{n+1}^v and M_{n+1}^z are conditionally independent given \mathcal{F}_n for $n \geq 0$.
- $\mathbb{E}[M_{n+1}^v | \mathcal{F}_n] = 0$ and $\mathbb{E}[M_{n+1}^z | \mathcal{F}_n] = 0$ for $n \geq 0$.
- $\mathbb{E}[\|M_{n+1}^z\| | \mathcal{F}_n] \leq c_z(1 + \|z_n\|)$ and $\mathbb{E}[\|M_{n+1}^v\| | \mathcal{F}_n] \leq c_v(1 + \|z_n\|)$ almost surely for some positive constants c_z and c_v .

Given our assumptions on the estimator, cost function, and step sizes we now show that (7) asymptotically tracks a trajectory of the continuous-time dynamics almost surely. Since h_1 , h_2 , and $v^*(z) = (J^T(z)J(z) + \lambda(z)I)^{-1}\omega(z)$ are not uniformly Lipschitz continuous in both z and v , we cannot directly invoke results from the literature. Instead, we adapt the proof of Theorem 2 in Chapter 6 of Borkar (2008) to show that $v_n \rightarrow v^*(z_n)$ almost surely. We then invoke Proposition 4.1 from Benaïm (1999) to show that z_n asymptotically tracks $\dot{z} = -h(z)$. We note that this approach only holds on the event $\{\sup_n \|z_n\| + \|v_n\| < \infty\}$. Thus, if the stochastic approximation process remains bounded, then under our assumptions we are sure to track a trajectory of the limiting ODE.

Lemma 5 *Under Assumptions 1-3, and on the event $\{\sup_n \|z_n\|_2 + \|v_n\|_2 < \infty\}$:*

$$(z_n, v_n) \rightarrow \{(z, v^*(z)) : z \in \mathbb{R}^d\},$$

almost surely.

Proof We first rewrite (7) as:

$$\begin{aligned} z_{n+1} &= z_n - b_n \left[\frac{a_n}{b_n} h_1(z_n, v_n) + \bar{M}_{n+1}^z \right] \\ v_{n+1} &= v_n - b_n [h_2(z_n, v_n) + M_{n+1}^v], \end{aligned}$$

where $\bar{M}_{n+1}^z = \frac{a_n}{b_n} M_{n+1}^z$. By assumption, $\frac{a_n}{b_n} \rightarrow 0$. Since h_1 is locally Lipschitz continuous, it is bounded on the event $\{\sup_n \|z_n\|_2 + \|v_n\|_2 < \infty\}$. Thus, $\frac{a_n}{b_n} h_1(z_n, v_n) \rightarrow 0$ almost surely.

From Lemma 1 in Chapter 6 of Borkar (2008), the above processes, on the event $\{\sup_n \|z_n\|_2 + \|v_n\|_2 < \infty\}$, converge almost surely to internally chain-transitive invariant sets of $\dot{v} = -h_2(z, v)$ and $\dot{z} = 0$. Since, for a fixed z , $h_2(z, v)$ is a Lipschitz continuous function of v with a globally asymptotically stable equilibrium at $(J^T(z)J(z) + \lambda(z)I)^{-1}\omega(z)$, the claim follows. \blacksquare

Having shown that $\|v_n - v^*(z_n)\|_2 \rightarrow 0$ almost surely, we now show that z_n will asymptotically track a trajectory of the limiting ODE. Let us first define $z(t, s, z_s)$ for $t \geq s$ to be the trajectory of $\dot{z} = -h(z)$ starting at z_s at time s .

Theorem 6 *Given Assumptions 1-3, let $t_n = \sum_{i=0}^{n-1} a_i$. On the event $\{\sup_n \|z_n\|_2 + \|v_n\|_2 < \infty\}$, for any integer $K > 0$ we have:*

$$\lim_{n \rightarrow \infty} \sup_{0 \leq h \leq K} \|z_{n+h} - z(t_{n+h}, t_n, z_n)\|_2 = 0.$$

Proof The proof makes use of Propositions 4.1 and 4.2 in Benaïm (1999) which are supplied in Appendix E.

We first rewrite the z_n process as:

$$z_{n+1} = z_n - a_n [h(z) - J^T(z_n)(v^*(z_n) - v_n) + M_{n+1}^z].$$

We note that, from Lemma 5, $(v^*(z_n) - v_n) \rightarrow 0$ almost surely. Since $\|J^T(z_n)\|_2 < L_\omega$, we can write this process as:

$$z_{n+1} = z_n - a_n [h(z) - \chi_n + M_{n+1}^z],$$

where $\chi_n \rightarrow 0$ almost surely. Since h is continuously differentiable, it is locally Lipschitz, and on the event $\{\sup_n \|z_n\| + \|v_n\| < \infty\}$ it is bounded. It thus induces a continuous globally integrable vector field, and therefore satisfies the assumptions for Propositions 4.1 in Benaïm (1999). Further, by assumption the sequence of step sizes and martingale difference sequences satisfy the assumptions of Proposition 4.2 in Benaïm (1999). Invoking Proposition 4.1 and 4.2 in Benaïm (1999) gives us the desired result. \blacksquare

Theorem 6 guarantees that LSS asymptotically tracks a trajectory of the limiting ODE. The approximation will therefore avoid non-Nash equilibria of the gradient dynamics. Further, the only locally asymptotically stable points for LSS must be the differential Nash equilibria of the game.

5. Numerical Examples

We now present two numerical examples that illustrate the performance of both the limiting ODE and LSS. The first is a zero-sum game played over a function in \mathbb{R}^2 that allows us to observe the behavior of both the limiting ODE around both local Nash and non-Nash equilibria. In the second example we use LSS to train a small generative adversarial network (GAN) to learn a mixture of eight Gaussians. Further numerical experiments and comments are provided in Appendix D.

5.1 2-D example

For the first example, we consider the game based on the following function f in \mathbb{R}^2 :

$$f(x, y) = e^{-0.01(x^2+y^2)}((0.3x^2 + y)^2 + (0.5y^2 + x)^2).$$

This function is a fourth-order polynomial that is scaled by an exponential to ensure that it is bounded. The gradient dynamics $\dot{z} = -\omega(z)$ associated with function have four LASE. By evaluating the Jacobian of ω at these points we find that three of the LASE are local Nash equilibria.

These are denoted by ‘x’ in Figure 1. The fourth LASE is a non-Nash equilibrium which is denoted with a star. In Figure 1, we plot the sample paths of both simGD and our limiting ODE from the same initial positions, shown with red dots. We clearly see that simGD converges to all four LASE, depending on the initialization. Our algorithm, on the other hand, only converges to the local Nash equilibria. When initialized close to the non-Nash equilibrium it diverges from the simGD path and ends up converging to a LNE. This numerical example also allows us to study the behavior of our

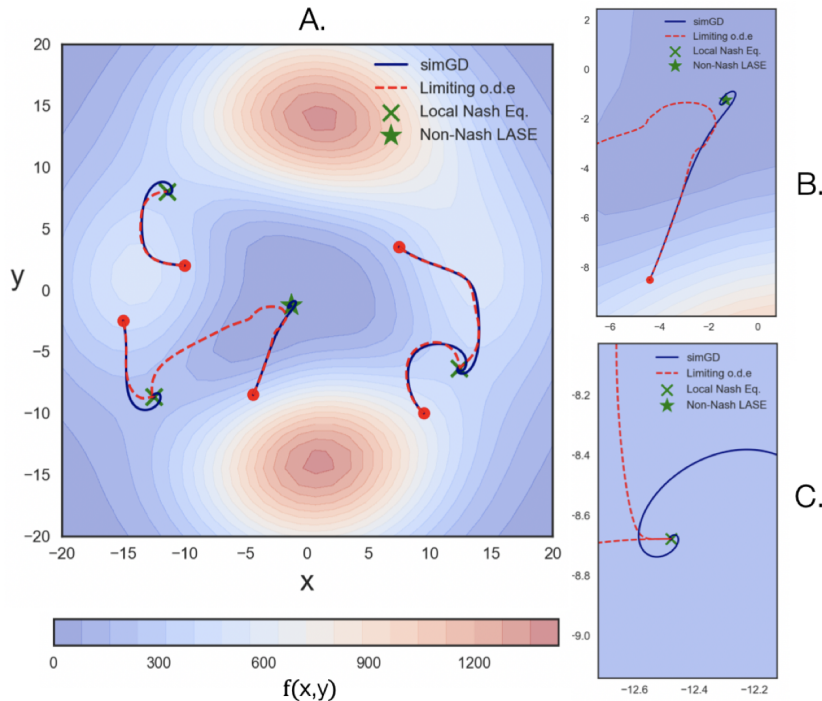


Figure 1: A. Convergence of simGD and the limiting ODE (4) to different LASE of the game. The local Nash equilibria are denoted by ‘x’ and the non-Nash equilibrium by a ‘*’. The heat map is used to show that this game is being played over a function and the Nash equilibria and non-Nash equilibria are all at different payoffs. B. From the same initialization where simGD converges to the non-Nash equilibrium, the ODE converges instead to a local Nash equilibria. C. We empirically validate that the ODE does not exhibit oscillatory behaviors near local Nash equilibria.

algorithm around LASE. By focusing on a local Nash equilibrium, as in Figure 1B, we observe that the limiting ODE approaches it directly even when simGD displays oscillatory behaviors. This empirically validates the second part of Theorem 4.

In Figure 2 we empirically validate that LSS asymptotically tracks the limiting ODE. When the fast timescale has not converged, the process tracks the gradient dynamics. Once it has converged however, we see that it closely tracks the limiting ODE which leads it to converge to only the local Nash equilibria. This behavior highlights an issue with the two-timescale approach. Since the non-Nash equilibria of the gradient dynamics are saddle points for the new dynamics they can slow down convergence. However, the process will eventually escape such points (Benaïm, 1999).

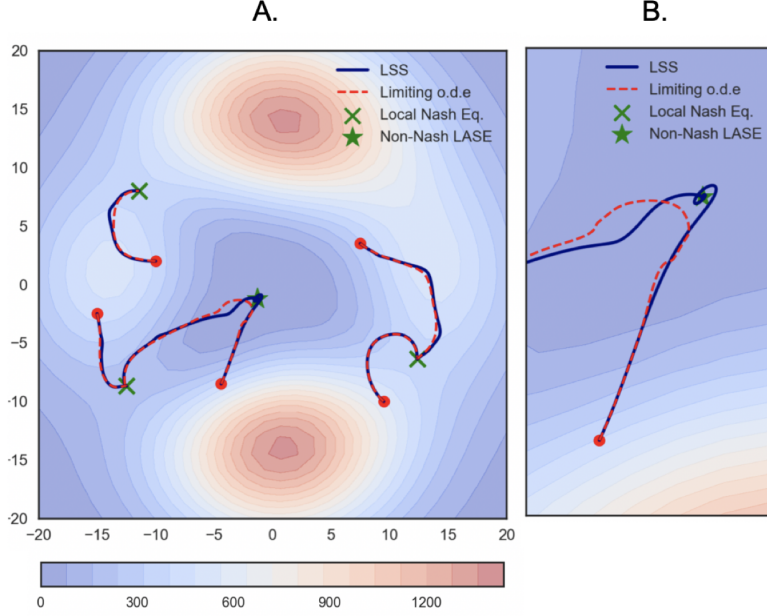


Figure 2: Behavior of the two-timescale approximation with respect to the limiting ODE. A. The two-timescale procedure closely tracks the limiting ODE from all four initializations. B. Focusing on the non-Nash equilibrium, the two-timescale procedure tracks the gradient-dynamics closely which leads it to begin oscillating around the equilibrium. Once the fast timescale has converged however, the process quickly exits the neighborhood of the non-Nash equilibrium and closely tracks the limiting ODE.

In our numerical experiments we let $\lambda(z) = \xi_1 \left(1 - e^{-\|\omega(z)\|^2}\right)$. We also make use of a damping function g as described in Section 3. The limiting ODE is therefore given by:

$$\dot{z} = -(\omega(z) + e^{-\xi_2 \|v\|^2} v),$$

where $v = J^T(z)(J^T(z)J(z) + \lambda(z)I)^{-1}J^T(z)\omega(z)$. For the two-timescale process, since there is no noise we use constant step sizes and the following update:

$$\begin{aligned} z_{n+1} &= z_n - \gamma_1(\omega(z_n) + e^{-\xi_2 \|J^T(z_n)v_n\|^2} J^T(z_n)v_n) \\ v_{n+1} &= v_n - \gamma_2(J^T(z_n)J(z_n)v_n + \lambda(z_n)v_n - J^T(z_n)\omega(z_n)), \end{aligned}$$

where $\xi_1 = 1e - 4, \xi_2 = 1e - 4, \gamma_1 = 0.004$, and $\gamma_2 = 0.005$.

5.2 Generative adversarial network

We now train a generative adversarial network with LSS. Both the discriminator and generator are fully connected neural networks with four hidden layers of 16 neurons each. The tanh activation function is used since it satisfies the smoothness assumptions for our functions. For the latent space, we use a 16-dimensional Gaussian with mean zero and covariance $\Sigma = 0.1I_{16}$. The ground truth

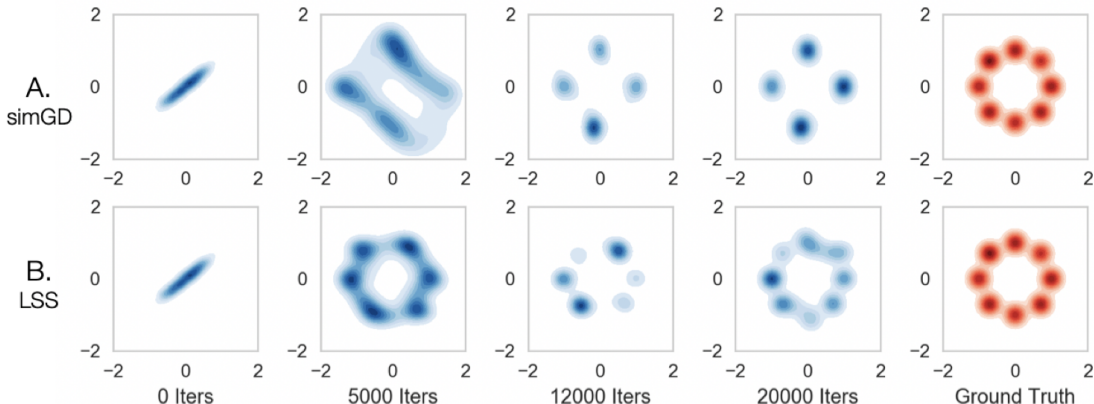


Figure 3: A GAN trained with A. simGD and B. LSS.

distribution is a mixture of eight Gaussians with their modes uniformly spaced around the unit circle and covariance $\Sigma = 1e - 4I_2$. In Figure 3, we show the progression of the generator at 0, 5000, 12000, and 20000 iterations for a GAN initialized with the same weights and biases and then trained with A. simGD and B. LSS. We can see empirically that, in this example, LSS converges to the true distribution while simGD quickly suffers mode collapse, showing how the adjusted dynamics can lead to convergence to better equilibria. Further numerical experiments are shown in Appendix D.

We caution that convergence rate per se is not necessarily a reasonable metric on which to compare performance in the GAN setting or in other game-theoretic settings. Competing algorithms may converge faster than our method when used to train GANs, but once because the competitors could be converging quickly to a non-Nash equilibrium, which is not desirable. Indeed, the optimal solution is known to be a local Nash equilibrium for GANs (Goodfellow et al., 2014; Nagarajan and Kolter, 2017). LSS may initially move towards a non-Nash equilibrium, while subsequently escaping the neighborhood of such points before converging. This will lead to a slower convergence rate, but a better quality solution.

6. Discussion

We have introduced local symplectic surgery, a new two-timescale algorithm for finding the local Nash equilibria of two-player zero-sum continuous games. We have established that this comes with the guarantee that the only hyperbolic critical points to which it can converge are the local Nash equilibria of the underlying game. This significantly improves upon previous methods for finding such points which, as shown in Appendix B, cannot give such guarantees. We have analyzed the asymptotic properties of the proposed algorithm and have shown that the algorithm can be implemented efficiently. Altogether, these results show that the proposed algorithm yields limit points with game-theoretic relevance while ruling out oscillations near those equilibria and having a similar per-iteration complexity as existing methods which do not come with the same guarantees. Our numerical examples allow us to empirically observe these properties.

It is important to emphasize that our analysis has been limited to neighborhoods of equilibria; the proposed algorithm can converge in principle to limit cycles at other locations of the space.

These are hard to rule out completely. Moreover, some of these limit cycles may actually have some game-theoretic relevance (Hommes and Ochea, 2012; Benaim and Hirsch, 1997). Another limitation of our analysis is that we have assumed the existence of local Nash equilibria in games. Showing that they exist and finding them is very hard to do in general. Our algorithm will converge to local Nash equilibria, but may diverge when the game does not admit equilibria or when the algorithm does not come any equilibria its region of attraction. Thus, divergence of our algorithm is not a certificate that no equilibria exist. Such caveats, however, are the same as those for other gradient-based approaches for finding local Nash equilibria.

Another drawback to our approach is the use of second-order information. Though the two-timescale approximation does not need access to the full Jacobian of the gradient dynamics, the update does involve computing Jacobian-vector products. This is similar to other recently proposed approaches but will be inherently slower to compute than pure first- or zeroth-order methods. Bridging this gap while retaining similar theoretical properties remains an interesting avenue of further research.

In all, we have shown that some of the inherent flaws to gradient-based methods in zero-sum games can be overcome by designing our algorithms to take advantage of the game-theoretic setting. Indeed, by using the structure of local Nash equilibria we designed an algorithm that has significantly stronger theoretical support than existing approaches.

References

- D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, 2018.
- B. Banerjee and J. Peng. Adaptive policy gradient in multiagent learning. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003.
- T. Basar and G. Olsder. *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics, 2 edition, 1998.
- M. Benaïm. Dynamics of stochastic approximation algorithms. In *Séminaire de Probabilités XXXIII*, pages 1–68. Springer Berlin Heidelberg, 1999.
- M. Benaïm and M. Hirsch. Dynamics of Morse-Smale urn processes. *Ergodic Theory and Dynamical Systems*, 15(6), 12 1995.
- M. Benaim and M. Hirsch. Learning processes, mixed equilibria and dynamical systems arising from repeated games. *Games and Economic Behavior*, 1997.
- M. Benaïm and M. Hirsch. Mixed equilibria and dynamical systems arising from fictitious play in perturbed games. *Games and Economic Behavior*, 29:36–72, 1999.
- V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- C. Papadimitriou C. Daskalakis, P. Goldberg. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39:195–259, 02 2009.

- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, UK, 2006.
- C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training GANs with Optimism. *arxiv:1711.00141*, 2017.
- J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with opponent-learning awareness. *CoRR*, abs/1709.04326, 2017.
- G. Gidel, H. Berard, P. Vincent, and S. Lacoste-Julien. A variational inequality perspective on generative adversarial nets. *CoRR*, 2018. URL <http://arxiv.org/abs/1802.10551>.
- R. Giordano, T. Broderick, and M. I. Jordan. Covariances, robustness, and variational Bayes. *Journal of Machine Learning Research*, 2018.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arxiv:1406.2661*, 2014.
- M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems 30*, 12 2017.
- C. H. Hommes and M. I. Ochea. Multiple equilibria and limit cycles in evolutionary games with logit dynamics. *Games and Economic Behavior*, 74(1):434–441, 2012.
- M. I. Jordan. Artificial intelligence: The revolution hasn’t happened yet. *Medium*, 2018.
- E. Mazumdar and L. J. Ratliff. On the convergence of gradient-based learning in continuous games. *ArXiv e-prints*.
- P. Mertikopoulos, H. Zenati, B. Lecouat, C. Foo, V. Chandrasekhar, and G. Piliouras. Mirror descent in saddle-point problems: Going the extra (gradient) mile. *CoRR*, abs/1807.02629, 2018a.
- Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *proceedings of the 29th annual ACM-SIAM symposium on discrete algorithms*, 2018b.
- L. M. Mescheder, S. Nowozin, and A. Geiger. The numerics of GANs. In *Advances in Neural Information Processing Systems 30*, 2017.
- V. Nagarajan and Z. Kolter. Gradient descent GAN optimization is locally stable. In *Advances in Neural Information Processing Systems 30*. 2017.
- N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, Cambridge, UK, 2007.
- L. J. Ratliff, S. A. Burden, and S. S. Sastry. Characterization and computation of local Nash equilibria in continuous games. In *Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing*, pages 917–924, Oct 2013.
- S. S. Sastry. *Nonlinear Systems*. Springer New York, 1999.

H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, December 2009. ISSN 1532-4435.

L. Yang. Active learning with a drifting distribution. In *Advances in Neural Information Processing Systems*. 2011.

Appendix A. Time-varying adjustment

In this section we analyze a slightly different version of (4) that allows us to remove the assumption that $\omega(z)$ is never an eigenvector of $J^T(z)(J^T(z)J(z) + \lambda(z)I)^{-1}J^T(z)$ with associated eigenvalue -1 . Though this assumption is relatively mild, since intuitively it will be very rare that ω is exactly the eigenvector of the adjustment matrix, we show that by adding a third term to (4) we can remove it entirely while retaining our theoretical guarantees. The new dynamics are constructed by adding a time-varying term to the dynamics that goes to zero only when $\omega(z)$ is zero. This guarantees that the only critical points of the limiting dynamics are the critical points of ω . The analysis of these dynamics is slightly more involved and requires generalizations of the definition of a LASE to handle time-varying dynamics. We first define an equilibrium of a potentially time-varying dynamical system $\dot{\theta} = g(\theta, t)$ as a point θ^* such that $g(\theta^*, t) \equiv 0$ for all $t \geq 0$. We can now generalize the definition of a LASE to the time-varying setting.

Definition 7 *A strategy $\theta^* \in \mathbb{R}^d$ is a locally uniformly asymptotically stable equilibrium of the time-varying continuous time dynamics $\dot{\theta} = -f(\theta, t)$ if θ^* is an equilibrium of $\dot{\theta} = -f(\theta, t)$, and $D_{\theta}f(\theta^*, t) \equiv J(\theta^*)$ and $\text{Re}(\lambda) > 0$ for all $\lambda \in \text{spec}(J(\theta^*))$.*

Locally uniformly asymptotically stable equilibria, under this definition, also have the property that they are locally exponentially attracting under the flow, $\dot{\theta} = -f(\theta, t)$. Further, since the linearization around a locally uniformly asymptotically stable equilibrium is time-invariant, we can still invoke converse Lyapunov theorems like those presented in Sastry (1999) when deriving the non-asymptotic bounds.

Having defined equilibria and a generalization of LASE for time-varying systems, we now introduce a time-varying version of the continuous-time ODE presented in Section 3 which allows us to remove the assumption that $\omega(z)$ is never an eigenvector of $J^T(z)(J^T(z)J(z) + \lambda(z)I)^{-1}J^T(z)$ with associated eigenvalue -1 . The limiting ODE is given by:

$$\dot{z} = -h_{TV}(z, t) = -(h(z) + g_{TV}(z, t)), \quad (8)$$

where $h(z)$ is as described in Section 3, $g(z, t)$ can be decomposed as:

$$g_{TV}(z, t) = \lambda_1(z)u(t),$$

where $\lambda_1 \in C^2(\mathbb{R}^d, \mathbb{R})$ satisfies:

- $0 \leq \lambda_1(z) \leq \xi_2$ for all $z \in \mathbb{R}^d$.
- $\lambda_1(z) = 0 \iff \omega(z) = 0$.
- $\omega(z) = 0 \implies D_z \lambda_1(z) = 0$,

and where $u : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ satisfies:

- $\nexists t_0 \geq 0$ such that $D_t u(t) \equiv 0 \quad \forall t \geq t_0$.
- $\|u(t)\| \leq \xi_3 \quad \forall t \geq 0$.

Thus we require that the time-varying adjustment term g_{TV} must be bounded and is equal to zero only when $\omega(z) = 0$. Most importantly, we require that for any z that is not a critical point of ω , g_{TV} must be changing in time. An example of a g_{TV} that satisfies these requirements is:

$$g_{TV}(z, t) = \xi_1 \left(1 - e^{-\xi_2 \|\omega(z)\|^2} \right) \cos(t) u_0, \quad (9)$$

for $u_0 \in \mathbb{R}^d$, $u_0 \neq 0$ and $\xi_1, \xi_2 > 0$.

These conditions, as the next theorem shows, allow us to guarantee that the only locally asymptotically stable equilibria are the differential Nash equilibria of the game.

Theorem 8 *Under Assumption 1 the continuous-time dynamical system $\dot{z} = -h_{TV}(z, t)$ satisfies:*

- z is a locally uniformly asymptotically stable equilibrium of $\dot{z} = -h_{TV}(z, t) \iff z$ is a DNE of the game $\{(f, -f), \mathbb{R}^d\}$.
- If z is an equilibrium point of $\dot{z} = -h_{TV}(z, t)$, then the Jacobian of h_{TV} at z is time-invariant and has real eigenvalues.

Proof We first show that:

$$h_{TV}(z, t) \equiv 0 \quad \forall t \geq 0 \iff \omega(z) = 0.$$

By construction $\omega(z) = 0 \implies h_{TV}(z, t) \equiv 0 \quad \forall t \geq 0$. To show the converse, we assume that there exists a z such that $h_{TV}(z, t) \equiv 0 \quad \forall t \geq 0$ but $\omega(z) \neq 0$. This implies that:

$$-g_{TV}(z, t) = \omega(z) + J^T(z) (J^T(z)J(z) + \lambda(z)I)^{-1} J^T(z)\omega(z) \quad \forall t \geq 0.$$

Since z is a constant and $\lambda_1(z) > 0$, taking the derivative of both sides with respect to t gives us the following condition on g under our assumption:

$$D_t u(t) = 0 \quad \forall t \geq 0.$$

By assumption this cannot be true. Thus, we have a contradiction and $h_{TV}(z, t) \equiv 0 \quad \forall t \geq 0 \implies \omega(z) = 0$.

Having shown that the critical points of $\dot{z} = -h_{TV}(z, t)$ are the same as that of $\dot{z} = -\omega(z)$, we now note that the Jacobian of $h_{TV}(z, t)$, at critical points, must be $S(z)$. Under the same development as the proof of Theorem 4 the Jacobian of h_{TV} is given by:

$$J_{TV}(z) = J(z) + J^T(z) + (D_z g_{TV}(z, t)).$$

Again, by construction $D_z g_{TV}(z, t) = 0$ when $\omega(z) = 0$. The third term therefore disappears and we have that $J_{TV}(z) = S(z)$. The proof now follows from that of Theorem 4. \blacksquare

We have shown that adding a time-varying term to the original adjusted dynamics allows us to remove the assumption that the adjustment term is never exactly $-\omega(z)$. As in Section 3 we can now construct a two-timescale process that asymptotically tracks (8). We assume that $u(t)$ is a deterministic function of a trajectory of an ODE:

$$\dot{\theta} = -h_3(\theta),$$

with a fixed initial condition θ_0 such that $u(t) = g_{TV}(\theta(t))$. We assume that f is Lipschitz-continuous and g_{TV} is continuous and bounded. Note that under our assumptions, $\|g_{TV}(\theta)\| \leq \xi_3$ for all $\theta \in \mathbb{R}^{d_\theta}$.

The form of $u(t)$ introduced in (9), $u(t) = \cos(t)u_0$ can be written as $u(t) = ([1, 0] * \theta(t))u_0$, where θ satisfies the linear dynamical system:

$$\dot{\theta} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \theta,$$

with $\theta_0 = [1, 0]$.

Given this setup, the continuous-time dynamics can be written as:

$$\begin{aligned} \dot{\theta} &= -h_3(\theta) \\ \dot{z} &= -h_4(z, \theta), \end{aligned} \tag{10}$$

where:

$$h_4(z, \theta) = \frac{1}{2}(\omega(z) + J^T(z) (J^T(z)J(z) + \lambda(z)I)^{-1} J^T(z)\omega(z) + \lambda_1(z)u(\theta)).$$

Having made this further assumption on the time-varying term, we now introduce the two-timescale process that asymptotically tracks (10). This process is given by:

$$\begin{aligned} \theta_{n+1} &= \theta_n - a_n h_3(\theta_n) \\ z_{n+1} &= z_n - a_n \hat{h}_5(z_n, v_n, \theta_n) \\ v_{n+1} &= v_n - b_n \hat{h}_6(z_n, v_n), \end{aligned} \tag{11}$$

where $\theta_0 = \bar{\theta}$

$$\begin{aligned} \mathbb{E}[\hat{h}_5(z, v, \theta)] &= h_5(z, v, \theta) := \frac{1}{2} (\omega(z) + J^T(z)v) + \lambda_1(z)u(\theta) \\ \mathbb{E}[\hat{h}_6(z, v)] &= h_6(z, v) := J^T(z)J(z)v - J^T(z)\omega(z) + \lambda(z)v. \end{aligned}$$

Proceeding as in Section 3, we write $\hat{h}_5(z, v, \theta) = h_5(z, v, \theta) + M^z$ and $\hat{h}_6(z, v) = h_6(z, v) + M^v$ where M^z and M^v are martingale difference sequences satisfying Assumption 3. We note that the θ_n process is deterministic.

This two-timescale process gives rise to the time-varying version of local symplectic surgery (TVLSS) outlined in Algorithm 2.

We can now proceed as in Section 4 to show that (11) has the same limiting behavior as the limiting ODE in (10).

Algorithm 2 Time-varying Local Symplectic Surgery

Input Functions $f, \omega, J, \lambda, g_{TV}$; Step sizes a_n, b_n ; Initial values (x_0, y_0, v_0)

Initialize $(x, y, v, n) \leftarrow (x_0, y_0, v_0, 0)$

while not converged **do**

$g_x \leftarrow D_x [f(x, y) + \omega^T(x, y)v]$
 $g_y \leftarrow D_y [-f(x, y) + \omega^T(x, y)v]$
 $g_v \leftarrow D_v [\|J(x, y)v - \omega(x, y)\|_2^2 + \lambda(x, y)\|v\|_2^2]$
 $x \leftarrow x - a_n(g_x + g_{TV}(x, y, n))$
 $y \leftarrow y - a_n(g_y + g_{TV}(x, y, n))$
 $v \leftarrow v - b_n g_v$
 $n \leftarrow n + 1$

end while

Output $(x^*, y^*) \leftarrow (x, y)$

Lemma 9 *Under Assumptions 1-3, and on the event $\{\sup_n \|z_n\| + \|v_n\| + \|\theta_n\| < \infty\}$:*

$$(\theta_n, z_n, v_n) \rightarrow \{(\theta, z, v^*(z)) : (z, \theta) \in \mathbb{R}^d \times \mathbb{R}^{d_\theta}\}$$

almost surely.

The proof of this lemma is exactly the same as that of Lemma 5, and we do not repeat the argument. We can now invoke Proposition 4.1 and 4.2 from Benaïm (1999) to show that the two-timescale process tracks a trajectory of the limiting ODE. For $t \geq s$, let $\theta(t, s, \theta_s)$ and $z(t, s, z_s)$ be the trajectories of the dynamical systems introduced in (10), starting at states θ_s and z_s respectively at time s .

Theorem 10 *Let Assumptions 1-3 hold. Further, assume h_3 is Lipschitz continuous and define $t_n = \sum_{i=0}^{n-1} a_i$. Then on the event $\{\sup_n \|z_n\| + \|v_n\| + \|\theta_n\| < \infty\}$, for any integer $K > 0$:*

$$\lim_{n \rightarrow \infty} \sup_{0 \leq h \leq K} \|z_{n+h} - z(t_{n+h}, t_n, z_n)\| = 0$$

$$\lim_{n \rightarrow \infty} \sup_{0 \leq h \leq K} \|\theta_{n+h} - \theta(t_{n+h}, t_n, z_n)\| = 0.$$

The proof of Theorem 10 is exactly like that of Theorem 6.

Together, Theorem 8 and Theorem 10 show that our approach can be modified to allow us to relax some of our assumptions, while still maintaining the theoretical properties of our algorithms.

Appendix B. Counter-examples for other algorithms

We now show that three state-of-the-art algorithms for finding the local Nash equilibria of zero-sum games are all attracted to non-Nash equilibria. This implies that these algorithms do not have guarantees on the local optimality of their limit points and cannot give guarantees on convergence to local Nash equilibria. To do this we construct a simple counter-example in \mathbb{R}^2 that demonstrates that all of the algorithms are attracted to some non-Nash limit point. The particular game we use is described below.

Example 2 Consider the game $\mathcal{G} = \{(f, -f), \mathbb{R}^2\}$ where:

$$f(x, y) = \frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} 1 & 1 \\ 1 & 0.1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Staying with our earlier notation, the player with variable x seeks to minimize f , and the player with variable y minimizes $-f$.

This game has only one critical point, $(x, y) = (0, 0)$, and the combined gradient dynamics for this game are linear and are given by:

$$\omega(x, y) = \begin{bmatrix} 1 & 1 \\ -1 & -0.1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

For all $x, y \in \mathbb{R}$ the Jacobian of ω is given by:

$$J(x, y) = \begin{bmatrix} 1 & 1 \\ -1 & -0.1 \end{bmatrix}.$$

Since the diagonal is not strictly positive, $(x, y) = (0, 0)$ is not an LNE. However, since the eigenvalues of $J(x, y)$ are $0.45 + 0.835i$ and $0.45 - 0.835i$, $(x, y) = (0, 0)$ is a LASE of the gradient dynamics.

We first show that even running SGA on two timescales as suggested by Heusel et al. (2017), cannot guarantee convergence to only local Nash equilibria. We assume that the maximizing player is on the slower timescale, and, in the following proposition, show that SGA on two timescales can still converge to non-Nash fixed points.

Proposition 11 *Simultaneous gradient ascent on two timescales can converge to non-Nash equilibria.*

Proof Consider the game introduced in Example 2, and the following dynamics:

$$\begin{aligned} x_{n+1} &= x_n - a_n(\omega_1(x_n, y_n)) \\ y_{n+1} &= y_n - b_n(\omega_2(x_n, y_n)), \end{aligned}$$

where ω_i denotes the i th component of ω as described in Example 2 and a_n, b_n are sequences of step sizes satisfying Assumption 2.

Since the dynamics $\dot{x} = -x - y$ have a unique equilibrium $x = -y$ for a fixed $y \in \mathbb{R}$ and the x process is on the faster timescale, Chapter 4 in Borkar (2008), assures us that $x_n \rightarrow -y_n$ asymptotically. Now, assuming that x_n has converged, we analyze the slower timescale. Plugging in for x_n , we get that, asymptotically, the dynamics will track:

$$\dot{y} = -y + 0.1y = -0.9y.$$

Since these dynamics have a unique (exponentially) stable equilibrium at $y = 0$, $(x_n, y_n) \rightarrow (0, 0)$. As we showed, the origin is non-Nash, so we are done. \blacksquare

The previous proposition shows that a two-timescale version of simultaneous gradient ascent will still be susceptible to non-Nash equilibria. This implies that such an approach cannot guarantee convergence to only local Nash equilibria. We now show that the consensus optimization approach introduced in Mescheder et al. (2017), can also converge to non-Nash points.

Proposition 12 *Consensus optimization can converge to non-Nash equilibria.*

Proof The update in consensus optimization is given by:

$$z_{n+1} = z_n - \gamma (\omega(z_n) + \lambda J^T(z_n)\omega(z_n)),$$

where $\lambda > 0$ is a hyperparameter to be chosen. We note that the signs are different than in Mescheder et al. (2017) because we consider simultaneous gradient descent as the base algorithm while they consider simultaneous gradient ascent. The limiting dynamics of this approach is given by:

$$\dot{z} = - (\omega(z) + \lambda J^T(z)\omega(z)).$$

At a critical point z where $\omega(z) = 0$, the Jacobian of this dynamics is given by:

$$J_{CO}(z) = (J(z) + \lambda J^T(z)J(z)).$$

Now, in the game described in Example 2, the Jacobian of the consensus optimization dynamics would be:

$$J_{CO}(x, y) = \begin{bmatrix} 1 + 2\lambda & 1 + 1.1\lambda \\ -1 + 1.1\lambda & -0.1 + 1.01\lambda \end{bmatrix}.$$

For $\lambda > 0$, $J_{CO}(x, y)$ has eigenvalues with strictly positive real parts, which implies that, with any choice of the hyper-parameter λ , consensus optimization will converge to the non-Nash equilibrium at (0,0) in Example 2. ■

The preceding proposition shows that the adjustment to the gradient dynamics proposed in Mescheder et al. (2017) does not solve the issue of non-Nash fixed points. This is not surprising since the primary goal for the algorithm is to reduce oscillation around equilibria and speed up convergence to the stable equilibria of the gradient dynamics. We note that as shown in Theorem 4, the proposed algorithm also achieves this goal.

The final algorithm we consider is the symplectic gradient adjustment proposed in Balduzzi et al. (2018). In the text, the authors do remark that the adjustment is not enough to guarantee convergence to only Nash equilibrium. For completeness, we make this assertion concrete by showing that the adjustment term is not enough to avoid the non-Nash equilibrium in Example 2.

Proposition 13 *The symplectic gradient adjustment to the gradient dynamics can converge to non-Nash equilibria.*

Proof The dynamics resulting from the symplectic gradient adjustment is given by:

$$z_{n+1} = z_n - \gamma \left(\omega(z_n) + \frac{\lambda}{2} (J(z_n) - J^T(z_n))^T \omega(z_n) \right),$$

where $\lambda > 0$ is a hyperparameter to be chosen. The limiting dynamics of this algorithm is given by:

$$\dot{z} = - \left(\omega(z) + \frac{\lambda}{2} (J(z) - J^T(z))^T \omega(z) \right).$$

At a critical point z where $\omega(z) = 0$, the Jacobian of the above dynamics is given by:

$$J_{SGA}(z) = (J(z) + \frac{\lambda}{2}(J(z) - J^T(z))^T J(z)).$$

Now, in the game described in Example 2, the Jacobian of the SGA dynamics is given by:

$$J_{SGA}(x, y) = \begin{bmatrix} 1 + \frac{\lambda}{2} & 1 + 0.1\frac{\lambda}{2} \\ 1 + \frac{\lambda}{2} & -0.1 + \frac{\lambda}{2} \end{bmatrix}.$$

This has eigenvalues with strictly positive real parts for all values of $\lambda > 0$, which implies that the symplectic gradient adjustment will converge to the non-Nash equilibrium at (0,0) in Example 2. ■

Appendix C. Efficient computation of the two-timescale update

We now show that by leveraging auto-differentiation tools we can compute the stochastic approximation update efficiently. In particular, using Jacobian-vector products, the computation of h_1 and h_2 can be done relatively quickly. We first note that a Jacobian-vector product calculates $J^T(z)u$ for a constant vector $u \in \mathbb{R}^d$, by calculating the gradient of $\omega^T(z)u$ with respect to v . This allows us to write the x and y updates in Algorithm 1 in a clean form.

The next proposition shows that, using the structure of J that was discussed in Section 2, the term $J(z)v$ can also be efficiently computed. Together these results show that the per-iteration complexity of LSS is on the same order as that of consensus optimization and the symplectic gradient adjustment.

Proposition 14 *The computation of $J(z)v$ requires two Jacobian-vector products, for $v \in \mathbb{R}^d$ a constant vector.*

Proof Let $v = (v_1, v_2)^T$ where $v_1 \in \mathbb{R}^{d_x}$ and $v_2 \in \mathbb{R}^{d_y}$. Then $J(z)v$ can be written as:

$$J(z)v = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} J^T(z) \begin{bmatrix} v_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} J^T(z) \begin{bmatrix} 0 \\ v_2 \end{bmatrix}.$$

We note that the above expression is only possible due to the structure of the Jacobian in two-player zero-sum games. Having written $J(z)$ as above, it is now clear that computing $J(z)v$ will require two Jacobian-vector products. ■

Appendix D. Further numerical examples

We now present further numerical experiments as well as the specific experimental setup we employed when training GANs. In Figure 4 we show further numerical experiments that show the training of a generative adversarial network from various initial conditions. In Figure 4A, both simGD and LSS quickly converge to the correct solution. In Figure 4B, however, simGD only correctly identifies five of the Gaussians in the mixture while LSS converges to the correct solution.

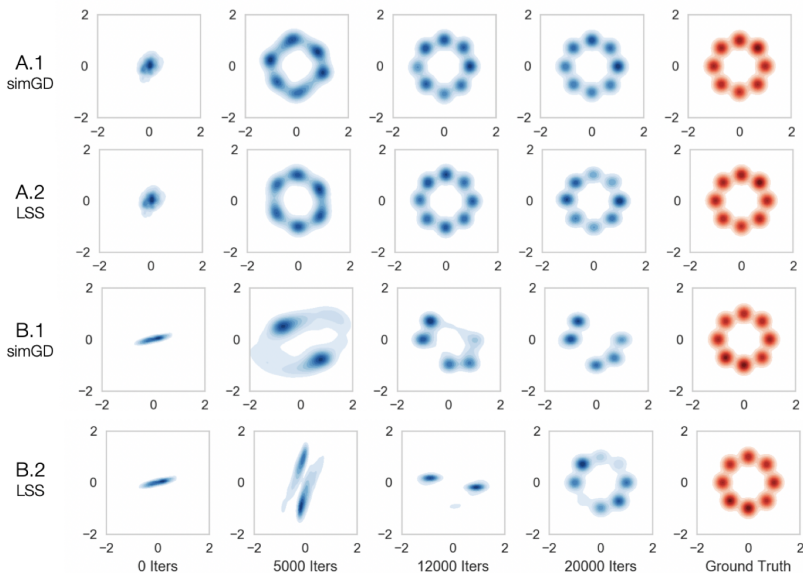


Figure 4: A. A GAN trained with (A.1) simGD and (A.2) with our two-timescale method. b. A GAN trained with (B.1) simGD and (B.2) with our two-timescale method.

For the training of the GAN, we randomly initialize the weights and biases using the standard TensorFlow initializations. For both the implementation of simGD and LSS used to generate Figure 3 and Figure 4, we used the RMSProp optimizer with step size $2e - 4$ for the x and y processes. For the v process in LSS, we used the RMSProp optimizer with step size $1e - 5$. We note that these do not satisfy our assumptions on the step size sequences for the two-timescale process, but are meant to show that the approach still works in practical settings. Lastly, we chose a damping function $g(z) = e^{-0.001\|\omega(z)\|^2}$ and a batch size of 128.

Appendix E. Useful propositions

In this section we present a useful theorem from previous work that we invoke in our proofs. The first proposition, a combination of Proposition 4.1 and 4.2 from Benaïm (1999), gives us conditions under which a stochastic approximation process is an asymptotic pseudo-trajectory of the underlying ODE.

Proposition 15 *Let h be a continuous globally integrable vector field. Further, let $x(t, s, x_s)$ for $t \geq s$ be a trajectory of the dynamical system $\dot{x} = -h(x)$ starting from state x_s at time s . Finally let the stochastic approximation process be given by:*

$$x_{n+1} = x_n + a_n(h(x_n) + \chi_n + M_{n+1}),$$

where:

1. $\sup_n \|x_n\| < \infty$
2. $\sup_n \mathbb{E}[\|M_n\|^2] < \infty$

3. $\sum_{i=0}^{\infty} a_n^2 < \infty$

4. $\sum_{i=0}^{\infty} a_n = \infty$

5. $\lim_{n \rightarrow \infty} \chi_n = 0$ *almost surely*.

Then:

$$\lim_{n \rightarrow \infty} \sup_{0 \leq h \leq K} \|x_{n+h} - x(t_{n+h}, t_n, x_n)\| = 0.$$