Linear Lower Bounds and Conditioning of Differentiable Games

Adam Ibrahim [1] **Waïss Azizian** [2] **Gauthier Gidel** [1] **Ioannis Mitliagkas** [1]

Abstract

Recent successes of game-theoretic formulations in ML have caused a resurgence of research interest in differentiable games. Overwhelmingly, that research focuses on methods and upper bounds on their speed of convergence. In this work, we approach the question of fundamental iteration complexity by providing lower bounds to complement the linear (i.e. geometric) upper bounds observed in the literature on a wide class of problems. We cast saddle-point and min-max problems as 2-player games. We leverage tools from single-objective convex optimisation to propose new linear lower bounds for convex-concave games. Notably, we give a linear lower bound for n-player differentiable games, by using the spectral properties of the update operator. We then propose a new definition of the condition number arising from our lower bound analysis. Unlike past definitions, our condition number captures the fact that linear rates are possible in games, even in the absence of strong convexity or strong concavity in the variables.

1. Introduction

Game formulations arise commonly in many fields, such as game theory (Harker and Pang, 1990), machine learning (Kim and Boyd, 2008; Goodfellow et al., 2014), and computer vision (Chambolle and Pock, 2011; Wang et al., 2014) among others, and encompass saddle-point problems (Palaniappan and Bach, 2016; Chambolle and Pock, 2011; Chen et al., 2017).

The machine learning community has been overwhelmingly using gradient-based methods to train differentiable games (Goodfellow et al., 2014; Salimans et al., 2016). These methods are not designed with game dynamics in mind (Mescheder et al., 2017), and to make matters worse, have been often tuned suboptimally (Gidel et al., 2019b). A recent series of publications in machine learning brings in tools from the minimax and game theory literature to offer better, faster alternatives (Daskalakis et al., 2018; Gidel et al., 2019a;b). This exciting trend begs the question: how fast can we go? Knowing the fundamental limits of this class of problems is critical in steering future algorithmic research.

In order to answer this question, the optimisation literature contains a few different approaches based on the distance between the iterates at a step t and the optimal choice of parameters. Given an optimisation algorithm, it is possible to show under certain assumptions on the objectives that this error is in $\mathcal{O}\left(\rho^t\right)$, where the rate of convergence ρ depends on the algorithm (Nesterov, 2004). If $\rho \in (0, 1)$, we say that the rate of convergence is linear, which corresponds to the error decaying exponentially fast. Hence, a lower bound on the rate of convergence limits how fast an algorithm may converge. This is important as it helps establish the tightness of upper bounds, which happens when they are matched by the lower bounds, and may otherwise indicate possible acceleration of the method considered. For example, in single-objective optimisation, the lower bound on the rate of convergence of first-order black box algorithms is known to be linear for smooth, strongly convex objectives, and can be derived via a domino-like coverage argument by Nesterov (2004). Another recent, spectral approach by Arjevani et al. (2016) complement these results by proposing linear lower bounds for a large class of optimisers in finite-dimensional settings. As the lower bounds for Nesterov's accelerated gradient obtained by those techniques match the upper bound, we know that Nesterov's accelerated gradient is optimal within a large class of methods for smooth, strongly convex objectives. Additionally, in optimisation, a natural concept of condition number arises to describe the difficulty of μ-strongly convex, L-smooth objectives. This condition number is the only problem-dependent quantity that appears in both the upper and lower bounds and is given by $\kappa = L/\mu$ (Nesterov, 2004). In optimisation, there is a clear distinction between strongly convex objectives, where the condition number is finite and linear rates are achievable, and general convex objectives where the condition number can be undefined and only sublinear rates are possible in

*Equal contribution [1]Mila, University of Montreal [2]Ecole Normale Supérieure, Paris. Correspondence to: Adam Ibrahim <<first>.<last>@umontreal.ca>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

general.

When studying lower bounds for convex-concave min-max, one is faced with a number of distinct challenges compared to the optimisation setting. In particular, there is no universally accepted definition of a condition number. Some commonly used definitions, like the one used in Chambolle and Pock (2011); Palaniappan and Bach (2016), are undefined for bilinear problems, which lack strong convexity and strong concavity in the variables. This is problematic because we know that both extragradient and gradient methods with negative momentum achieve linear convergence in bilinear games (Korpelevich, 1976; Gidel et al., 2019b). *Can we get a condition number that captures the fact that linear rates are possible even in the absence of strong convexity and strong concavity?*

We show that it is possible by providing new lower bounds, obtained by casting saddle-point and min-max problems as games and leveraging existing proof techniques originally designed for smooth strongly convex, single-objective optimisation. These bounds also yield a meaningful condition number for the bilinear case, in the absence of strong convexity and strong concavity in the variables. Our contributions are summarised as follows:

1. We generalise *Nesterov's domino argument* and design a difficult min-max problem to derive a linear lower bound on the rate of convergence of several first-order black box optimisation algorithms for 2-player games and min-max problems. In order to get an asymptotic rate using the domino bound, one needs to resort to the analysis of infinite-dimensional problems.
2. We propose a linear lower bound for finite-dimensional problems by generalising the p-SCLI framework proposed by Arjevani et al. (2016) to n-objective optimisation algorithms. This lower bound stems from the spectral properties of the algorithms on quadratics, and is valid for any number of players, and in particular 2-player games and min-max problems. This bound is tight for $n = 1$ since it reduces to the one presented by Arjevani et al. (2016) for strongly convex, smooth single-player optimisation.
3. We provide a formulation of the condition number of 2-player games consistent with the existing literature on upper bounds for games and min-max problems. In particular, this condition number is finite for bilinear games.

After the results of this work were made available online, several researchers have proposed methods to match some of our bounds in the smooth strongly-convex-strongly-concave setting (Fallah et al., 2020; Lin et al., 2020) and the bilinear setting (Azizian et al., 2020), which is merely convex-concave, thereby establishing the tightness of some of the bounds and the optimality of those methods in those regimes.

The rest of the paper is organised as follows. We purposely discuss preliminaries first in Section 2 to introduce the general framework used to present in Section 3 the relevant literature in the context of our results. In Section 4, we provide lower bounds using Nesterov's domino argument, and in Section 5 we improve on those bounds using the spectral technique. We conclude with some discussion.

2. Preliminaries

2.1. Differentiable games

Following the definition of Balduzzi et al. (2018), a *differentiable game* is characterised by n players, each associated with a set of parameters $\mathbf{w}_i \in \mathbb{R}^{d_i}$ and a twice continuously differentiable objective function $l_i : \mathbb{R}^d \to \mathbb{R}$ of all the parameters $\mathbf{w} = (\mathbf{w}_1, ..., \mathbf{w}_n) \in \mathbb{R}^d$, where $d = \sum_{i=1}^n d_i$. In particular, if $\sum_{i=1}^n l_i(\mathbf{w}) = 0$, we say that the game is *zero-sum*.

Often, we seek to minimise the objectives l_i, and look for *Nash equilibria* $\mathbf{w}^* = (\mathbf{w}_1^*, ..., \mathbf{w}_n^*)$, which satisfy[1] for all i

$$\mathbf{w}_i^* \in \arg\min_{\mathbf{w}_i} l_i\left(\mathbf{w}_1^*, ..., \mathbf{w}_{i-1}^*, \mathbf{w}_i, \mathbf{w}_{i+1}^*, ..., \mathbf{w}_n^*\right) . \quad (1)$$

In order to find the Nash equilibria, we may look for stationary points, corresponding to the zeros of the vector field $\boldsymbol{v}(\mathbf{w}) = (\nabla_{\mathbf{w}_1} l_1(\mathbf{w}) \dots \nabla_{\mathbf{w}_n} l_n(\mathbf{w}))^\top$. In single-objective optimisation, which corresponds to a 1-player game, we know that stationary points of $\boldsymbol{v}$ do not necessarily represent minima of the objective function, and higher order information, such as the Hessian, is necessary to determine whether a stationary point is a minimum. The same is true for a game with several players (Balduzzi et al., 2018), where the *Jacobian* of $\boldsymbol{v}$, given by

$$\nabla \boldsymbol{v}(\mathbf{w}) = \begin{pmatrix} \nabla^2_{\mathbf{w}_1} l_1(\mathbf{w}) & \dots & \nabla_{\mathbf{w}_n} \nabla_{\mathbf{w}_1} l_1(\mathbf{w}) \\ \vdots & & \vdots \\ \nabla_{\mathbf{w}_1} \nabla_{\mathbf{w}_n} l_n(\mathbf{w}) & \dots & \nabla^2_{\mathbf{w}_n} l_n(\mathbf{w}) \end{pmatrix} \quad (2)$$

gives sufficient conditions to determine whether a stationary point is a Nash equilibrium. Note that our lower bound analysis encompasses games with stable stationary points that are not Nash equilibria.

[1] Of course, we could be trying to maximise some players' objectives, but we can without loss of generality work with minima since $\arg\max f = \arg\min(-f)$

2.2. Quadratic games

In order to gain insight on general games, we focus on quadratic games[2], corresponding to games with quadratic objectives l_i. In our analysis, we will mostly discuss two-player games, where the players respectively control the parameters $\boldsymbol{x} \in \mathbb{R}^{d_1}$ and $\boldsymbol{y} \in \mathbb{R}^{d_2}$. The quadratic objectives take the form

$$\begin{aligned} l_1(\boldsymbol{x}, \boldsymbol{y}) &= \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{S}_1 \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{M}_{12} \boldsymbol{y} + \boldsymbol{x}^\top \mathbf{b}_1 \\ l_2(\boldsymbol{x}, \boldsymbol{y}) &= \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{S}_2 \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{M}_{21} \boldsymbol{x} + \boldsymbol{y}^\top \mathbf{b}_2 \end{aligned} \tag{3}$$

with $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ symmetric. In that case the vector field is given by

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{y}) &= \begin{pmatrix} \boldsymbol{S}_1 \boldsymbol{x} + \boldsymbol{M}_{12} \boldsymbol{y} + \mathbf{b}_1 \\ \boldsymbol{M}_{21} \boldsymbol{x} + \boldsymbol{S}_2 \boldsymbol{y} + \mathbf{b}_2 \end{pmatrix} \\ &= \boldsymbol{A} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} + \mathbf{b} \\ \text{with } \boldsymbol{A} &\triangleq \begin{pmatrix} \boldsymbol{S}_1 & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{S}_2 \end{pmatrix}, \quad \mathbf{b} \triangleq \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \end{aligned} \tag{4}$$

where $\boldsymbol{A}$ is the Jacobian of $\boldsymbol{v}$. For n-player quadratic games, the vector field and Jacobian $\boldsymbol{A}$ take the form

$$\begin{aligned} \boldsymbol{v}(\mathbf{w}) &= \boldsymbol{A}\mathbf{w} + \mathbf{b} \\ \text{with } \boldsymbol{A} &\triangleq \begin{pmatrix} \boldsymbol{S}_1 & \dots & \boldsymbol{M}_{1n} \\ \vdots & & \vdots \\ \boldsymbol{M}_{n1} & \dots & \boldsymbol{S}_n \end{pmatrix}, \quad \mathbf{b} \triangleq \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \end{aligned} \tag{5}$$

where the $\boldsymbol{S}_i$ are symmetric. For more details on quadratic games, see Appendix A. Further assumptions on the dimensionality or properties of $\boldsymbol{S}_i$ will be introduced in the rest of the paper as they become relevant (e.g. positive semi-definiteness in the next subsection).

We shall henceforth refer to the Jacobian of the vector field of quadratic n-player games simply as the Jacobian, since our analysis will be solely based on quadratic objectives. Interestingly, the problem of finding $\mathbf{w}$ such that $\boldsymbol{v}(\mathbf{w}) = 0$ consists in solving a system of linear equations (SLE) (Richardson, 1911). In fact, several techniques to precondition systems of linear equation make use of casting the SLE as a game and optimising it with proximal methods, such as Benzi and Golub (2004).

In this paper, we will denote the spectrum of a matrix $\boldsymbol{M}$ by $\sigma(\boldsymbol{M})$, and define the *block spectral bounds* $\mu_1, \mu_2, \mu_{12}, L_1, L_2, L_{12}$ as constants bounding the spectra of the blocks in the Jacobian of eq. 4:

$$\begin{aligned} \mu_1 \leq |\sigma(\boldsymbol{S}_1)| \leq L_1 \qquad \mu_2 \leq |\sigma(\boldsymbol{S}_2)| \leq L_2 \\ \mu_{12}^2 \leq |\sigma(\boldsymbol{M}_{12}\boldsymbol{M}_{12}^\top)| \leq L_{12}^2 \end{aligned} \tag{6}$$

[2]Note that quadratic games are inherently relevant; e.g. in reinforcement learning to learn a linear value function from the mean squared projected Bellman error (Du et al., 2017)

where we assume that $\boldsymbol{M}_{12}$ is a wide or square matrix (if it is a tall matrix, we use $\mu_{12}^2 \leq |\sigma(\boldsymbol{M}_{12}^\top \boldsymbol{M}_{12})| \leq L_{12}^2$ to define μ_{12} and L_{12} instead of the last inequality of eq. 6).

2.3. Min-max of quadratics as 2-player quadratic games

Consider the family $\mathcal{P}$ of min-max problems of the form

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^{d_1}} \max_{\boldsymbol{y} \in \mathbb{R}^{d_2}} \quad & f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{y} + \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{S}_1 \boldsymbol{x} - \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{S}_2 \boldsymbol{y} \\ & \qquad + \boldsymbol{x}^\top \mathbf{b}_1 - \boldsymbol{y}^\top \mathbf{b}_2 + c \qquad (\mathcal{P}) \\ \text{where} \quad & \sigma(\boldsymbol{M}\boldsymbol{M}^\top),\ \sigma(\boldsymbol{S}_1),\ \sigma(\boldsymbol{S}_2) \subseteq [0, +\infty) \end{aligned}$$

with possible constraints and where the dimension need not be finite, e.g. $\boldsymbol{x}, \boldsymbol{y} \in \ell_2 \triangleq \{\mathbf{u} \in \mathbb{R}^{\mathbf{N}} \mid \sum_i^\infty \mathbf{u}_i^2 < \infty\}$. The optimisation of such a problem is equivalent to finding a pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that,

$$\boldsymbol{x}^* \in \arg\min f(\boldsymbol{x}, \boldsymbol{y}^*) \ \text{ and } \ \boldsymbol{y}^* \in \arg\max f(\boldsymbol{x}^*, \boldsymbol{y}) \tag{7}$$

Noting that $\arg\max f = \arg\min(-f)$, we get that this optimisation problem is equivalent to a zero-sum 2-player game with objectives $l_1 = -l_2 = f$ (see eq. 1). This problem can be reduced to searching for the Nash equilibria of the 2-player quadratic game with simplified objectives $l_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{S}_1 \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{y} + \boldsymbol{x}^\top \mathbf{b}_1$ and $l_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{S}_2 \boldsymbol{y} - \boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{y} + \boldsymbol{y}^\top \mathbf{b}_2$, where the $\boldsymbol{S}_i$ have been symmetrised (see Appendix A for an explanation of the symmetrisation of $\boldsymbol{S}_i$ and why the objectives can be simplified). Eq. 4 yields the vector field

$$\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} \boldsymbol{S}_1 & \boldsymbol{M} \\ -\boldsymbol{M}^\top & \boldsymbol{S}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} + \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \tag{8}$$

Therefore, the pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ from eq. 7 exists if and only if the corresponding games with vector field given in eq. 8 admit a Nash equilibrium since $\boldsymbol{S}_1, \boldsymbol{S}_2 \succeq 0$. Note that we could also go from a quadratic game satisfying the above to a min-max formulation. Hence, any lower bound on quadratic games of the form of eq. 8 is a lower bound on min-max problems (in $\mathcal{P}$), and vice versa.

3. Background

3.1. Existing bounds for 2-player quadratic min-max problems

Some upper bounds on the rate of convergence of certain optimisation algorithms exist for unconstrained problems in $\mathcal{P}$. These upper bounds on the rate of convergence ρ imply that for any problem in $\mathcal{P}$, the iterates will converge to a solution in $\mathcal{O}(\rho^t)$. For clarity's sake, we reformulate these upper bounds to be consistent with the notation of $\mathcal{P}$ and eq. 6. Letting $\kappa = \frac{L_{12}}{\sqrt{\mu_1 \mu_2}}$, Chen and Rockafellar (1997) analyse the forward-backward algorithm, and

find a convergence in $\mathcal{O}\left(\left(\sqrt{1-\left(\frac{\min(\mu_1,\mu_2,\mu_{12})}{\max(L_1,L_2,L_{12})}\right)^2}\right)^t\right)$. Chambolle and Pock (2011) give an algorithm for which the convergence is in $\mathcal{O}\left(\left(\sqrt{1-\frac{2}{\kappa+2}}\right)^t\right)$. Palaniappan and Bach (2016) present an accelerated version of the forward-backward algorithm with variance reduction with convergence in $\mathcal{O}\left(\left(1-\frac{1}{1+2\kappa}\right)^t\right)$. Note that asymptotically, the Chambolle-Pock and accelerated Forward-Backward rates match up to a factor of 2 on κ. Finally, Gidel et al. (2019b) give an upper bound in $\mathcal{O}\left(\left(1-\frac{1}{4L_{12}^2/\mu_{12}^2}\right)^t\right)$ on the convergence of alternating gradient descent with negative momentum for non-singular *bilinear games*, i.e. quadratic games satisfying eq. 8 with $\boldsymbol{S}_1 = \boldsymbol{S}_2 = 0$ and non-singular Jacobian.

A key problem with those rates of convergence is that the tightness of the upper bound is not established. Such information is important since it may indicate that the algorithm can be accelerated. Ideally, one would use the rate of convergence of the hardest problem (i.e. slowest convergence) in the class of problems, which would be a tight upper bound. If one can only find a lower bound on the rate of convergence of the hardest problem, then any upper bound on the entire problem class must be greater than that lower bound to avoid a contradiction. This is because the (upper bound on the) rate of convergence for a class of problems must apply to *any* problem in the class, and hence be greater than any lower bound derived on any particular problem within that class. Usually, it is not possible to find the problem with the slowest convergence, so one may have to guess a hard enough problem. If the lower bound on that problem matches the upper bound for the problem class, then we have established that not only this problem is one of the hardest problems in the class, but also that the upper bound is tight. Therefore, it is important to remember that the goal of lower bounds generally is not to apply to every problem within the class, unlike upper bounds, but to help estimate how much the upper bounds on the whole class can be improved given the presence of hard problems in the class.

Unlike upper bounds, relevant lower bounds for first-order methods on saddle-point problems are scarcer in the literature. Nemirovsky (1992) gives a lower bound in $\mathcal{O}(1/t)$ for a limited number of steps. Ouyang and Xu (2018) also leverage Krylov subspace techniques, and show lower bounds in $\mathcal{O}(1/t)$ in the monotone case and $\mathcal{O}(1/t^2)$ in the strongly monotone case, assuming the number of iterations is less than half the dimension of the parameters. Note that Ouyang and Xu (2018) do not assume smoothness of the objective in $\boldsymbol{y}$. A key issue is that since these bounds are only valid for a limited number of steps, they do not yield bounds that can be compared with the upper bounds previously mentioned. In contrast, the lower bounds presented in this work are valid for any number of steps and are linear, and therefore provide a direct limit to the acceleration of methods achieving linear convergence on two-player games. Additionally, our lower bounds also yield condition numbers that give intuition about the difficulty inherent to a problem, and can be computed in a plug-and-play fashion using either bounds on the spectrum of the full Jacobian, or on the spectra of its blocks.

3.2. Lower bound techniques for convex optimisation with bounded spectrum

In single-objective optimisation, i.e. a 1-player game, the Jacobian in eq. 2 reduces to the Hessian of the objective, denoted $\boldsymbol{H}(\boldsymbol{x})$. In that case, if there exists $\mu, L \in \mathbb{R}^{++}$ such that for all $\boldsymbol{x}$ in the domain considered

$$\mu \preceq \boldsymbol{H}(\boldsymbol{x}) \preceq L$$

the objective is μ-strongly convex and has L-Lipschitz gradients, and the convergence rates (i.e. upper bounds) are known to be linear in the number of iterations for several classes of algorithms (Nemirovsky and Yudin, 1983; Nesterov, 2004). In the context of convex minimisation, various lower bounds have been derived depending on whether the objective is strongly convex and/or has Lipschitz gradients (see (Bubeck et al., 2015) for an overview).

Nesterov's lower bound In particular, Nesterov (2004) gives an information-based complexity bound for μ-strongly convex objectives with L-Lipschitz gradients, by showing that there is a μ-strongly convex example in $\ell_2 \to \mathbb{R}$ with L-Lipschitz gradients for which first-order black box methods, i.e. methods using only past iterates and gradients of past iterates at every update, converge linearly at a rate at least $\rho = 1 - \frac{2}{\sqrt{\kappa}+1}$, where the condition number is given by $\kappa = L/\mu$. The proof relies on the fact that at iteration t, only the t first components of the estimates $\boldsymbol{x}_t$ have been updated from their initial values, where $\boldsymbol{x}_0 = 0$. This is then used to lower bound the distance to the optimum. Since an infinite number of iterations is required to converge in ℓ_2 if the solution $\boldsymbol{x}^*$ has an infinite number of nonzero components, we obtain asymptotic rates. An important caveat is that an infinite-dimensional example does not directly yield a lower bound for finite-dimensional problems.

p-SCLI Arjevani et al. (2016) introduce the p-SCLI framework to provide bounds for a large class of methods used for optimising μ-strongly convex objectives with L-Lipschitz gradients. Roughly speaking, an algorithm is p-SCLI if its update rule on quadratics $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^\top \mathbf{b}$ with $\boldsymbol{A}$ symmetric is a linear combination of the p previous iterates and $\mathbf{b}$, where the coefficients are matrices that depend on

$\boldsymbol{A}$ and are assumed to be simultaneously triangularisable. The spectral properties of the update rule are used to derive lower bounds on the rate of convergence of p-SCLI algorithms. The lower bound on the rate of convergence of p-SCLI methods is given by $\rho = 1 - \frac{2}{\sqrt[p]{\kappa}+1}$ for $\kappa = L/\mu$. This allows us to recover lower bounds for gradient descent ($p = 1$) or Nesterov (1983)'s accelerated gradient descent ($p = 2$) that match the upper bounds. A key advantage is that these bounds are more refined than Nesterov's by introducing the dependence on p (for example, both GD and Nesterov's accelerated gradient descent are black box first-order methods, but this bound is tighter for $p = 1$ methods such as GD), and do not rely on an infinite-dimensional example, but rather on the spectral properties of the methods. Such bounds highlight how lower bounds can suggest potential acceleration: in the example previously given, the addition of momentum to gradient descent turns the method from $p = 1$ to $p = 2$, thereby decreasing the lower bound and allowing for potentially faster convergence. Finally, the p-SCLI framework also yields upper bounds, and the authors also show a general method to accelerate algorithms on quadratics with the hope that the acceleration is relevant to more general classes of objectives, albeit at a cost too prohibitive to be practical.

Interestingly, both techniques produce a tight lower bound from a quadratic objective, indicating that quadratics are asymptotically as hard as any other strongly convex, smooth problem in single-objective optimisation. This motivates the use of quadratic games to derive the lower bounds presented in this paper as we generalise those methods to the multi-objective setting. When there are several players, however, the Jacobian is no longer symmetric, and its spectrum will generally be complex, and hence several of the arguments used in single-objective optimisation fail to apply directly.

4. Parametric Lower Bounds from Nesterov's Domino Argument

In this section, we will only discuss min-max problems. The class $\mathcal{F}$ of counterexamples considered is

$$\min_{\boldsymbol{x}\in\ell_2}\max_{\boldsymbol{y}\in\ell_2} \quad f(\boldsymbol{x},\boldsymbol{y}) = c\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{y} - d_1\boldsymbol{x}^\top \boldsymbol{e_1} + d_2\boldsymbol{y}^\top \boldsymbol{e_1} + \frac{\mu_1}{2}\|\boldsymbol{x}\|^2 - \frac{\mu_2}{2}\|\boldsymbol{y}\|^2 \qquad (\mathcal{F})$$

where $\boldsymbol{M}$ is an infinite-dimensional bidiagonal matrix i.e. $\forall i, M_{ii} = a_0$ and $M_{i,i+1} = a_1$ with all other entries set to 0, such that $ca_0a_1 \neq 0$ and $\mu_1, \mu_2 \in \mathbb{R}^{++}$. Since these problems are in $\mathcal{P}$, the lower bounds of this section are in particular bounds on the optimisation of min-max problems.

Definition 1 (Two-step linear span assumption)**.** *A first-order black box method for 2-player games satisfies the two-step linear span assumption on $\mathcal{F}$ if for problems in $\mathcal{F}$ with Jacobian $\boldsymbol{A}$ (cf eq. 5):*

$$\mathbf{w}_t \in \mathbf{w}_0 + \mathit{Span}\big(\mathbf{w}_0, ..., \mathbf{w}_{t-1}, \boldsymbol{A}\mathbf{w}_0, ..., \boldsymbol{A}\mathbf{w}_{t-1}, \boldsymbol{A}^2\mathbf{w}_0, ..., \boldsymbol{A}^2\mathbf{w}_{t-1}, \mathbf{b}, \boldsymbol{A}\mathbf{b}\big) \qquad (9)$$

Examples of such methods include simultaneous gradient descent, negative momentum and extragradient. One way to design challenging problems for these methods is to construct problems with a dense solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for which only one new component of the iterates may change from its initial value at every iteration (Nesterov, 2004), a phenomenon we will refer to as the *domino argument* (see Appendix B.1 for some intuition, where we show that the argument also applies to cases where diagonal matrices are used as coefficients in the span, and to alternating implementations of any algorithm satisfying the two-step linear span assumption on $\mathcal{F}$ thanks to the properties of bidiagonal Toeplitz matrices).

4.1. A first lower bound for games with block spectral bounds μ_1, μ_2, L_{12}

Proposition 2 (Naive bound)**.** *For any problem class containing quadratic games, there exists a function $f : \ell_2 \times \ell_2 \to \mathbb{R}$ corresponding to a problem in $\mathcal{P}$ with block spectral bounds $\mu_1 = L_1, \mu_2 = L_2, L_{12} \in \mathbb{R}^{++}$ as defined in eq. 6, that has condition number $\kappa = \frac{L_{12}}{\sqrt{\mu_1\mu_2}}$ such that for any number of iterations $t \geq 1$ and any procedure satisfying the two-step linear span assumption (see def. 1), the following lower bound holds:*

$$\|(\boldsymbol{x}_t, \boldsymbol{y}_t) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| \geq \left(1 - \frac{2}{\sqrt{\kappa^2+1}+1}\right)^{t+1} \cdot \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| \qquad (10)$$

We invite the reader to consult Appendix B.2 for the proof. This lower bound on the distance to a solution also yields a lower bound on the minimum number of steps necessary for all subsequent iterates to be within a target distance — typically referred to as *iteration complexity*. We may interpret the proposition as being a bound for 2-player games with block spectral bounds μ_1, μ_2, and L_{12} as the bound provided holds for a problem sharing the same block spectral bounds. Similarly, we also get a bound on problems with block spectral bounds L_1, L_2 and L_{12} by replacing μ_i by L_i appropriately in κ.

We appear to obtain the same condition number as in the upper bound literature. If we assume this bound and condition number to be representative of a finite-dimensional bound as was the case in convex optimisation, we easily see an apparent contradiction from the upper bound on the rate of convergence of alternating gradient descent with negative momentum for bilinear games given by (Gidel et al.,

2019b). Indeed, if we let $\mu_1, \mu_2 \to 0$, the rate of convergence in Prop. 2 goes to 1, whereas the upper bound of negative momentum is not affected and may indicate fast convergence. This illustrates how the condition number of the upper bounds is not general enough to be representative of inherent difficulty: it can be shown that for the problem used in the proof of the proposition, $\mu_{12} = 0$. As such, it is not surprising that the bound failed to hold against the upper bound of negative momentum on bilinear games; they were not comparable as by definition bilinear games have $\mu_{12} > 0$. This shows that μ_{12} encodes critical information that this condition number was not able to capture. Nevertheless, an important point is that the bound itself is correct and represents a problem with slow convergence; it just fails to yield a condition number that accurately captures difficulty as μ_{12} does not appear.

However, by refining our proof technique, we can derive a bound which avoids this issue, and yields tighter bounds for games for which we know $\mu_{12}, L_{12}, \mu_1, \mu_2$.

4.2. Improved lower bound for games with block spectral bounds $\mu_1, \mu_2, \mu_{12}, L_{12}$

Theorem 3. *For any problem class containing quadratic games, there exists a function $f : \ell_2 \times \ell_2 \to \mathbb{R}$ corresponding to a problem in $\mathcal{P}$ with block spectral bounds $\mu_1 = L_1, \mu_2 = L_2, L_{12} \in \mathbb{R}^{++}, \mu_{12} \in \mathbb{R}^+$ as defined in eq. 6, that has condition number $\kappa = \sqrt{\frac{L_{12}^2+\mu_1\mu_2}{\mu_{12}^2+\mu_1\mu_2}}$, such that for any number of iterations $t \geq 1$ and any procedure satisfying the two-step linear span assumption, the following lower bound holds:*

$$\|(\boldsymbol{x}_t, \boldsymbol{y}_t) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| \geq \left(1 - \frac{2}{\kappa + 1}\right)^{t+1} \cdot \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| \quad (11)$$

The same result holds for any problem class containing bilinear games, if one sets $\mu_1 = L_1 = \mu_2 = L_2 = 0$.

Corollary 4 (Iteration complexity bound)**.** *For the same problem classes and under the same assumptions as Theorem 3, the minimal number of steps t required to reach a target distance ϵ from the solution, that is $\|(\boldsymbol{x}_t, \boldsymbol{y}_t) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| < \epsilon$, is given by*

$$t \geq \frac{\kappa - 1}{2} \log\left(\frac{\|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\|}{\epsilon}\right) - 1 \quad (12)$$

This generalises Prop. 2. The proof can be found in Appendix B.3, where we also show how we used spectral properties of Toeplitz matrices in Banach algebras to create the hard problems yielding the bound. The proof of the iteration complexity bound can be found in Appendix D. As with the lower bounds, note that the iteration complexity bound does not necessarily hold for every problem in the class; the idea is that to reach a target error ϵ we know that there is at least one problem that requires at least the number of steps indicated in the bound, and therefore that to optimise over a problem class that satisfies the assumptions, we will need in general at least the number of steps given in the bound, to account for the hard problems.

It is important to emphasize that as is the case with Nesterov's argument for single-objective optimisation, this bound is still based on an infinite-dimensional problem, and that upper bounds generally are proven for finite-dimensional settings. We may hope that this bound also holds in finite dimension, since we are not aware of upper bounds contradicting it, and were not able to generate finite-dimensional 2-player games for which the bound did not hold empirically. The condition number appearing in Thm. 3 is more expressive than the one found in the upper bound literature $\kappa = L_{12}/\sqrt{\mu_1\mu_2}$, and is lower bounded by 1, instead of 0. A limitation, however, is that this κ is not able to dissociate $L_1 \neq \mu_1, L_2 \neq \mu_2$, which is a problem in terms of expressivity of the condition number. It may also threaten the tightness of the lower bound since intuition from convex optimisation would suggest that objectives with matching lower and upper bounds on the spectra are easier to optimise. This stems from the fact that the closed form solution for problems in $\mathcal{P}$ when $\boldsymbol{S}_i$ is non-scalar, which we would need for all block spectral bounds to appear in κ, is complicated and the associated condition number is impractical. Therefore, we leave the matter of deriving a practical bound based on the domino argument involving all μ_i and L_i as future work.

Interestingly, the rate takes the same form as in strongly convex smooth optimisation, suggesting that for general n-player games, we may still get a lower bound of the form $\rho \geq 1 - \frac{2}{\kappa+1}$ for some generalised condition number κ. This intuition will be highlighted in the next section, by deriving lower bounds from the spectral properties of the update operators of a large class of optimisation methods for n-player games. The results we are about to introduce will also address the matter of $L_i \neq \mu_i$, and will be based on finite-dimensional problems.

5. p-SCLI-n for n-player Games

5.1. Definitions and examples

Let $\mathcal{Q}^{d_1,\ldots,d_n}$ denote the set of n-player quadratic games, i.e. games comprised of n quadratic objectives $l_i : \mathbb{R}^{d_i} \to \mathbb{R}$, and $f_{\boldsymbol{A},\mathbf{b}}(\mathbf{w}) \in \mathcal{Q}^{d_1,\ldots,d_n}$ be a game with vector field $\boldsymbol{A}\mathbf{w} + \mathbf{b}$ as indicated in eq. 5. The following definition is a direct generalisation of the definition of p-SCLI algorithms given by Arjevani et al. (2016) to n-player games.

Definition 5 (p-SCLI-n optimisation algorithms for n-player games)**.** *Let $\mathcal{A}$ be an optimisation algorithm for*

n-player quadratic games. Then $\mathcal{A}$ is a p-stationary canonical linear iterative method for n-player games (p-SCLI-n) if there exist functions $C_0, ..., C_{p-1}, N$ from $\mathbb{R}^{d\times d}$ to $\mathbb{R}^{d\times d}$-valued random variables, such that the following conditions are satisfied for all $f_{\boldsymbol{A},\mathbf{b}}(\mathbf{w}) \in \mathcal{Q}^{d_1,...,d_n}$:

1. *Given an initialisation $\mathbf{w}^0, \dots, \mathbf{w}^{p-1} \in \mathbb{R}^d$, the update rule at iteration $t \geq p$ is given by*

$$\mathbf{w}^t = \sum_{i=0}^{p-1} C_i(\boldsymbol{A})\mathbf{w}^{t-p+i} + N(\boldsymbol{A})\mathbf{b} \tag{13}$$

2. *$C_0(\boldsymbol{A}), ..., C_{p-1}(\boldsymbol{A}), N(\boldsymbol{A})$ are independent from previous iterations*

3. *$\mathbb{E}C_i(\boldsymbol{A})$ are finite and simultaneously triangularisable*

We will refer to the C_i as the coefficient matrices and N as the inversion matrix.

An important fact is that if $n = 1$, this definition becomes the same as the one given by Arjevani et al. (2016). A key difference, however, is that the Jacobian $\boldsymbol{A}$ will generally not be symmetric for $n > 1$; only the blocks $\boldsymbol{M}_{ii}$ will be, and hence we may not assume the spectrum $\sigma(\boldsymbol{A})$ to be positive since it will generally be complex. Fortunately, several results from Arjevani et al. (2016) hold nevertheless, as discussed in Appendix C.1. Before introducing the results, let us give examples of algorithms used to optimise games that are p-SCLI-n, as evidenced by their update rule on quadratic games.

Simultaneous Gradient Descent (GD) The update rule is given by $\mathbf{w}_i^t = \mathbf{w}_i^{t-1} - \eta_i \nabla_{\mathbf{w}_i} l_i(\mathbf{w}^{t-1})$, which can be rewritten with $\boldsymbol{\eta} = \text{Diag}\,(\eta_1, ..., \eta_n)$ as:

$$\begin{aligned}\mathbf{w}^t &= \mathbf{w}^{t-1} - \boldsymbol{\eta}\left(\boldsymbol{A}\mathbf{w}^{t-1} + \mathbf{b}\right) \\ &= (\boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A})\mathbf{w}^{t-1} - \boldsymbol{\eta}\mathbf{b}\end{aligned} \tag{14}$$

This shows that simultaneous gradient descent is a 1-SCLI-n algorithm.

Simultaneous Momentum GD The update rule is $\mathbf{w}_i^t = \mathbf{w}_i^{t-1} - \eta_i \nabla_{\mathbf{w}_i} l_i(\mathbf{w}^{t-1}) + \beta_i(\mathbf{w}_i^{t-1} - \mathbf{w}_i^{t-2})$ which can rewritten with $\boldsymbol{\beta} = \text{Diag}\,(\beta_1, ..., \beta_n)$ and $\boldsymbol{\eta}$ as before:

$$\begin{aligned}\mathbf{w}^t &= \mathbf{w}^{t-1} - \boldsymbol{\eta}\left(\boldsymbol{A}\mathbf{w}^{t-1} + \mathbf{b}\right) + \boldsymbol{\beta}(\mathbf{w}^{t-1} - \mathbf{w}^{t-2}) \\ &= (\boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A} + \boldsymbol{\beta})\mathbf{w}^{t-1} - \boldsymbol{\beta}\mathbf{w}^{t-2} - \boldsymbol{\eta}\mathbf{b}\end{aligned} \tag{15}$$

Therefore, simultaneous gradient descent with momentum is a 2-SCLI-n, if we assume $\boldsymbol{\beta}$ to be scalar (since we need the coefficient matrices $C_i(\boldsymbol{A})$ to be simultaneously triangularisable).

Extragradient (Korpelevich, 1976) The update rule is $\mathbf{w}_i^t = \mathbf{w}_i^{t-1} - \eta_i \nabla_{\mathbf{w}_i} l_i(\mathbf{w}^{t-1} - \boldsymbol{\eta}\boldsymbol{v}(\mathbf{w}^{t-1}))$, which can be rewritten as:

$$\begin{aligned}\mathbf{w}^t &= \mathbf{w}^{t-1} - \boldsymbol{\eta}(\boldsymbol{A}(\mathbf{w}^{t-1} - \boldsymbol{\eta}(\boldsymbol{A}\mathbf{w}^{t-1} + \mathbf{b})) + \mathbf{b}) \\ &= (\boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A} + (\boldsymbol{\eta}\boldsymbol{A})^2)\mathbf{w}^{t-1} - (\boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A})\boldsymbol{\eta}\mathbf{b}\end{aligned} \tag{16}$$

This shows that extragradient is a 1-SCLI-n.

Simultaneous Stochastic Gradient Descent The reasoning is the same as the one presented by Arjevani et al. (2016): we approximate $\nabla f_{\boldsymbol{A},\mathbf{b}}(\mathbf{w}) = \boldsymbol{A}\mathbf{w} + \mathbf{b}$ with stochastic gradients $\boldsymbol{G}_\omega(\mathbf{w})$ and denote the error by $\boldsymbol{e}_\omega(\mathbf{w}) = \boldsymbol{G}_\omega(\mathbf{w}) - (\boldsymbol{A}\mathbf{w} + \mathbf{b})$. Then the update rule for fixed $\boldsymbol{\eta}$ is given by

$$\begin{aligned}\mathbf{w}^t &= \mathbf{w}^{t-1} - \boldsymbol{\eta}\boldsymbol{G}_{\omega_{t-1}}(\mathbf{w}^{t-1}) \\ &= (\boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A})\mathbf{w}^{t-1} - \boldsymbol{\eta}\mathbf{b} - \boldsymbol{\eta}\boldsymbol{e}_{\omega_{t-1}}(\mathbf{w}^{t-1})\end{aligned} \tag{17}$$

Under certain assumptions, e.g. if $\boldsymbol{e}_\omega(\mathbf{w}) = \boldsymbol{A}_\omega\mathbf{w} + \boldsymbol{N}_\omega\mathbf{b}$ and $\mathbb{E}\boldsymbol{A}_\omega = \mathbb{E}\boldsymbol{N}_\omega = 0$, then the update rule becomes

$$\mathbf{w}^t = (\boldsymbol{I} - \boldsymbol{\eta}(\boldsymbol{A} + \boldsymbol{A}_{\omega_{t-1}}))\mathbf{w}^{t-1} - \boldsymbol{\eta}(\boldsymbol{I} + \boldsymbol{N}_{\omega_{t-1}})\mathbf{b} \tag{18}$$

and we get a 1-SCLI-n.

One last definition is required before we introduce the p-SCLI lower bounds. Our definition generalises that of Arjevani et al. (2016).

Definition 6 (Consistency of p-SCLI-n optimisation algorithms)**.** *Let $\mathcal{Q}_{\boldsymbol{A}}^{d_1,...,d_n} \subseteq \mathcal{Q}^{d_1,...,d_n}$ denote the set of quadratic n-player games with non-singular Jacobian $\boldsymbol{A}$ (see eq. 5). Then $\mathcal{A}$ is consistent with respect to $\boldsymbol{A}$ if for any game $f_{\boldsymbol{A},\mathbf{b}} \in \mathcal{Q}_{\boldsymbol{A}}^{d_1,...,d_n}$ and any initialisation, $\mathcal{A}$ converges to a stationary point of $f_{\boldsymbol{A},\mathbf{b}}$ or equivalently if the sequence of iterates $(\mathbf{w}^t)$ (see eq. 13) satisfies*

$$\mathbf{w}^t \to -\boldsymbol{A}^{-1}\mathbf{b} \tag{19}$$

Equivalently, as Arjevani et al. (2016) argue in their section 3.1, consistency with respect to some invertible Jacobian $\boldsymbol{A}$ is equivalent to having $\mathcal{A}$ converge on $f_{\boldsymbol{A},\mathbf{b}}$ and

$$\sum_{i=0}^{p-1} \mathbb{E}C_i(A) = I_d + \mathbb{E}N(\boldsymbol{A})\boldsymbol{A} \tag{20}$$

Note that all three examples of optimisation algorithms discussed in this subsection satisfy eq. 20.

5.2. Parametric lower bound for p-SCLI-n with scalar inversion matrix

We are now ready to introduce the lower bound for p-SCLI-n methods with scalar inversion matrix.

Proposition 7. *Let $\mathcal{A}$ be a p-SCLI-n algorithm with scalar inversion matrix for optimising games over $\mathbb{R}^{d_1} \times ... \times \mathbb{R}^{d_n}$.*

Then for quadratics $f_{\mathbf{A},\mathbf{b}} \in \mathcal{Q}^{d_1,\dots,d_n}$, if $\mathcal{A}$ is consistent with respect to $\boldsymbol{A}$ and if $0 \notin \sigma(\boldsymbol{A})$, we have the following lower bound on the (linear) rate of convergence ρ:

$$\rho \geq \frac{\sqrt[p]{\kappa}-1}{\sqrt[p]{\kappa}+1} = 1 - \frac{2}{\sqrt[p]{\kappa}+1} \tag{21}$$

where the condition number κ is defined as $\kappa \triangleq \frac{\max|\sigma(\boldsymbol{A})|}{\min|\sigma(\boldsymbol{A})|}$ where $\sigma(\boldsymbol{A})$ is the spectrum of $\boldsymbol{A}$.

While this lower bound is valid for p-SCLI-n methods on *any single quadratic game where the assumptions apply*, it also gives us a more general result for problem classes containing quadratics as an immediate corollary, similarly to the lower bounds of the previous section.

Theorem 8. *For any problem class containing quadratic games, there exists a game such that the lower bound given in eq. 21 holds for a p-SCLI-n method.*

Corollary 9 (Iteration complexity bound)**.** *For the same problem classes and under the same assumptions as Theorem 8, the minimal number of steps t to satisfy $\max_{i=0,\dots,p-1} \left\|\mathbb{E}\mathbf{w}^{t+i} - \mathbb{E}\mathbf{w}^*\right\| < \epsilon$ is given by*

$$t \geq \left(\frac{\sqrt[p]{\kappa}-1}{2}\right) \log\left(\frac{C}{\epsilon}\right) \tag{22}$$

for some strictly positive constant C.

See Appendix C.1 for the proof of Prop. 7, and Appendix D for the iteration complexity bound's proof. Interestingly, by setting $n = 1$, this bound captures the 1-player case for μ-strongly convex objectives with L-Lipschitz gradients, where $\kappa = \frac{\max\sigma(\boldsymbol{A})}{\min\sigma(\boldsymbol{A})} = L/\mu$, verifying the intuition discussed at the end of the previous section, and showing that this technique yields a tight lower bound for $n = 1$. Moreover, this form is valid for n-player games (and min-max problems) in finite dimension, and κ arises naturally from the spectral properties of the update rules of the p-SCLI-n methods and is lower bounded by 1. Additionally, our bounds are valid for some stochastic methods. In single-objective optimisation (i.e. $n = 1$), while linear rates are not achievable for general stochastic problems, for which the worst-case bounds are sublinear, under certain conditions linear rates are possible (Loizou and Richtárik, 2017). Conditions of this type can be satisfied in over-parameterised neural networks (Vaswani et al., 2019). Hence, our linear lower bounds may be useful even for stochastic problems.

However, while the moduli in the $n > 1$ case allow us to handle complex spectra and matches the classical definition of condition number from linear algebra, several analyses have shown that not only the modulus, but also the relative size of the real and imaginary parts of elements of the spectrum matter (Mescheder et al., 2017; Gidel et al., 2019b). Such an analysis may yield more expressive bounds, but is out of the scope of this work. We will nevertheless give a more explicit form of the bound for 2-player games for which $d_1 = d_2$ that will make the μ_i and L_i appear.

Some explicit bounds for p-SCLI-2 with $d_1 = d_2$ Prop. 7 may be used to derive lower bounds for 2-player games for which $d_1 = d_2$. These bounds depend on the value of the μ_i and L_i defined as in eq. 6. Namely, let

$$\begin{aligned}\Delta_\mu &= (\mu_1+\mu_2)^2 - 4(\mu_1\mu_2+\mu_{12}^2) \\ &= (\mu_1-\mu_2)^2 - 4\mu_{12}^2 \qquad (23)\\ \Delta_L &= (L_1+L_2)^2 - 4(L_1L_2+L_{12}^2) \\ &= (L_1-L_2)^2 - 4L_{12}^2 \qquad (24)\end{aligned}$$

Table 1 gives lower bounds on the condition number that may then be plugged into eq. 21 to get lower bounds on two-players games corresponding to min-max problems (and are therefore lower bounds for general 2-player games).

Table 1. Lower Bounds on the Condition Number

	$\Delta_\mu < 0$	$\Delta_\mu \geq 0$
$\Delta_L < 0$	$\kappa = \sqrt{\frac{L_1L_2+L_{12}^2}{\mu_1\mu_2+\mu_{12}^2}}$	$\kappa \geq 2\frac{\sqrt{L_1L_2+L_{12}^2}}{\mu_1+\mu_2-\sqrt{\Delta_\mu}}$
$\Delta_L \geq 0$	$\kappa \geq \frac{1}{2}\frac{L_1+L_2+\sqrt{\Delta_L}}{\sqrt{\mu_1\mu_2+\mu_{12}^2}}$	$\kappa \geq \frac{L_1+L_2+\sqrt{\Delta_L}}{\mu_1+\mu_2-\sqrt{\Delta_\mu}}$

See Appendix C.2 for the counterexample in $\mathcal{P}$ leading to these bounds. This result resolves the issues raised in the discussion of the domino bounds as it uses all of the μ_i and L_i, and is proven from finite-dimensional problems. In fact, if one sets $\mu_1 = L_1$ and $\mu_2 = L_2$ such that μ_1 and μ_2 are small (in particular, smaller than μ_{12}) and L_{12} is large (in particular, larger than L_1, L_2), table 1 yields $\kappa = \sqrt{\frac{\mu_1\mu_2+L_{12}^2}{\mu_1\mu_2+\mu_{12}^2}}$, which coincides with the κ from Thm. 3. More generally, for $p = 1$, if both Δ_L and Δ_μ are negative, we get a tighter bound from the p-SCLI-2 formalism for 1-SCLI-2 methods that also satisfy the two-step linear span assumption than from Thm. 3. Finally, it provides the same plug-and-play convenience as p-SCLI to derive bounds for a large class of algorithms that may not satisfy the first-order black box assumption. On the other hand, the bounds may not be tight for $p \geq 3$, as it was the case for single-objective p-SCLI (Arjevani et al., 2016).

An interesting case is $p = 2$: for 2-SCLI-2 methods that satisfy the two-step linear span assumption such as negative momentum, the rate stemming from the p-SCLI-2 analysis appears to be smaller than the rate from the improved domino bound. This may be because the proof techniques used in our generalisation of p-SCLI yield bounds that can be improved for $p > 1$. In particular, a key difference between Thm. 3 and 8 is that as explained in the background section, lower bounds generally need not apply to every

single problem within a class. However, due to the proof technique used, the bound given in Prop. 7 has to apply to every single quadratic game where the assumptions hold. Because some games might be inherently easier than others, such games may bottleneck the bound in Prop. 7 and Thm. 8, and introduce looseness. Indeed, first, the number of players does not appear in Prop. 7's proof, meaning that the proof yielded a bound that should hold for $n = 1$ which might inherently have faster convergence than $n > 1$ due to having to optimise only one objective over one set of parameters. Second, in the proof, because we do not impose a structure for the quadratic games, the bound has to hold for purely cooperative games (that is, the objective of each player does not depend on other players' parameters), in which case the situation is analogous to n single-objective optimisation problems, where we may expect the convergence to be faster than in games with interactions between players. As such, it is worth considering whether fixing the number of players and the structure of the games earlier in the proof could improve the bound.

6. Conclusion

In this work, we provide linear lower bounds and condition numbers for any problem class containing quadratic or bilinear games. We give a lower bound on the rate of convergence of first-order black box methods for 2-player games (which directly applies to min-max and saddle point problems) satisfying the two-step linear span assumption by generalising Nesterov's lower bound for the optimisation of strongly convex, smooth convex objectives to 2-player games (R.Q. 1) and constructing a novel class of hard problems using spectral properties of a class of operators in Banach algebras. Moreover, we generalise the framework of p-SCLI, which requires symmetricity of the Hessian in single-objective optimisation, to provide a bound for a large class of optimisers for n-player games by extending the results of p-SCLI to quadratic games with non-symmetric Jacobian, which for $n = 1$ recovers (Arjevani et al., 2016)'s tight bounds. We then give explicit bounds for 2-player games, which apply to min-max and saddle point problems (R.Q. 2). Finally, we derived formulations for the condition number that matched (in the case of the first domino bound), or were more general (in the case of the improved domino bound, and p-SCLI-n and p-SCLI-2 bounds) than the existing ones in the upper bound literature (R.Q. 3). As in the single-objective case, our bounds and condition numbers suggest that optimisers may converge faster on games for which the eigenvalues are at a similar, remote distance from the origin (e.g. on a circle) than on games for which some eigenvalues are close to and others are far from 0.

Following the initial release of this work, several other authors have built upon the bounds presented in this paper. The bound from Theorem 3 is tight on the class of smooth strongly-convex-strongly-concave games, as it is matched by the upper bound presented by Fallah et al. (2020), and up to logarithmic factors, by the upper bound of Lin et al. (2020). Additionally, this same lower bound was also shown to be tight on the class of bilinear games with non-singular Jacobian, which are merely convex-concave, by Prop. 4 of Azizian et al. (2020). As a corollary, our results establish the optimality of those methods on the aforementioned classes of problems.

However, several directions remain to be explored. For example, we raised the question of whether the p-SCLI-n bound could be tightened for $p > 1$ by improving the proof technique, especially given that we are not aware of faster rates of convergence in the literature than those of $p = 1$ or those of the improved domino bound. Moreover, we would like to present a more exhaustive overview of the extension of p-SCLI, and discuss the resulting upper and lower bounds for various commonly used algorithms. In particular, we would like to extend our lower bounds to p-SCLI-n with diagonal inversion matrices, as Arjevani et al. (2016) did in the p-SCLI framework, and provide bounds in the 2-player case when $d_1 \neq d_2$. Furthermore, we believe tighter bounds may be derived, for example by adding constraints on the C_i or by looking not only at the modulus of the eigenvalues but also at their arguments, as done by Gidel et al. (2019b), since we know that the relative size of the imaginary part and real part (even at fixed modulus) affects the dynamics in games (Mescheder et al., 2017). Finally, it would be important to understand when and how linear convergence is possible in non strongly-convex-strongly-concave settings. We plan on exploring several of these directions in future work.

Acknowledgements

The authors would like to thank Damien Scieur for useful discussions and feedback. This work was partially supported by the FRQNT new researcher program (2019-NC-257943), the NSERC Discovery grant (RGPIN-2019-06512), a startup grant by IVADO, a Microsoft Research collaborative grant and a Canada CIFAR AI chair. Gauthier Gidel was partially supported by a Borealis AI Graduate Fellowship.

References

Yossi Arjevani, Shai Shalev-Shwartz, and Ohad Shamir. On lower and upper bounds in smooth and strongly convex optimization. *The Journal of Machine Learning Research*, 17(1):4303–4353, 2016.

Waïss Azizian, Damien Scieur, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. Accelerating smooth games by manipulating spectral shapes. *arXiv preprint*

arXiv:2001.00602, 2020.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pages 363–372, 2018.

Michele Benzi and Gene H Golub. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 2004.

Gilles Brassard and Paul Bratley. Fundamentals of algorithmics. 1996.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

George HG Chen and R Tyrrell Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, 1997.

Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Accelerated schemes for a class of variational inequalities. *Mathematical Programming*, 165(1):113–149, 2017.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018.

Ronald G Douglas. *Banach algebra techniques in operator theory*, volume 179. Springer Science & Business Media, 2012.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1049–1058. JMLR. org, 2017.

Alireza Fallah, Asuman Ozdaglar, and Sarath Pattathil. An optimal multistage stochastic gradient method for minimax problems. *arXiv preprint arXiv:2002.05683*, 2020.

Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *International Conference on Learning Representations*, 2019a.

Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, pages 1802–1811, 2019b.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Patrick T Harker and Jong-Shi Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical programming*, 48(1-3):161–220, 1990.

Seung-Jean Kim and Stephen Boyd. A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM Journal on Optimization*, 19(3):1344–1367, 2008.

GM Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Tianyi Lin, Chi Jin, Michael Jordan, et al. Near-optimal algorithms for minimax optimization. *arXiv preprint arXiv:2002.02417*, 2020.

Nicolas Loizou and Peter Richtárik. Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. In *Advances in Neural Information Processing Systems*, pages 1825–1835, 2017.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

AS Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2004.

Yurii E Nesterov. A method for solving the convex programming problem with convergence rate O (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *arXiv preprint arXiv:1808.02901*, 2018.

Balamurugan Palaniappan and Francis Bach. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, pages 1416–1424, 2016.

Lewis Fry Richardson. Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470): 307–357, 1911.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, pages 1195–1204, 2019.

Hongxing Wang, Chaoqun Weng, and Junsong Yuan. Multi-feature spectral clustering with minimax optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4113, 2014.

Fuzhen Zhang, editor. *The Schur Complement and Its Applications*, volume 4 of *Numerical Methods and Algorithms*. Springer-Verlag, New York, 2005. ISBN 978-0-387-24271-2.

A. n-player quadratic games

While our work focuses mostly on 2-player games, one of the main results, Prop. 7, is independent of the number of players, and is proven for general n. Therefore, a short discussion of the form of quadratic n-player games is provided below.

For an n-player game, the general form of a quadratic is given by

$$l_i(\mathbf{w}) = \sum_{j=1}^{n}\sum_{k=1}^{n} \mathbf{w}_j^\top \boldsymbol{M}_{ijk}\mathbf{w}_k + \sum_{j=1}^{n} \mathbf{w}_j^\top \mathbf{b}_{ij} + c_i\,. \tag{25}$$

Because the dynamics depend only on the $\nabla_{\mathbf{w}_i} l_i(\mathbf{w})$, we will get equivalent dynamics by pruning the terms that do not depend on $\mathbf{w}_i$ and working directly with the simpler objectives

$$\begin{aligned} l_i(\mathbf{w}) &= \frac{1}{2}\mathbf{w}_i^\top \boldsymbol{M}_{ii}\mathbf{w}_i + \sum_{j\neq i}^{n} \mathbf{w}_i^\top \boldsymbol{M}_{iij}\mathbf{w}_j + \sum_{j\neq i}^{n} \mathbf{w}_j^\top \boldsymbol{M}_{iji}\mathbf{w}_i + \mathbf{w}_i^\top \mathbf{b}_{ii} \\ &= \frac{1}{2}\mathbf{w}_i^\top \boldsymbol{M}_{ii}\mathbf{w}_i + \sum_{j\neq i}^{n} \mathbf{w}_i^\top \boldsymbol{M}_{ij}\mathbf{w}_j + \mathbf{w}_i^\top \mathbf{b}_i \end{aligned} \tag{26}$$

where we have let $\boldsymbol{M}_{ij} \triangleq \boldsymbol{M}_{iij} + \boldsymbol{M}_{iji}^\top, \mathbf{b}_i \triangleq \mathbf{b}_{ii}$, $1 \leq i, j \leq n$. Note that we may assume the $\boldsymbol{M}_{ii}$ to be symmetric, since in general $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = \frac{1}{2}\boldsymbol{x}^\top(\boldsymbol{A} + \boldsymbol{A}^\top)\boldsymbol{x}$. Thus, we can write:

$$\nabla_{\mathbf{w}_i} l_i(\mathbf{w}) = \begin{pmatrix}\boldsymbol{M}_{i1} & \dots & \boldsymbol{M}_{in}\end{pmatrix}\mathbf{w} + \mathbf{b}_i \tag{27}$$

which yields the following equation for the vector field:

$$\boldsymbol{v}(\mathbf{w}) = \boldsymbol{A}\mathbf{w} + \mathbf{b}, \qquad \boldsymbol{A} \triangleq \begin{pmatrix} \boldsymbol{S}_1 & \dots & \boldsymbol{M}_{1n} \\ \vdots & & \vdots \\ \boldsymbol{M}_{n1} & \dots & \boldsymbol{S}_n \end{pmatrix}, \quad \mathbf{b} \triangleq \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \tag{28}$$

where $\boldsymbol{A}$ is the Jacobian of $\boldsymbol{v}$ and where we let $\boldsymbol{S}_i \triangleq \boldsymbol{M}_{ii}$.

B. Proofs of Nesterov's bounds for games

The proofs in this section are based on min-max problems for a class of functions[3] $f : \ell_2 \times \ell_2 \to \mathbb{R}$ such that

$$f(\boldsymbol{x}, \boldsymbol{y}) = c\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{y} - d_1\boldsymbol{x}^\top \boldsymbol{e_1} + d_2\boldsymbol{y}^\top \boldsymbol{e_1} + \frac{\mu_1}{2}\|\boldsymbol{x}\|^2 - \frac{\mu_2}{2}\|\boldsymbol{y}\|^2 \tag{29}$$

where $\boldsymbol{e_1}$ is a vector with a 1 in the first entry and 0 elsewhere, $c, d_1, d_2 \in \mathbb{R}$, and $\mu_1, \mu_2 \in \mathbb{R}^{++}$, with $\boldsymbol{M}$ upper bidiagonal matrix such that

$$\boldsymbol{M} = \begin{bmatrix} a_0 & a_1 & 0 & 0 & \dots \\ 0 & a_0 & a_1 & 0 & \dots \\ 0 & 0 & a_0 & a_1 & \dots \\ \vdots & & & \ddots & \ddots \end{bmatrix} \tag{30}$$

where $a_0, a_1 \neq 0$.

As Nesterov (2004), we shall assume that $\boldsymbol{x}_0, \boldsymbol{y}_0$ are initialised at 0, as otherwise we may work with $\boldsymbol{x} - \boldsymbol{x}_0$ and $\boldsymbol{y} - \boldsymbol{y}_0$ in the counterexample and perform the change of variable $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{x}_0$, $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{y}_0$ (which would give us zero-initialisation) and switch back at the end of the analysis.

B.1. On the domino argument

More about the domino argument can be found in Nesterov (2004); here, we shall give the intuition as to why it works. Let us introduce the ingredients of the update rule under our assumptions.

$$\boldsymbol{A} = \begin{pmatrix} \mu_1 & \boldsymbol{M} \\ -\boldsymbol{M}^\top & \mu_2 \end{pmatrix} \qquad \boldsymbol{A}^2 = \begin{pmatrix} \mu_1^2 - \boldsymbol{M}\boldsymbol{M}^\top & (\mu_1+\mu_2)\boldsymbol{M} \\ -(\mu_1+\mu_2)\boldsymbol{M}^\top & \mu_2^2 - \boldsymbol{M}^\top\boldsymbol{M} \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} -d_1\boldsymbol{e}_1 \\ d_2\boldsymbol{e}_2 \end{pmatrix} \tag{31}$$

$$\boldsymbol{A}\mathbf{b} = \begin{pmatrix} -\mu_1 d_1 \boldsymbol{e}_1 + d_2\boldsymbol{M}\boldsymbol{e}_1 \\ d_1\boldsymbol{M}^\top\boldsymbol{e}_1 + \mu_2 d_2 \boldsymbol{e}_1 \end{pmatrix} \qquad \boldsymbol{M}\boldsymbol{e}_1 = \begin{pmatrix} a_0 & 0 & \dots \end{pmatrix}^\top \qquad \boldsymbol{M}^\top\boldsymbol{e}_1 = \begin{pmatrix} a_0 & a_1 & 0 & \dots \end{pmatrix}^\top \tag{32}$$

$$\boldsymbol{A}\mathbf{w} = \begin{pmatrix} \mu_1\boldsymbol{x} + \boldsymbol{M}\boldsymbol{y} \\ -\boldsymbol{M}^\top\boldsymbol{x} + \mu_2\boldsymbol{y} \end{pmatrix} \qquad \boldsymbol{A}^2\mathbf{w} = \begin{pmatrix} (\mu_1^2 - \boldsymbol{M}\boldsymbol{M}^\top)\boldsymbol{x} + (\mu_1+\mu_2)\boldsymbol{M}\boldsymbol{y} \\ -(\mu_1+\mu_2)\boldsymbol{M}^\top\boldsymbol{x} + (\mu_2^2 - \boldsymbol{M}^\top\boldsymbol{M})\boldsymbol{y} \end{pmatrix} \tag{33}$$

B.1.1. One-step linear span assumption

If the algorithm follows the one-step assumption (e.g. gradient descent), which we define as

$$\mathbf{w}_t \in \mathbf{w}_0 + \text{Span}\big(\mathbf{w}_0, ..., \mathbf{w}_{t-1}, \boldsymbol{A}\mathbf{w}_0, ..., \boldsymbol{A}\mathbf{w}_{t-1}, \mathbf{b}\big) \tag{34}$$

note that the part of $\mathbf{b}$ contributing to the update rules of both $\boldsymbol{x}$ and $\boldsymbol{y}$ will have be a vector with a single non-zero entry as its first entry, i.e. $(*, 0, ...)$. Therefore,

$$\boldsymbol{x}_t \in \text{Span}\big((*, 0, ...), \boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}, \boldsymbol{M}\boldsymbol{y}_0, ..., \boldsymbol{M}\boldsymbol{y}_{t-1}\big) \tag{35}$$

$$\boldsymbol{y}_t \in \text{Span}\big((*, 0, ...), \boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}, \boldsymbol{M}^\top\boldsymbol{x}_0, ..., \boldsymbol{M}^\top\boldsymbol{x}_{t-1}\big) \tag{36}$$

Since $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$ but $(\boldsymbol{x}^*, \boldsymbol{y}^*) \neq 0$, we want to see, at every iteration t, how many components of $\boldsymbol{x}_t, \boldsymbol{y}_t$ have been *initialised*, i.e. received information from (which components of) past iterations and therefore could have changed from their initial values of zero. The dependence of the components of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ on past iterates, based on the one-step linear span assumption, is summarised below:

Comp. $i = 1$ of $\boldsymbol{x}_t$: comp. 1 from const. vector, i from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$, and $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$
Comp. $i = 1$ of $\boldsymbol{y}_t$: comp. 1 from const. vector, i from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$, and i from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$
Comp. $i \geq 2$ of $\boldsymbol{x}_t$: comp. i from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$, and $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$
Comp. $i \geq 2$ of $\boldsymbol{y}_t$: comp. i from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$, and $i-1, i$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$

[3] As evoked in the discussion of the improved bound, for these functions, a limitation is that $\mu_1 = L_1$ and $\mu_2 = L_2$, but we were not able to find counterexamples with $L_i \neq \mu_i$ for which the bound had simple enough closed form, even when choosing terms of the form $\boldsymbol{x}^\top\boldsymbol{M}_{\boldsymbol{x}}\boldsymbol{x}, \boldsymbol{y}^\top\boldsymbol{M}_{\boldsymbol{y}}\boldsymbol{y}$ with $\boldsymbol{M}_{\boldsymbol{x}}, \boldsymbol{M}_{\boldsymbol{y}}$ bidiagonal or tridiagonal.

Therefore, if $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$, it is clear that the only terms in the update rule that may initialise any new component of $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ are the constant vectors. Thus, in $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ only the first component will be initialised if we are using simultaneous first-order black box methods satisfying the one-step linear span assumption, and additionally the second component of $\boldsymbol{y}_1$ if we extended the definition of the linear span assumption to use $\boldsymbol{x}_t$ instead of $\boldsymbol{x}_{t-1}$ when computing $\boldsymbol{y}_t$.

We then move on to $(\boldsymbol{x}_2, \boldsymbol{y}_2)$ and compute from the rules above which components, i.e. values of i, can be initialised given the initialisation of the past iterates. For simultaneous methods, we see that we still cannot initialise the second component of $\boldsymbol{x}_2$ since that would require the second component of $\boldsymbol{x}_0$ or $\boldsymbol{x}_1$, or the second or third components of either $\boldsymbol{y}_0$ or $\boldsymbol{y}_1$ to have been initialised. Nevertheless, given that the second component of $\boldsymbol{y}_2$ depends on the first component of $\boldsymbol{x}_0, \boldsymbol{x}_1$, we may initialise a second component in $\boldsymbol{y}_2$. However, a third component would require either the third component of $\boldsymbol{y}_0, \boldsymbol{y}_1$ or the second or third components of $\boldsymbol{x}_0, \boldsymbol{x}_1$ to be already initialised, which is not the case. Therefore, in simultaneous one-step methods, only 1 component of $\boldsymbol{x}_2$ and 2 components of $\boldsymbol{y}_2$ will be initialised at most.

This logic is applied in table 2, which indicates the number of components in both sets of parameters that have been updated from their initial value (e.g. that are nonzero if we initialise the parameters at 0) at each iteration.

Table 2. Number of components initialised in $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ at iteration t, for methods using $\mathbf{w}_i, \boldsymbol{A}\mathbf{w}_i, \mathbf{b}$

Iteration	Simultaneous		Alt. $\boldsymbol{x}_t$ instead of $\boldsymbol{x}_{t-1}$ for $\boldsymbol{y}_t$		Alt. $\boldsymbol{y}_t$ instead of $\boldsymbol{y}_{t-1}$ for $\boldsymbol{x}_t$	
t	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$
0	0	0	0	0	0	0
1	1	1	1	2	1	1
2	1	2	2	3	2	2
3	2	2	3	4	3	3
4	2	3	4	5	4	4

A simple proof by induction can generalise that for both alternating or simultaneous updates, at most $t+1$ components of $\boldsymbol{x}_t, \boldsymbol{y}_t$ have been initialised. The consequence is that at iteration t we have $\boldsymbol{x}_t(i) = \boldsymbol{x}_0(i), \boldsymbol{y}_t(i) = \boldsymbol{y}_0(i)$ for $i > t+1$, where $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$. Note that this still holds if we compute elements of the span with diagonal matrices as coefficients. This can be summarised as the following.

Lemma 10 (One-step linear span domino argument)**.** *Suppose $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$. Then for algorithms satisfying the one-step linear span assumption (where elements of the span may be computed using diagonal matrices as coefficients), we have*

$$\begin{matrix} \boldsymbol{x}_t(i) = 0 \\ \boldsymbol{y}_t(i) = 0 \end{matrix} \qquad \textit{for } i > t+1 \tag{37}$$

B.1.2. Two-step linear span assumption

For an algorithm satisfying the two-step assumption such as extragradient (see eq. 16 for the update rule), i.e. if we have

$$\mathbf{w}_t \in \mathbf{w}_0 + \text{Span}\big(\mathbf{w}_0, ..., \mathbf{w}_{t-1}, \boldsymbol{A}\mathbf{w}_0, ..., \boldsymbol{A}\mathbf{w}_{t-1}, \boldsymbol{A}^2\mathbf{w}_0, ..., \boldsymbol{A}^2\mathbf{w}_{t-1}, \mathbf{b}, \boldsymbol{A}\mathbf{b}\big) \tag{38}$$

the part of $\mathbf{b}$ and $\boldsymbol{A}\mathbf{b}$ contributing to the update rule on $\boldsymbol{x}$ will be a vector of the form $(*, 0, ...)$ and the part contributing to the update rule on $\boldsymbol{y}$ will have the form $(*, *, 0, ...)$. Therefore,

$$\boldsymbol{x}_t \in \text{Span}\big((*, 0, 0, ...), \boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}, \boldsymbol{M}\boldsymbol{y}_0, ..., \boldsymbol{M}\boldsymbol{y}_{t-1}, \boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{x}_0, ..., \boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{x}_{t-1}\big) \tag{39}$$

$$\boldsymbol{y}_t \in \text{Span}\big((*, *, 0, ...), \boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}, \boldsymbol{M}^\top\boldsymbol{x}_0, ..., \boldsymbol{M}^\top\boldsymbol{x}_{t-1}, \boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{y}_0, ..., \boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{y}_{t-1}\big) \tag{40}$$

We can see from eq. 56 ($\boldsymbol{M}^\top\boldsymbol{M}$ yields the same matrix with a_0^2 instead of $a_0^2 + a_1^2$ in the first entry) that the i-th component of $\boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{x}$ depends on the $i-1, i, i+1$-th components of $\boldsymbol{x}$ for $i \geq 2$. Since only the number of initialised components will interest us, we want to see, at every iteration t, how many components of $\boldsymbol{x}_t, \boldsymbol{y}_t$ received information from past iterations and therefore could have changed from their initial values of zero. The dependence of the components of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ on past

iterates, based on the two-step linear span assumption, is summarised below:

Comp. $i = 1$ of $\boldsymbol{x}_t$: comp. 1 from const., $i, i+1$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$, and $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$
Comp. $i = 1$ of $\boldsymbol{y}_t$: comp. 1 from const., $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$, and i from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$
Comp. $i = 2$ of $\boldsymbol{x}_t$: comp. $i-1, i, i+1$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$, and $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$
Comp. $i = 2$ of $\boldsymbol{y}_t$: comp. 1 from const., $i-1, i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$, and $i-1, i$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$
Comp. $i > 2$ of $\boldsymbol{x}_t$: comp. $i-1, i, i+1$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$, and $i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$
Comp. $i > 2$ of $\boldsymbol{y}_t$: comp. $i-1, i, i+1$ from $\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}$, and $i-1, i$ from $\boldsymbol{x}_0, ..., \boldsymbol{x}_{t-1}$

so for the first few iterations we get Table 3.

Table 3. Number of components initialised in $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ for methods using $\mathbf{w}_i, \boldsymbol{A}\mathbf{w}_i, \boldsymbol{A}^2\mathbf{w}_i, \mathbf{b}, \boldsymbol{A}\mathbf{b}$

Iteration	Simultaneous		Alt. $\boldsymbol{x}_t$ instead of $\boldsymbol{x}_{t-1}$ for $\boldsymbol{y}_t$		Alt. $\boldsymbol{y}_t$ instead of $\boldsymbol{y}_{t-1}$ for $\boldsymbol{x}_t$	
t	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$	# dim $\boldsymbol{x}_t$	# dim $\boldsymbol{y}_t$
0	0	0	0	0	0	0
1	1	2	1	2	2	2
2	2	3	2	3	3	3
3	3	4	3	4	4	4
4	4	5	4	5	5	5

Hence, we can prove once again by induction that in any case at iteration t we have $\boldsymbol{x}_t(i) = \boldsymbol{x}_0(i), \boldsymbol{y}_t(i) = \boldsymbol{y}_0(i)$ for $i > t+1$ and $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$ for methods also accessing $\boldsymbol{A}^2\mathbf{w}_i, \boldsymbol{A}\mathbf{b}$. Here again, this still holds if we multiply the entries in our span by diagonal matrices. We can once again summarise this as a lemma.

Lemma 11 (Two-step linear span domino argument)**.** *Suppose $(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$. Then for algorithms satisfying the two-step linear span assumption (where elements of the span may be computed using diagonal matrices as coefficients), we have*

$$\begin{matrix} \boldsymbol{x}_t(i) = 0 \\ \boldsymbol{y}_t(i) = 0 \end{matrix} \qquad \text{for } i > t+1 \tag{41}$$

B.2. Proof of Prop. 2

We look for stationary points $(\boldsymbol{x}^*, \boldsymbol{y}^*)$:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*, \boldsymbol{y}^*) = c\boldsymbol{M}\boldsymbol{y}^* - d_1\boldsymbol{e_1} + \mu_1\boldsymbol{x}^* = 0 \tag{42}$$
$$\nabla_{\boldsymbol{y}} f(\boldsymbol{x}^*, \boldsymbol{y}^*) = c\boldsymbol{M}^\top\boldsymbol{x}^* + d_2\boldsymbol{e_1} - \mu_2\boldsymbol{y}^* = 0 \tag{43}$$

Therefore, denoting $x_i = \boldsymbol{x}^*(i), y_i = \boldsymbol{y}^*(i)$, the components of stationary points satisfy the recurrence:

$$x_1 : a_0 c y_1 + a_1 c y_2 - d_1 + \mu_1 x_1 = 0 \tag{44}$$
$$y_1 : a_0 c x_1 + d_2 - \mu_2 y_1 = 0 \tag{45}$$

and for $n \geq 2$:

$$x_n : a_0 c y_n + a_1 c y_{n+1} + \mu_1 x_n = 0 \tag{46}$$
$$y_n : a_1 c x_{n-1} + a_0 c x_n - \mu_2 y_n = 0 \tag{47}$$

We can rewrite the above as

$$x_n = -a_0 \frac{c}{\mu_1} y_n - a_1 \frac{c}{\mu_1} y_{n+1} \tag{48}$$
$$y_n = a_1 \frac{c}{\mu_2} x_{n-1} + a_0 \frac{c}{\mu_2} x_n \tag{49}$$

and using eq. 49 to substitute y_n in eq. 46 we get a recurrence on x only:

$$a_0 a_1 \frac{c^2}{\mu_2} x_{n-1} + a_0^2 \frac{c^2}{\mu_2} x_n + a_1^2 \frac{c^2}{\mu_2} x_n + a_0 a_1 \frac{c^2}{\mu_2} x_{n+1} + \mu_1 x_n = 0 \tag{50}$$

which can be rewritten as

$$a_0 a_1 x_{n+1} + \left(\frac{\mu_1 \mu_2}{c^2} + a_0^2 + a_1^2 \right) x_n + a_0 a_1 x_{n-1} = 0 \tag{51}$$

The roots of the characteristic polynomial of the above linear recurrence are given by

$$\chi_\pm = \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) \pm \sqrt{\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right)^2 - 4a_0^2 a_1^2}}{2a_0 a_1} \tag{52}$$

and the solution to the linear recurrence is given by $x_n = C_1 \chi_+^n + C_2 \chi_-^n$ (see (Brassard and Bratley, 1996) for a reference on solving linear recurrences). Note that

$$\begin{aligned} \chi_\pm + 1 &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) \pm \sqrt{\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right)^2 - 4a_0^2 a_1^2} + 2a_0 a_1}{2a_0 a_1} \\ &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + (a_0 - a_1)^2\right) \pm \sqrt{\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right)^2 - 4a_0^2 a_1^2}}{2a_0 a_1} \end{aligned} \tag{53}$$

Suppose $a_0 a_1 > 0$. As $\frac{\mu_1\mu_2}{c^2} > 0$, we have $\chi_- + 1 < 0$ i.e. $|\chi_-| > 1$. Similarly, if we had $a_0 a_1 < 0$ instead, we would have $\chi_- - 1 > 0$ which also yields $|\chi_-| > 1$. Therefore, χ_- is not a solution as it will not yield a $\boldsymbol{x}$ in ℓ_2. However, note that $\chi_+ \chi_- = 1$ which implies that we always have $|\chi_+| < 1$. Hence, we are only concerned with $\chi \triangleq \chi_+$. Moreover, note that the square root always exist as we can rewrite the content of the square root to show that it is always positive:

$$\begin{aligned} \chi &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) + \sqrt{\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right)^2 - 4a_0^2 a_1^2}}{2a_0 a_1} \\ &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) + \sqrt{\left(\frac{\mu_1\mu_2}{c^2}\right)^2 + 2\frac{\mu_1\mu_2}{c^2}\left(a_0^2 + a_1^2\right) + \left(a_0^2 + a_1^2\right)^2 - 4a_0^2 a_1^2}}{2a_0 a_1} \\ &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) + \sqrt{\left(\frac{\mu_1\mu_2}{c^2}\right)^2 + 2\frac{\mu_1\mu_2}{c^2}\left(a_0^2 + a_1^2\right) + \left(a_0^2 - a_1^2\right)^2}}{2a_0 a_1} \end{aligned} \tag{54}$$

In order to simplify the results, we let $a_0 = -a_1 = 1$ and we get:

$$\chi = \left(\frac{\mu_1\mu_2}{2c^2} + 1\right) - \sqrt{\left(\frac{\mu_1\mu_2}{2c^2}\right)^2 + \frac{\mu_1\mu_2}{c^2}} \tag{55}$$

One may note that $L_{12} = c\sqrt{\rho(\boldsymbol{M}\boldsymbol{M}^\top)}$. As we have

$$\boldsymbol{M}\boldsymbol{M}^\top = \begin{bmatrix} a_0^2 + a_1^2 & a_0 a_1 & 0 & 0 & \dots \\ a_0 a_1 & a_0^2 + a_1^2 & a_0 a_1 & 0 & \dots \\ 0 & a_0 a_1 & a_0^2 + a_1^2 & a_0 a_1 & \dots \\ \vdots & & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ \vdots & & \ddots & \ddots & \ddots \end{bmatrix} \tag{56}$$

we note that $\boldsymbol{M}\boldsymbol{M}^\top$ is a tridiagonal Toeplitz matrix, for which the upper end of the spectrum is given by (see Theorem 7.20 of Douglas (2012))

$$\begin{aligned} \sup |\sigma(\boldsymbol{M}\boldsymbol{M}^\top)| &= \operatorname*{ess\,sup}_{\theta \in [0, 2\pi)} \left(a_0 a_1 e^{-i\theta} + (a_0^2 + a_1^2) + a_0 a_1 e^{i\theta}\right) \\ &= \operatorname*{ess\,sup}_{\theta \in [0, 2\pi)} \left(2a_0 a_1 \cos\theta + (a_0^2 + a_1^2)\right) = 4 \end{aligned} \tag{57}$$

and therefore $L_{12} = 2c$. Defining the condition number as

$$\kappa = \frac{L_{12}}{\sqrt{\mu_1 \mu_2}} \tag{58}$$

to retrieve the condition number from the upper bound literature, we get that

$$\begin{aligned}
\chi &= \left(\frac{2}{\kappa^2} + 1\right) - \sqrt{\frac{4}{\kappa^4} + \frac{4}{\kappa^2}} \\
&= \left(\frac{2}{\kappa^2} + 1\right) - \frac{2}{\kappa^2}\sqrt{\kappa^2 + 1} \\
&= 1 - 2\frac{\sqrt{\kappa^2+1} - 1}{(\kappa^2+1) - 1} \\
&= 1 - \frac{2}{\sqrt{\kappa^2+1} + 1}
\end{aligned} \tag{59}$$

Going back to the recurrence, and given that the recurrence on y_n can be shown to be the same as eq. 51, we get that if $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a stationary point of f in $\ell_2 \times \ell_2$, then

$$\boldsymbol{x}^*(i) = x_i = c_1 \chi^i \tag{60}$$

$$\boldsymbol{y}^*(i) = y_i = c_2 \chi^i \tag{61}$$

where c_1, c_2 can be determined from the initial conditions given in eq. 44 and 45. Using the domino argument, which yields that $\forall i > t+1, \boldsymbol{x}_t(i) = 0$, we get that the distance to the optimum of $\boldsymbol{x}$ is given by

$$\begin{aligned}
\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 = \sum_{i=1}^{t+1} (\boldsymbol{x}_t(i) - \boldsymbol{x}^*(i))^2 + \sum_{i=t+2}^{\infty} (\boldsymbol{x}_t(i) - \boldsymbol{x}^*(i))^2 &\geq \sum_{i=t+2}^{\infty} (\boldsymbol{x}^*(i))^2 \\
&= c_1^2 \sum_{i=t+2}^{\infty} \chi^{2i} \\
&= c_1^2 \sum_{i=1}^{\infty} \chi^{2(i+t+1)} \\
&= \chi^{2(t+1)} \|\boldsymbol{x}^*\|^2
\end{aligned} \tag{62}$$

Similarly, we can show that $\|\boldsymbol{y}_t - \boldsymbol{y}^*\|^2 \geq \chi^{2(t+1)} \|\boldsymbol{y}^*\|^2$.

Changing back our variables to $\boldsymbol{x} \to \boldsymbol{x} - \boldsymbol{x}_0$, $\boldsymbol{y} \to \boldsymbol{y} - \boldsymbol{y}_0$ yields the bound for arbitrary initialisation.

B.3. Proof of Thm. 3

If one computes μ_{12} for the function used in the previous bound, it becomes clear that $\mu_{12} = 0$. As such, it is not surprising that the bound failed to hold vs the upper bound of negative momentum on bilinear games with $\mu_{12} > 0$: the previous bound failed to be general enough because the example has the worst possible value of μ_{12}. An important note, however, is that the previous bound may still hold in finite dimensions if we only used it to lower bound the rate of convergence of games with $\mu_{12} = 0$, but it can easily be checked that the rate in the improved bound with $\mu_{12} = 0$ reduces to the first bound.

In order to address this, we will pick values of a_0 and a_1 that allow the counterexample to handle any value of μ_{12}. The proof of the improved domino bound follows the same line of argumentation as the proof of the first bound. We resume from eq. 57, and set $c = 1$, and suppose that $a_1 < 0, a_0 > 0$ such that $|a_1| \leq a_0$. Theorem 7.20 of Douglas (2012) yields that:

$$\max \sigma(\boldsymbol{M}\boldsymbol{M}^\top) = a_0^2 + a_1^2 - 2a_0 a_1 = (a_0 - a_1)^2 \tag{63}$$

$$\min \sigma(\boldsymbol{M}\boldsymbol{M}^\top) = a_0^2 + a_1^2 + 2a_0 a_1 = (a_0 + a_1)^2 \tag{64}$$

Thus, we have $\mu_{12}^2 = (a_0 + a_1)^2,\ L_{12}^2 = (a_0 - a_1)^2$ and since we assumed $|a_1| \leq a_0$, we get that $\mu_{12} = a_0 + a_1, L_{12} = a_0 - a_1$ which allows us to choose a_0, a_1 to make μ_{12}, L_{12} appear in the bound:

$$a_0 = \frac{L_{12} + \mu_{12}}{2} \tag{65}$$

$$a_1 = \frac{\mu_{12} - L_{12}}{2} \tag{66}$$

Noting further that $a_0^2 + a_1^2 = \frac{L_{12}^2 + \mu_{12}^2}{2},\ a_0^2 - a_1^2 = \mu_{12} L_{12}$, we have that

$$\begin{aligned}
\chi &= \frac{-\left(\frac{\mu_1\mu_2}{c^2} + a_0^2 + a_1^2\right) + \sqrt{\left(\frac{\mu_1\mu_2}{c^2}\right)^2 + 2\frac{\mu_1\mu_2}{c^2}\left(a_0^2 + a_1^2\right) + \left(a_0^2 - a_1^2\right)^2}}{2a_0 a_1} \\
&= \frac{-\left(\mu_1\mu_2 + \frac{L_{12}^2 + \mu_{12}^2}{2}\right) + \sqrt{\left(\mu_1\mu_2\right)^2 + 2\mu_1\mu_2\left(\frac{L_{12}^2 + \mu_{12}^2}{2}\right) + \left(\mu_{12}L_{12}\right)^2}}{\frac{\mu_{12}^2 - L_{12}^2}{2}} \\
&= \frac{-\left(2\mu_1\mu_2 + L_{12}^2 + \mu_{12}^2\right) + 2\sqrt{\left(\mu_1\mu_2\right)^2 + \mu_1\mu_2\left(L_{12}^2 + \mu_{12}^2\right) + \left(\mu_{12}L_{12}\right)^2}}{\mu_{12}^2 - L_{12}^2 + \mu_1\mu_2 - \mu_1\mu_2}
\end{aligned} \tag{67}$$

Letting $d_\mu = \mu_1\mu_2 + \mu_{12}^2,\ d_L = \mu_1\mu_2 + L_{12}^2$,

$$\begin{aligned}
\chi &= \frac{-\left(d_\mu + d_L\right) + 2\sqrt{\mu_1\mu_2\left(\mu_1\mu_2 + L_{12}^2\right) + \mu_{12}^2\left(\mu_1\mu_2 + L_{12}^2\right)}}{d_\mu - d_L} \\
&= \frac{\left(d_\mu + d_L\right) - 2\sqrt{d_\mu d_L}}{d_L - d_\mu} \\
&= \frac{\left(\sqrt{d_L} - \sqrt{d_\mu}\right)^2}{\sqrt{d_L}^2 - \sqrt{d_\mu}^2} \\
&= \frac{\sqrt{d_L} - \sqrt{d_\mu}}{\sqrt{d_L} + \sqrt{d_\mu}} \\
&= 1 - \frac{2}{\sqrt{\frac{d_L}{d_\mu}} + 1}
\end{aligned} \tag{68}$$

Letting $\kappa = \frac{d_L}{d_\mu} = \frac{L_{12}^2 + \mu_1\mu_2}{\mu_{12}^2 + \mu_1\mu_2}$ and proceeding as in the proof of the previous bound with the new value of χ yields Thm. 3. Note that as promised, this rate reduces to that of the first bound if $\mu_{12} = 0$:

$$\begin{aligned}
1 - \frac{2}{\sqrt{\frac{d_L}{d_\mu}} + 1} &= 1 - \frac{2}{\sqrt{\frac{L_{12}^2 + \mu_1\mu_2}{\mu_1\mu_2}} + 1} \\
&= 1 - \frac{2}{\sqrt{\kappa_{old}^2 + 1} + 1}
\end{aligned} \tag{69}$$

$$\tag{70}$$

where κ_{old} is the condition number of eq. 58.

C. Proofs of p-SCLI-n

C.1. Proof of Prop. 7

In this section, we follow Arjevani et al. (2016) to derive results for the p-SCLI-n methods. First, we reproduce several definitions and theorems that are proven in Arjevani et al. (2016) and that apply directly to the generalisation. Here, $\boldsymbol{A}$ will denote the Jacobian of some quadratic game with $f_{\boldsymbol{A},\mathbf{b}} \in \mathcal{Q}^{d_1,\dots,d_n}$ such that 0 is not in the spectrum of $\boldsymbol{A}$.

Definition 12 (Characteristic polynomial of a p-SCLI-n)**.** *Let $\mathcal{A}$ be a p-SCLI-n optimisation algorithm with coefficient matrices C_i as defined in def. 5. Then for $\boldsymbol{X} \in \mathbb{R}^{d\times d}$, the characteristic polynomial of $\mathcal{A}$ is given by*

$$\mathcal{L}(\lambda, \boldsymbol{X}) \triangleq \boldsymbol{I}_d\lambda^p - \sum_{i=0}^{p-1} \mathbb{E}C_i(\boldsymbol{X})\lambda^i \tag{71}$$

and its root radius is

$$\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{X})) = \rho(\det \mathcal{L}(\lambda, \boldsymbol{X})) = \max\{|\lambda| \mid \det \mathcal{L}(\lambda, \boldsymbol{X}) = 0\}$$

Theorem 13 (Consistency - characteristic polynomial (Based on Theorem 5 of Arjevani et al. (2016)))**.** *A p-SCLI-n algorithm $\mathcal{A}$ with characteristic polynomial $\mathcal{L}(\lambda, \boldsymbol{X})$ and inversion matrix $N(\boldsymbol{X})$ is consistent with respect to $\boldsymbol{A}$ if and only if the following two conditions hold:*

$$1.\ \mathcal{L}(1, \boldsymbol{A}) = -\mathbb{E}N(\boldsymbol{A})\boldsymbol{A} \tag{72}$$
$$2.\ \rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A})) < 1 \tag{73}$$

We may rephrase theorem 13 of Arjevani et al. (2016) (and lower bound t^{m-1} by 1 since $m \in \mathbb{N}$) as the following to use the root radius of the characteristic polynomial to show linear rates:

Theorem 14 (Based on Theorem 13 of Arjevani et al. (2016))**.** *If $\boldsymbol{A}$ is the Jacobian of a quadratic game and $\mathcal{A}$ is a p-SCLI-n, there exists an initialisation point $\mathbf{w}_0 \in \mathbb{R}^d$ such that*

$$\max_{i=0,\dots,p-1} \left\|\mathbb{E}\mathbf{w}^{t+i} - \mathbb{E}\mathbf{w}^*\right\| \in \Omega(\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))^t) \tag{74}$$

In other words, this means that $\mathcal{A}$ cannot converge on $f_{\boldsymbol{A},\mathbf{b}}$ with linear rate faster than $\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))$, up to a constant. As Arjevani et al. (2016) argue, in both deterministic and stochastic settings, a lower bound on $\|\mathbb{E}\left[\mathbf{w}^t - \mathbf{w}^*\right]\|^2$ implies[4] a lower bound on $\mathbb{E}\left\|\mathbf{w}^t - \mathbf{w}^*\right\|^2$, since

$$\mathbb{E}\left[\left\|\mathbf{w}^t - \mathbf{w}^*\right\|^2\right] = \mathbb{E}\left[\left\|\mathbf{w}^t - \mathbb{E}\mathbf{w}^t\right\|^2\right] + \left\|\mathbb{E}\left[\mathbf{w}^t - \mathbf{w}^*\right]\right\|^2 \tag{75}$$

We can now focus on finding a lower bound on $\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))$.

Proposition 15. *Let $\mathcal{A}$ be a p-SCLI-n optimisation algorithm with inversion matrix $N(\boldsymbol{X})$ that is consistent with respect to $\boldsymbol{A}$. Then,*

$$\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A})) \geq \max_{j=1,\dots,d} |\sqrt[p]{|\sigma_j(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A})|} - 1| \tag{76}$$

where the $\sigma_j(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A})$ are elements of the spectrum (eigenvalues) of $-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A}$.

C.1.1. Proof of Prop. 15

Our proof starts exactly as the one presented by Arjevani et al. (2016) for the $n = 1$ particular case, where the authors assume that $\boldsymbol{A}$ is symmetric with strictly positive spectrum. However, we will generalise the proof to cover non-symmetric matrices and matrices that may not have strictly positive spectrum, since the Jacobian of a quadratic n-player game generally does not have these properties.

[4] Note that since we only use in this paper a lower bound on the second term of the right hand-side of the equation to bound the left hand-side, one may derive in stochastic settings tighter lower bounds than the ones presented in this paper by factoring in the first term of the right hand-side. We leave this as future work.

Let $\mathcal{A}$ be a deterministic p-SCLI-n optimisation algorithm with characteristic polynomial $\mathcal{L}(\lambda, \boldsymbol{X})$ and inversion matrix $N(\boldsymbol{X})$, and $f_{A,\mathbf{b}}(\mathbf{w}) \in \mathcal{Q}^{d_1,\dots,d_n}$ represent a quadratic n-player game. Since $\mathcal{A}$ is p-SCLI-n, its (expected) coefficient matrices $\mathbb{E}C_i$ evaluated on $\boldsymbol{A}$ are simultaneously triangularisable, so $\exists \boldsymbol{Q} \in \mathbb{R}^{d\times d}$ such that for $i = 0, ..., p-1$, we have

$$\boldsymbol{T}_i \triangleq \boldsymbol{Q}^{-1}\mathbb{E}C_i(\boldsymbol{A})\boldsymbol{Q} \tag{77}$$

where $\boldsymbol{T}_i$ is triangular. Thus,

$$\begin{aligned}\det \mathcal{L}(\lambda, \boldsymbol{A}) &= \det\left(\boldsymbol{Q}^{-1}\mathcal{L}(\lambda, \boldsymbol{A})\boldsymbol{Q}\right)\\ &= \det\left(\boldsymbol{I}_d\lambda^p - \sum_{i=0}^{p-1}\boldsymbol{T}_i\lambda^i\right)\end{aligned} \tag{78}$$

Since $\boldsymbol{I}_d\lambda^p - \sum_{i=0}^{p-1}\boldsymbol{T}_i\lambda^i$ is a upper triangular matrix, its determinant is given by

$$\det \mathcal{L}(\lambda, \boldsymbol{A}) = \prod_{j=1}^{d}\ell_j(\lambda) \tag{79}$$

where

$$\ell_j(\lambda) = \lambda^p - \sum_{i=0}^{p-1}\sigma_j^i\lambda^i \tag{80}$$

and where $\sigma_1^i, \dots, \sigma_d^i,\ i = 0, \dots, p-1$ denote the elements on the diagonal of $\boldsymbol{T}_i$, which are just the eigenvalues of $\mathbb{E}C_i$ ordered according to $\boldsymbol{Q}$. Hence, the root radius of the characteristic polynomial of $\mathcal{A}$ is

$$\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A})) = \max\{|\lambda| \mid \ell_j(\lambda) = 0 \text{ for some } j = 1, ..., d\} \tag{81}$$

On the other hand, by consistency condition (72) we get that for all $j = 1, ..., d$

$$\ell_j(1) = \sigma_j\left(\mathcal{L}(1, \boldsymbol{A})\right) = \sigma_j\left(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A}\right) \tag{82}$$

In the case of p-SCLI-1, the authors prove their Corollary 7 (i.e. our prop. 15 without taking the modulus of the eigenvalues) by using a lemma (see Lemma 6 in Arjevani et al. (2016)) that gives a lower bound on each $\rho(\ell_j(\lambda))$ by using the sign of $\ell_j(1) = \sigma_j\left(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A}\right)$. Lemma 6 of Arjevani et al. (2016) is proven using the following lemma, which we can in fact use to handle arbitrary eigenvalues (e.g. complex or negative).

Lemma 16 (Lemma 15 of Arjevani et al. (2016))**.** *Let $q_r^*(z) \triangleq (z - (1 - \sqrt[p]{r}))^p$ where r is some non-negative constant. Suppose $q(z)$ is a monic polynomial of degree p with complex coefficients. Then,*

$$\rho(q(z)) \leq |\sqrt[p]{|q(1)|} - 1| \iff q(z) = q^*_{|q(1)|}(z)$$

The proof of the lemma can be found in Arjevani et al. (2016). Here, we can use the lemma directly on each ℓ_j with $q = \ell_j$ and $r = |q(1)| = |\ell_j(1)| = |\sigma_j\left(-\mathbb{E}[N(\boldsymbol{A})]A\right)|$. Indeed, since $r \geq 0$,

- if $q(z) = q_r^*(z) = (z - (1 - \sqrt[p]{r}))^p$ then clearly $\rho(q(z)) = |1 - \sqrt[p]{r}|$
- if $q(z) \neq q_r^*(z)$, then we have $\rho(q(z)) > |\sqrt[p]{|q(1)|} - 1|$

Which implies that for any j we have $\rho(\ell_j(\lambda)) \geq |\sqrt[p]{|\ell_j(1)|} - 1| = |\sqrt[p]{|\sigma_j\left(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A}\right)|} - 1|$. Using this in eq. 81 yields

$$\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A})) \geq \max_{j=1,\dots,d}|\sqrt[p]{|\sigma_j(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A})|} - 1| \tag{83}$$

C.1.2. Deriving the optimal ρ for scalar inversion matrices

We are now ready to obtain the general lower bound. Consider $f_{\boldsymbol{A},\mathbf{b}} \in \mathcal{Q}^{d_1,\dots,d_n}$ with $0 \notin \sigma(\boldsymbol{A})$ and a consistent p-SCLI-n algorithm $\mathcal{A}$. Let $\mu = \min|\sigma(\boldsymbol{A})|$, $L = \max|\sigma(\boldsymbol{A})|$ where $\sigma(\boldsymbol{A})$ is the spectrum of $\boldsymbol{A}$. For a scalar inversion matrix i.e. $\mathbb{E}[N(\boldsymbol{A})] = \nu$ we have from eq. 83:

$$\begin{aligned}\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A})) &\geq \max_{j=1,\dots,d} |\sqrt[p]{|\sigma_j(-\mathbb{E}[N(\boldsymbol{A})]\boldsymbol{A})|} - 1| = \max_{j=1,\dots,d} |\sqrt[p]{|\nu\sigma_j(\boldsymbol{A})|} - 1| \\ &= \max\left\{|\sqrt[p]{|\nu|\mu} - 1|, |\sqrt[p]{|\nu|L} - 1|\right\} \end{aligned} \tag{84}$$

Note that consistency (eq. 73) constrains $\nu \in \left(\frac{-2^p}{L}, \frac{2^p}{L}\right) \setminus \{0\}$. We proceed as Arjevani et al. (2016) in the p-SCLI-1 case, and study the ranges of $|\nu|$ by using $\max(a,b) = \frac{a+b+|a-b|}{2}$ to obtain table 4.

Table 4. Lower bound for ρ by subranges of $|\nu|$ and minimiser $|\nu^*|$

	$\sqrt[p]{\lvert\nu\rvert\mu} - 1 < 0$			$\sqrt[p]{\lvert\nu\rvert\mu} - 1 \geq 0$		
	Range	Minimiser	Bound	Range	Minimiser	Bound
$\sqrt[p]{\lvert\nu\rvert L} - 1 \leq 0$	$(0, 1/L]$	$1/L$	$1 - \sqrt[p]{\frac{\mu}{L}}$		N/A	
$\sqrt[p]{\lvert\nu\rvert L} - 1 > 0$	$(1/L, 1/\mu)$	$\left(\frac{2}{\sqrt[p]{L} + \sqrt[p]{\mu}}\right)^p$	$\frac{\sqrt[p]{L/\mu}-1}{\sqrt[p]{L/\mu}+1}$	$[1/\mu, 2^p/L)$	$1/\mu$	$\sqrt[p]{\frac{L}{\mu}} - 1$

Note that case 3 requires $p > \log_2 L/\mu$. Hence,

$$\rho \geq \min\left\{1 - \sqrt[p]{\frac{\mu}{L}}, \frac{\sqrt[p]{L/\mu} - 1}{\sqrt[p]{L/\mu} + 1}, \sqrt[p]{\frac{L}{\mu}} - 1\right\} = \frac{\sqrt[p]{L/\mu} - 1}{\sqrt[p]{L/\mu} + 1} \tag{85}$$

where $\mu = \min|\sigma(\boldsymbol{A})|$, $L = \max|\sigma(\boldsymbol{A})|$.

C.2. Finding a suitably hard example for 2-player with $d_1 = d_2$

We now only need to find a hard counterexample. We present the argument for $d_1 = d_2 = 2$, which can easily be generalised for arbitrary d. Consider the matrix

$$\boldsymbol{A} = \begin{pmatrix} \mu_1 & 0 & \mu_{12} & 0 \\ 0 & L_1 & 0 & L_{12} \\ -\mu_{12} & 0 & \mu_2 & 0 \\ 0 & -L_{12} & 0 & L_2 \end{pmatrix} \tag{86}$$

corresponding to the Jacobian of a quadratic game in $\mathcal{Q}^{d_1,d_2}$.

First we compute the characteristic polynomial of A, using the formula for the determinant of a block matrix (see Zhang (2005, Section 0.3) for instance):

$$\det(XI - A) = \det\begin{pmatrix} X-\mu_1 & 0 & -\mu_{12} & 0 \\ 0 & X-L_1 & 0 & -L_{12} \\ \mu_{12} & 0 & X-\mu_2 & 0 \\ 0 & L_{12} & 0 & X-L_2 \end{pmatrix} \tag{87}$$

$$= \det\left(\begin{pmatrix} (X-\mu_1)(X-\mu_2) & 0 \\ 0 & (X-L_1)(X-L_2) \end{pmatrix} + \begin{pmatrix} \mu_{12}^2 & 0 \\ 0 & L_{12}^2 \end{pmatrix}\right) \tag{88}$$

$$= (X^2 - (\mu_1+\mu_2)X + \mu_1\mu_2 + \mu_{12}^2)(X^2 - (L_1+L_2)X + L_1L_2 + L_{12}^2) \tag{89}$$

The discriminants of these two quadratic equations are, respectively:

$$\Delta_\mu = (\mu_1+\mu_2)^2 - 4(\mu_1\mu_2 + \mu_{12}^2) = (\mu_1-\mu_2)^2 - 4\mu_{12}^2 \tag{90}$$

$$\Delta_L = (L_1+L_2)^2 - 4(L_1L_2 + L_{12}^2) = (L_1-L_2)^2 - 4L_{12}^2 \tag{91}$$

which yields the following eigenvalues:

$$\lambda_{\mu\pm} = \frac{\mu_1 + \mu_2}{2} \pm \sqrt{\left(\frac{\mu_1 - \mu_2}{2}\right)^2 - \mu_{12}^2}$$
$$\lambda_{L\pm} = \frac{L_1 + L_2}{2} \pm \sqrt{\left(\frac{L_1 - L_2}{2}\right)^2 - L_{12}^2} \tag{92}$$

We distinguish four cases, which are presented in the following table:

Table 5. Lower bounds on the condition number

	$\Delta_\mu < 0$	$\Delta_\mu \geq 0$
$\Delta_L < 0$	$\kappa = \sqrt{\frac{L_1 L_2 + L_{12}^2}{\mu_1 \mu_2 + \mu_{12}^2}}$	$\kappa \geq 2\frac{\sqrt{L_1 L_2 + L_{12}^2}}{\mu_1 + \mu_2 - \sqrt{\Delta_\mu}}$
$\Delta_L \geq 0$	$\kappa \geq \frac{1}{2}\frac{L_1 + L_2 + \sqrt{\Delta_L}}{\sqrt{\mu_1 \mu_2 + \mu_{12}^2}}$	$\kappa \geq \frac{L_1 + L_2 + \sqrt{\Delta_L}}{\mu_1 + \mu_2 - \sqrt{\Delta_\mu}}$

where we used that $\kappa = \frac{\max|\sigma(\boldsymbol{A})|}{\min|\sigma(\boldsymbol{A})|}$.

We now discuss these four cases:

- If $\Delta_\mu < 0$ and $\Delta_L < 0$, we have that

$$|\lambda_{\mu\pm}| = \left|\frac{\mu_1 + \mu_2}{2} \pm i\sqrt{\mu_{12}^2 - \left(\frac{\mu_1 - \mu_2}{2}\right)^2}\right|$$
$$= \sqrt{\mu_1 \mu_2 + \mu_{12}^2} \tag{93}$$

  Similarly we get

$$|\lambda_{L\pm}| = \sqrt{L_1 L_2 + L_{12}^2} \tag{94}$$

  Clearly then $\min|\sigma(\boldsymbol{A})| = |\lambda_{\mu\pm}|$ and $\max|\sigma(\boldsymbol{A})| = |\lambda_{L\pm}|$, which yields $\kappa = \sqrt{\frac{L_1 L_2 + L_{12}^2}{\mu_1 \mu_2 + \mu_{12}^2}}$.

- If $\Delta_\mu \geq 0$ and $\Delta_L \geq 0$, $\lambda_{L+},\ \lambda_{L-},\ \lambda_{\mu+}$ and $\lambda_{\mu-}$ are all real. We have that,

$$\lambda_{\mu-} \geq \min|\sigma(\boldsymbol{A})|, \quad \text{and} \quad \lambda_{L+} \leq \max|\sigma(\boldsymbol{A})|, \tag{95}$$

  which yields the result.

- If $\Delta_\mu < 0$ and $\Delta_L \geq 0$, it holds that,

$$|\lambda_{\mu\pm}| = \min|\sigma(\boldsymbol{A})|, \quad \text{and} \quad \lambda_{L+} \leq \max|\sigma(\boldsymbol{A})|, \tag{96}$$

  from which we obtain the result.

- Similarly, if $\Delta_\mu \geq 0$ and $\Delta_L < 0$, it holds that,

$$\lambda_{\mu-} \geq \min|\sigma(\boldsymbol{A})|, \quad \text{and} \quad |\lambda_{L\pm}| = \max|\sigma(\boldsymbol{A})|. \tag{97}$$

One could wonder whether our lower bounds on κ when at least one of the discriminant is non-negative are actually equalities. We provide an example showing that it is not the case when $\Delta_L \geq 0$ and $\Delta\mu \geq 0$. A similar one can be found when $\Delta_L < 0$ and $\Delta\mu \geq 0$.

Take $\mu_{12} = 0$ and $L_{12} = \frac{|L_1 - L_2|}{2}$. Then $\Delta_L \geq 0$ and $\Delta\mu \geq 0$. Then,

$$\lambda_{\mu+} = \frac{\mu_1 + \mu_2}{2} + \sqrt{\left(\frac{\mu_1 - \mu_2}{2}\right)^2 - \mu_{12}^2} = \max(\mu_1, \mu_2)$$
$$\lambda_{L\pm} = \frac{L_1 + L_2}{2} \pm \sqrt{\left(\frac{L_1 - L_2}{2}\right)^2 - L_{12}^2} = \frac{L_1 + L_2}{2}\,. \tag{98}$$

Choose $\mu_1 = L_1$, $\mu_2 = L_2$ and $L_1 \neq L_2$. Then $\lambda_{\mu+} > \lambda_{L\pm}$. However we have $\lambda_{\mu-} = \min|\sigma(A)|$ and so in this case $\kappa = \lambda_{\mu+}/\lambda_{\mu-}$.

D. Lower bounds on the iteration complexity

The lower bounds on the distance of the iterates to a minimiser also yield lower bounds on the number of iterations required to reach a maximum error ϵ on the iterates (iteration complexity).

p-SCLI-n case: For the p-SCLI-n bound, suppose $\max_{i=0,\dots,p-1} \left\|\mathbb{E}\mathbf{w}^{t+i} - \mathbb{E}\mathbf{w}^*\right\| < \epsilon$. Then from Theorem 14, we know that for some strictly positive real C,

$$\begin{aligned} C\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))^t \leq \epsilon &\implies \rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))^t \leq \frac{\epsilon}{C} \\ &\implies t \log\left(\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))\right) \leq \log\left(\frac{\epsilon}{C}\right) \\ &\implies t \log\left(\frac{1}{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}\right) \geq \log\left(\frac{C}{\epsilon}\right) \end{aligned} \tag{99}$$

Noting that $\log(x) \leq x - 1$ for $x > 0$, we get that $\log\left(\frac{1}{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}\right) \leq \frac{1-\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}$ and hence,

$$t\frac{1 - \rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))} \geq t \log\left(\frac{1}{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}\right) \geq \log\left(\frac{C}{\epsilon}\right) \tag{100}$$

Therefore, we get that

$$t \geq \frac{\rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))}{1 - \rho_\lambda(\mathcal{L}(\lambda, \boldsymbol{A}))} \log\left(\frac{C}{\epsilon}\right) \tag{101}$$

where one may use Prop. 7 to get

$$t \geq \left(\frac{\sqrt[p]{\kappa} - 1}{2}\right) \log\left(\frac{C}{\epsilon}\right) \tag{102}$$

where κ is given as in Prop. C.1. One may also use in the 2-player case the κ given in Table 1.

2-player domino bound: For the two player bound stemming from the domino argument, it is equally straightforward to establish from Thm. 3 that given an upper bound ϵ on $\|(\boldsymbol{x}_t, \boldsymbol{y}_t) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\|$, then using that $1 - \frac{2}{\kappa+1} \geq \exp\left(-\frac{2}{\kappa-1}\right)$,

$$\exp\left(-\frac{2}{\kappa - 1}\right)^{t+1} \cdot \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\| \leq \epsilon \implies t \geq \frac{\kappa - 1}{2} \log\left(\frac{\|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}^*, \boldsymbol{y}^*)\|}{\epsilon}\right) - 1 \tag{103}$$

where κ is defined as in Thm. 3.