Computing Equilibrium Points of Electrostatic Potentials

Abheek Ghosh
Department of Computer Science, University of Oxford, UK

Paul W. Goldberg
Department of Computer Science, University of Oxford, UK

Alexandros Hollender
All Souls College, University of Oxford, UK

Abstract

We study the computation of *equilibrium points* of electrostatic potentials: locations in space where the electrostatic force arising from a collection of charged particles vanishes. This is a novel scenario of optimization in which solutions are guaranteed to exist due to a nonconstructive argument, but gradient descent is unreliable due to the presence of singularities.

We present an algorithm based on piecewise approximation of the potential function by Taylor series. The main insight is to divide the domain into a grid with variable coarseness, where grid cells are exponentially smaller in regions where the function changes rapidly compared to regions where it changes slowly. Our algorithm finds approximate equilibrium points in time poly-logarithmic in the approximation parameter, but these points are not guaranteed to be close to exact solutions. Nevertheless, we show that such points can be computed efficiently under a mild assumption that we call "strong non-degeneracy". We complement these algorithmic results by studying a generalization of this problem and showing that it is CLS-hard and in PPAD, leaving its precise classification as an intriguing open problem.

2012 ACM Subject Classification Theory of computation → Continuous optimization; Theory of computation → Problems, reductions and completeness; Computing methodologies → Equation and inequality solving algorithms

Keywords and phrases Total search problems, TFNP, PPAD, CLS, polynomial equations

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.69

Related Version *Full Version*: `https://arxiv.org/abs/2512.03249`

Funding A.G. and P.W.G. were supported by EPSRC Grant EP/X040461/1 "Optimisation for Game Theory and Machine Learning".

Acknowledgements The authors wish to thank the reviewers for suggestions that helped improve the presentation of the paper.

1 Introduction

Suppose that we have a finite set P of electrically charged particles, immobile and located at given points in space. These particles give rise to an electrostatic potential: a function V that maps any point x in space to the amount of energy needed to move a unit positive charge from a point at infinity to x. The negated gradient of V constitutes the force acting on a positively charged particle at x: it is attracted to negatively charged particles in P and repelled by positively charged ones. Each $p \in P$ exerts a force at x proportional to r^{-2}, r being the distance between x and p. In this paper, we study the computation of *equilibrium points* where the gradient of V is zero, meaning that the forces exerted by members of P cancel each other out.

For simplicity, let us for now consider the setting where all particles are positively charged.[1] If P consists of a single particle, then no equilibrium point exists, as any (mobile) particle would just be pushed away from the single particle. However, if P contains at least two particles, an equilibrium point is always guaranteed to exist! This is usually proved by using Morse theory.[2] Importantly, the proof of existence is nonconstructive and thus does not yield an efficient algorithm for finding such an equilibrium point.

More generally, although the three-dimensional setting corresponds to the physical world, this problem can be considered in d-dimensional space for any $d \geq 2$, and one can also consider more general potentials, e.g., proportional to r^{-k} for some k. The aforementioned nonconstructive existence result continues to hold and in this paper we initiate the study of conditions under which approximate equilibrium points can be computed efficiently.

Contribution. We study the numerical computation of equilibrium points of electrostatic potentials: given locations of charged particles, and the sign and magnitude of their charges, we want to find approximate equilibrium points (where the force is within some given ε of zero). For this problem, we present an algorithm that runs in polynomial time when the dimension d of the space is constant.

▶ **Informal Theorem 1.** *When the dimension d is constant, an ε-approximate equilibrium point of an electrostatic potential can be computed in time polynomial in the input and* $\log(1/\varepsilon)$.

In particular, our algorithm is efficient in the physical setting where $d = 3$ and even when ε is inverse-exponential.[3] Furthermore, it can handle the presence of both positive and negative charges.

Our algorithm has a conceptual link with the *trust region* approach in optimization [4], in that we divide up the domain into regions within which we approximate the objective function with a more tractable kind of function (a polynomial). The difference is that here, we exploit our knowledge of the structure of the objective function so as to identify regions within which we have a prior guarantee that the approximation achieves the desired precision everywhere in the region. Our algorithm extends to a class of functions with "well-behaved" derivatives, where this subdivision of the domain remains efficient.

One important caveat of our algorithm is that it outputs *weak* approximations of equilibrium points, namely points where the force is approximately zero. Such points are not guaranteed to be close to exact equilibrium points (where the force is exactly zero). Ideally, we would also like to be able to compute *strong* approximations of equilibrium points, namely points that are actually close to exact equilibrium points.

---

[1] Incidentally, the setting where all particles are negatively charged corresponds to the setting of a gravitational field. Furthermore, note that an equilibrium point for a given instance of the problem remains an equilibrium point if the sign of every charged particle is flipped. So, the setting where all particles are positively charged also corresponds to the setting of a gravitational field for the purpose of studying equilibrium points.

[2] Morse theory [19, Chapter 32], which is traditionally used to prove existence in this setting, only guarantees existence for *generic* instances, i.e., when the positions of the charged particles lie outside a set of measure zero. Nevertheless, an equilibrium point is guaranteed to exist even without this genericity assumption; we provide a proof sketch in the full version of the paper.

[3] The attentive reader might have observed that it is trivial to find a point where the force has norm less than ε: it suffices to pick a point that is sufficiently far away from our set of particles P. In order to be able to exclude these artificial solutions, our algorithm also takes as input the description of a bounded domain, and only looks for solutions inside that domain. Furthermore, it is possible to pick a large enough domain that is guaranteed to contain all exact solutions (and thus, in particular, at least one solution).

▶ **Open Problem 1.** *Is there an efficient algorithm that uses algebraic methods in a more direct manner, avoiding the use of numerical approximation?*

The hope would be that such an algorithm would be able to output strong approximate solutions, instead of weak solutions. Although we leave this question open, we show that our numerical approach can be used to compute strong approximate solutions under an additional assumption that we call *strong non-degeneracy*. This condition requires that at any equilibrium point, the determinant of the Hessian of the potential is bounded away from zero.

▶ **Informal Theorem 2.** *When the dimension d is constant and the instance is δ-strongly non-degenerate, a point ε-close to an exact equilibrium point can be computed in time polynomial in the input,* $\log(1/\varepsilon)$*, and* $\log(1/\delta)$*.*

Note that we cannot hope to compute equilibrium points exactly, as these might be irrational.[4] As with our first algorithm, the algorithm is only efficient when the dimension d is fixed. Although this is sufficient for the physical setting, the electrostatic potential in high dimension is relevant to generative models in machine learning. Perhaps the main question left open by our work is thus the following.

▶ **Open Problem 2.** *Is there an efficient algorithm for computing weak (or strong) approximate equilibrium points of electrostatic potentials when the dimension d is not fixed?*

As a first step towards understanding the high-dimensional setting, we show that for the general class of "well-behaved" functions (for which Informal Theorems 1 and 2 hold) finding approximate equilibrium points becomes intractable when d is not fixed.

▶ **Informal Theorem 3.** *There is no polynomial-time algorithm for finding (weak) approximate equilibrium points of well-behaved functions when d is not fixed, unless* $\mathsf{NP} = \mathsf{P}$*.*

Finally, going back to constant dimension, we consider potentials that go beyond the class of well-behaved functions (which contain the electrostatic potentials as a special case). Namely, we consider potentials that are continuously differentiable, have a singularity at the location of the charge, and decrease monotonically as we move away from the charge. Here we show that, already in two dimensions, it is intractable to find an approximate equilibrium point of a system with two charges with these very general potential functions.

▶ **Informal Theorem 4.** *There is no polynomial-time algorithm for finding (weak) approximate equilibrium points of two charges in $\mathbb{R}^2$ with general potential functions, unless* $\mathsf{CLS} = \mathsf{FP}$*.*

CLS [5, 6] is a subclass of TFNP, the class of computational search problems having solutions that are guaranteed to exist, and once they are found, they are easily verifiable. It is believed that CLS is different from FP (the class of search problems solvable in polynomial time), and this is supported by various cryptographic lower bounds [2, 3, 12].

Interestingly, the best upper bound we obtain for the computational problem in Informal Theorem 4 is membership in the class PPAD, a superclass of CLS.

▶ **Open Problem 3.** *What is the complexity of computing an approximate equilibrium point of a general potential with singularities? Is it* CLS*-complete,* PPAD*-complete, or something in between?*

[4] For example, consider two positive charges with magnitudes 1 and 2, one unit apart, and an inverse-square force power law: the equilibrium point between them is $\sqrt{2} - 1$ distant from the charge with value 1.

Given that $\mathsf{CLS} = \mathsf{PPAD} \cap \mathsf{PLS}$ [6], a natural first step would be to investigate whether the problem lies in PLS, i.e., whether it admits a local search algorithm where every individual step can be performed in polynomial time (but the number of steps need not be polynomial).

1.1 Technical Overview

In this section, we give a brief overview of the techniques we use to obtain our results.

1.1.1 Algorithms in constant dimension

To keep the exposition simple in this overview, we focus on the case where all charges are positive. Recall that our goal is to compute approximate equilibrium points of an electrostatic potential.

Two charges. In the case where there are only two particles, it is easy to compute a solution to any desired accuracy. Indeed, if we consider the segment connecting the two charges, then the force pushes inwards on both ends. As a result, using binary search, one can compute a point that is ε-close to an equilibrium point in time $\log(1/\varepsilon)$.

However, for three or more particles, it becomes unclear how to proceed. We begin by presenting two natural approaches and why they fail (or only work in restricted cases).

Attempt 1: Gradient descent. Since we are looking for a point where the gradient of the potential is zero, it is natural to attempt to use gradient descent (or ascent). These gradient methods are normally guaranteed to converge to a point with zero gradient, but in our setting this is not the case, for the following two reasons:

- If we use gradient ascent, then the algorithm might just start ascending towards the position of a particle, where there is no solution. Indeed, there is a singularity at the position of each charge, and the value of the potential goes to infinity as we approach it.
- If we use gradient descent, then the algorithm might just diverge further and further away from our set of particles P, just like a particle that is being pushed away.

Furthermore, a solution may need to be a saddle-point as opposed to a local maximum or minimum. Indeed, for the "natural" power-law function in which the force between two points is proportional to r^{-2}, all solutions are saddle-points![5] The take-away is that the convergence of gradient-based methods is highly dependent on the initialization, and it is unclear whether any theoretical guarantees for convergence can be proved.

Attempt 2: System of polynomial equations. Optimization techniques can be very powerful but they are designed to work for very general and rather unstructured functions. In our setting however, the electrostatic potential is very structured; indeed, it is given by some simple algebraic expressions. It is thus natural to attempt to use this algebraic form.

The equations that an equilibrium point $\boldsymbol{x} \in \mathbb{R}^d$ must satisfy, namely that $\nabla V(\boldsymbol{x}) = 0$, can be written down explicitly. When the dimension d is a constant, it is known that systems of polynomial equations (and inequalities) can be solved efficiently to high accuracy, see, e.g., [20, 11]. Unfortunately, in our case, the equations are not polynomial, since they also

[5] In the 3-dimensional case, this goes back to Maxwell [17, Chapter VI Art. 112]. In d dimensions, if each charge contributes an amount proportional to r^{1-d} to the force at x, the total potential obeys the Laplace equation, implying that solutions are saddle points.

involve roots. A straightforward remedy to this is to introduce additional variables to obtain roots using only polynomial terms; for example, adding a variable y and the equations $y^2 = x, y \geq 0$, ensures that y is the square root of x. Thus, one can eliminate the root terms at the cost of introducing additional variables (and equations/inequalities). When the number of charges is constant, the number of root terms is also constant, and thus the resulting system still has a constant number of variables overall. In that case, the problem can be solved efficiently using the aforementioned polynomial solvers. However, if the number of charges is not constant, this approach fails, as the solvers no longer run in polynomial time when the number of variables is not constant.

Our solution: Taylor approximation in a grid with variable coarseness. This second (partially) failed attempt suggests a somewhat different approach based on numerical approximation. Namely, if we could approximate the electrostatic potential by a polynomial, we could then use the aforementioned solvers for systems of polynomial equations to obtain an approximate solution. Unfortunately, it is not possible to approximate the potential induced by a charge by a single polynomial. Instead, we attempt to divide the domain into small grid cells, approximate the potential by a polynomial in each grid cell, and finally use the polynomial equation solver in each grid cell.

However, if we divide the domain into an evenly spaced grid and use the Taylor approximation of the potential in each grid cell, we encounter an issue. The quality of the approximation depends on how close the grid cell is to the singularity. The closer we are to the singularity, the worse the approximation error. In order to make the error small enough, the grid cells would need to be exponentially fine. Since we can only allow a polynomial number of grid cells overall, we cannot use this fine resolution over the whole domain. Instead, we use a custom grid that becomes finer as we move closer to the singularity.[6] An important point is that, since the potential goes to infinity as we approach the singularity, we can eliminate a small region around the singularity from the domain, because we can guarantee that there is no solution there. Figure 1 shows what this grid looks like for a single charge, and Figure 2 shows the final grid obtained by taking the grids induced by three different charges.

It turns out that this approach – decomposing into a grid of variable coarseness and using Taylor approximation in each grid cell – works more generally as long as a function is *well-behaved* in a sense that we define formally. Namely, we carefully define this well-behavedness property so as to capture the condition that allows for a custom grid with a polynomial number of cells to be used on the function. As a result, our algorithm can be used on a wide variety of potential functions with similar general properties.

The main limitation to our approach is that the algorithm's running time is exponential in the dimension d, and thus is only polynomial for constant d. The dependence on d arises in two places: the number of grid cells, and the running time of the solvers for systems of polynomial inequalities [11, 20], which we use within each grid cell.

Strong approximation. As mentioned earlier, the aforementioned approach yields *weak* approximate equilibrium points, i.e., points where the force is approximately zero, but not necessarily points that are close to exact equilibrium points. In order to find strong approximate solutions, we make the following observation: if a point $\boldsymbol{x}$ is a weak approximate

[6] To be more precise, the grid is fine in a certain coordinate, if we are close to the singularity in that coordinate.

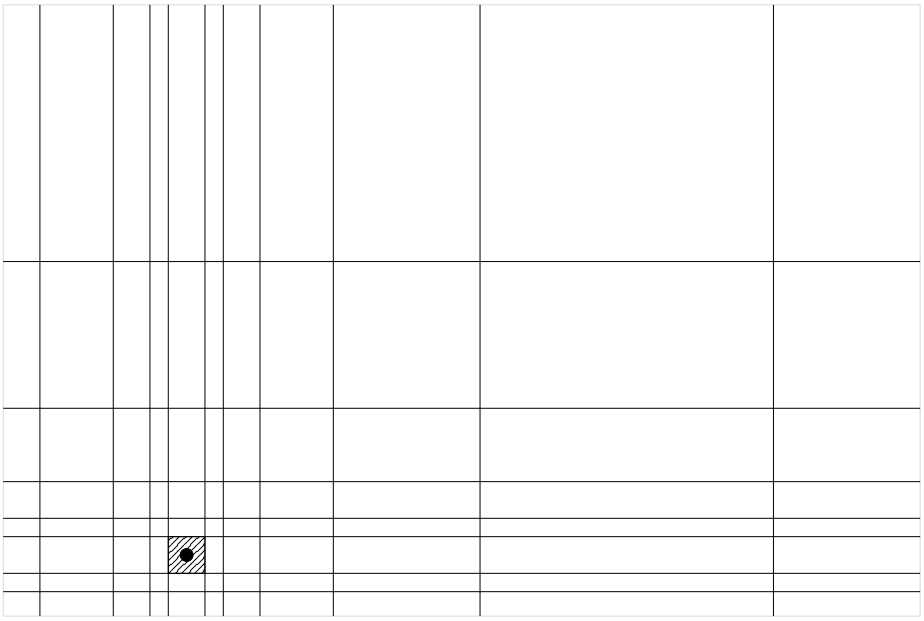


Figure 1 The grid we use to ensure good approximation of the potential induced by a single charge. The charge is located at the black dot. The shaded region around the position of the charge is removed from the domain, since we can guarantee that no solution is located there. Note that the grid becomes finer in a given coordinate when we are close to the singularity in that coordinate.

solution (i.e., $\nabla V(\boldsymbol{x}) \approx 0$) and the determinant of the Hessian at x is sufficiently far from zero, then there must be an exact equilibrium point close to $\boldsymbol{x}$. The intuition is that the Hessian must have full rank in a sufficiently large region around $\boldsymbol{x}$, and as a result, given that $\nabla V(\boldsymbol{x})$ is almost zero, it must take value exactly zero in some point in the vicinity of $\boldsymbol{x}$. More formally, this can be established by using the Poincaré-Miranda theorem. Our algorithmic approach can find such a point (with gradient close to zero and determinant of the Hessian bounded away from zero) efficiently, if it exists.

1.1.2 Generalized potentials

As mentioned earlier, the existence of an equilibrium point of an electrostatic potential can be established using Morse theory [19]. However, this approach has two drawbacks:

- It only yields existence for *generic* instances, i.e., outside some set of measure zero.
- It does not give us containment in any subclass of TFNP for the associated computational problem.

Regarding the second issue, as far as we know, Morse theory has not been studied in the context of total search problems, and this is an interesting direction for future work.

▶ **Open Problem 4.** *Can one define a computational problem to capture Morse theory and use it to prove membership in some* TFNP *class for problems where the existence is shown using this theory?*

Coming back to our electrostatic potential, we bypass both of the aforementioned issues by proposing a different proof of existence that takes care of both of these issues. Here, we give some brief intuition about how the proof can be stated for the two-dimensional setting using a generalization of the hairy ball theorem.

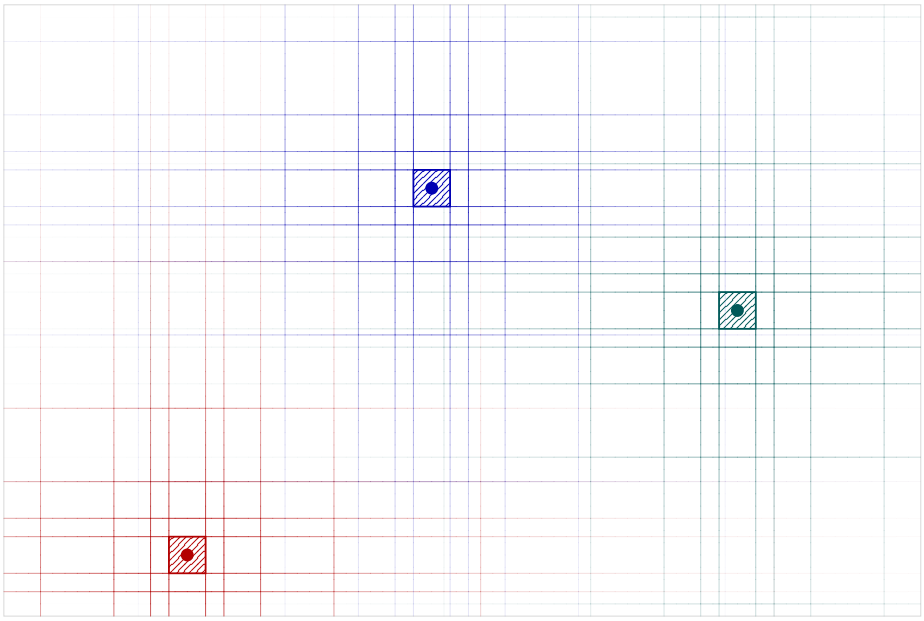


Figure 2 The final grid obtained by superimposing the grids induced by three different charges. The grid lines extend to infinity, but are faded here for improved visibility. Our algorithm solves a system of polynomial inequalities in each grid cell.

A proof of existence using the hairy ball theorem. Let a collection P of charged particles in $\mathbb{R}^2$ be given and assume that the sum of the magnitudes of the charges is strictly positive.[7] Then, there exists a sufficiently large region $S \subseteq \mathbb{R}^2$ containing P such that the force points outside S at any point on the boundary of S. Indeed, if we are far enough from the set of charges P, then, by our assumption, the positive charges will dominate the negative ones, and the force will point away from P.

Next, consider points that are very close to a positive charge $p \in P$. The force must point away from p. Now, remove a small region around p from the set S. Note that the force points towards the inside of S around the boundary of the small region that we removed. For each charge, we remove a sufficiently small region around it. For negative charges, the force will point outside of S around the boundary of the small regions that we remove around these charges. See Figure 3 for an illustration of the set S.

The last step of the proof is to take two copies of S, one with the force field negated, and connect them together along their corresponding boundaries, interpolating the force field smoothly at those boundaries. The resulting object lies in three-dimensional space and it is a torus of genus $|P|$, i.e., "a donut with $|P|$ holes". Due to the boundary conditions that the force had on the set S, we can argue that the "gluing operation" that we performed did not introduce any new solutions.[8] Thus, we have obtained a continuous tangent vector field on

[7] If the sum is strictly negative, then just flip the signs of all the charges and note that this does not change the set of solutions. If the sum of magnitudes is zero, then an equilibrium point is not guaranteed to exist. Consider for example a setting with two charges, one of magnitude $+1$ and the other of magnitude -1.

[8] The force fields will usually not exactly agree on the boundaries that are "glued" together. However, they will point in the same general direction (when viewed as tangent fields) and thus it is possible to interpolate smoothly between them without introducing any new solutions.

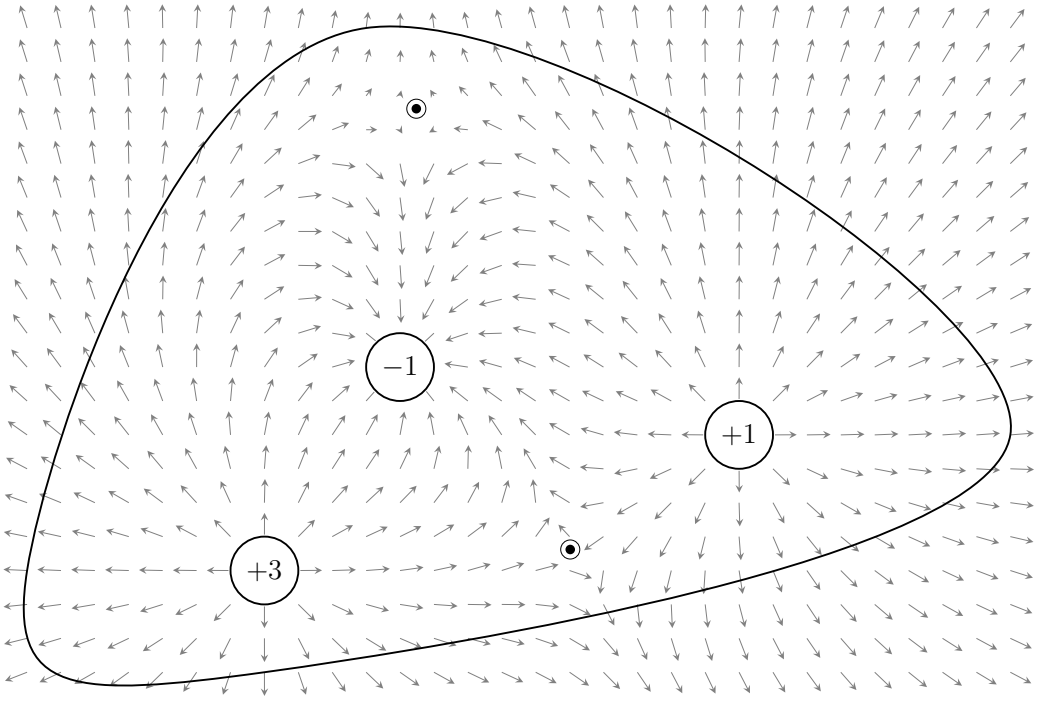


Figure 3 An example with two positive charges (magnitudes +3 and +1) and one negative charge (magnitude −1). The force field is shown, but the length of the vectors is capped so as not to clutter the picture. The two solutions are shown as black dots. The set delimited by the black lines (inside the black outer boundary, and excluding the small round black regions) is an example of a set S that we can use for our proof.

the surface of the torus of genus $|P|$. Now, a generalization of the hairy ball theorem states that such a vector field must have a zero, and by our construction, this zero must be an equilibrium point of the original potential. Furthermore, it is known that the problem of finding a zero of a tangent vector field on such a surface lies in the class PPAD [10].

A generalized potential problem. Motivated by this proof of existence, we ask the question of determining the complexity of computing an equilibrium point of a potential that satisfies certain boundary conditions. Note that our existence proof only used the fact that the potential has a set P of points such that:

- If we move sufficiently far away from the set P, then the force points away from P.
- For any $p \in P$, if we move sufficiently close to p, then the force points either away from p, or towards p.

What is the complexity of finding an equilibrium point of a potential that satisfies these conditions? The problem lies in PPAD by our existence proof. We provide a lower bound by showing that the problem is CLS-hard.

In more detail, our hard instance is two-dimensional and has $|P| = 2$. The potential is obtained by taking the sum of two potentials. Each of those two potentials has a singularity at one of the two points in P and decreases monotonically as we move away from that point. By a careful construction of these two potentials (which are represented as arithmetic circuits), we are able to embed an instance of a CLS-hard problem, ensuring that any equilibrium point yields a solution to this hard problem. We reduce from the problem of finding a

Karush-Kuhn-Tucker (KKT) point in a two-dimensional min-max optimization problem. This problem is known to be CLS-hard [13]. Interestingly, the best upper bound for that problem is also PPAD and determining its exact complexity is a major open problem. We leave open whether a reduction can also be obtained in the other direction, and more generally the question of determining the exact complexity of the problem.

1.2 Further Related Work

Computing equilibrium points of the electrostatic potential is arguably one of the most basic computational problems in electrostatics. Previous papers propose algebraic methods to solve the problem, based on general-purpose solvers whose worst-case running time is exponential in the input size [18, 21]. Computing an ε-approximation in $\mathsf{poly}(1/\varepsilon)$ time is easy in constant dimensions by creating a uniform grid, but to the best of our knowledge, there are no papers that find such a point in $\mathsf{poly}(\log(1/\varepsilon))$ time. Our problem is also related to molecular dynamics [7]. In molecular dynamics, a large number of molecules interact with each other via intermolecular forces, including the electrostatic force, and the goal is to understand their structure and long-term dynamical behavior. Our problem is a much simplified version of this setting, restricted to the electrostatic force and to computing an equilibrium configuration with many fixed and one movable charge.

Historically, the emphasis has been on bounding the number of equilibrium points of the potential induced by a set of charges P (in terms of the cardinality of P and the dimension of space d) as opposed to their computation. Our results do not have new implications for the number of equilibrium points, but we hope for new insights arising from the study of their computation. Maxwell [17] conjectured that in three dimensions ($d = 3$), the number of equilibrium points is upper-bounded by $(n-1)(n-2)$; the appendix of Gabrielov et al. [8] provides a scholarly discussion of the origin of that conjecture and notes that it remains unproven. Using Milnor's theorem, Zolotov [25] obtained upper bounds on the number of equilibrium points for the following larger class of problems. Let n be the number of point charges, each charge i having some weight $q_i \in \mathbb{R} \setminus \{0\}$. The contribution to the potential V from any point obeys a power law, contributing $q_i/{r_i}^m$, where r_i is the distance to the point i. The "real life" power law would set $m = 1$, but in principle, other values can be considered; the domain of interest is restricted to a subspace, or a superspace, of the one spanned by the n fixed charges. If m is even, Zolotov's upper bound is exponential in d, but if m is odd (which includes the "real life" scenario), the upper bound has an exponent of $d + n$. This improves an earlier upper bound due to [8]. A line of work obtains sharper bounds for two to four dimensions ($2 \leq d \leq 4$) [14, 15, 9, 22].

Electric fields (in high dimension) are the basis of the *Poisson Flow generative model* [23] in the context of generative models in machine learning. Data points (such as images) lie in high-dimensional space and are treated as positively charged particles in space augmented with an additional dimension, denoted z. Assuming data points lie on the $z = 0$ hyperplane, a notional positively charged point with $z > 0$ tends to be repelled in a direction of increasing z. As the distance r from the data points increases, the distribution of directions tends to become uniform. This suggests a generative model based on: starting from a uniformly sampled point on a hemisphere with large r, follow in reverse the path taken. A further paper [24] extends this approach to a multi-dimensional augmentation of data space: data points sit in a $\boldsymbol{z} = \mathbf{0}$ subspace, for a vector $\boldsymbol{z}$ of additional components. These models are incorporated into a broader class of physics-based generative models in [16].

2 Notation and Preliminaries

Let $[n] = \{1, 2, \ldots, n\}$ for any positive integer $n \in \mathbb{Z}_{>0}$. Let $\mathbb{1}(\cdot)$ denote the indicator function.

Partial Derivatives. For a smooth (repeatedly differentiable) function $f : X \to \mathbb{R}$, where $X \subseteq \mathbb{R}^d$, we denote the partial derivatives using a multi-index notation as follows: Let $\boldsymbol{s} \in \mathbb{Z}^d_{\geq 0}$ be a d-dimensional multi-index, i.e., a d-tuple with non-negative integers. Let $|\boldsymbol{s}| = ||\boldsymbol{s}||_1 = \sum_{j\in[d]} s_j$ be the sum of the elements in $\boldsymbol{s}$ and $\boldsymbol{s}! = \prod_{j\in[d]}(s_j!)$ be the product of the factorials of the elements in $\boldsymbol{s}$. For an $\boldsymbol{x} \in \mathbb{R}^d$, let $\boldsymbol{x}^{\boldsymbol{s}} = \prod_{j\in[d]} x_j^{s_j}$. Let $D^{\boldsymbol{s}}$ denote the following partial derivative $D^{\boldsymbol{s}} = \prod_{j\in[d]} \left(\frac{\partial}{\partial x_j}\right)^{s_j}$, i.e., $D^{\boldsymbol{s}}$ partially differentiates a function with respect to the first variable s_1 times, the second variable s_2 times, and so on. Let $M_f^{(k)}(\boldsymbol{x})$ denote the maximum absolute value among all k-th order partial derivatives of f at $\boldsymbol{x} \in X$, i.e.,

$$M_f^{(k)}(\boldsymbol{x}) = \max_{|\boldsymbol{s}|=k} |(D^{\boldsymbol{s}} f)(\boldsymbol{x})|.$$

The gradient of f is denoted by $\nabla f = (\frac{\partial f}{\partial x_j})_{j\in[d]}$.

Stationary Points. We define weak and strong stationary points of a function f in a domain X.

▶ **Definition 2.1** (Weak Approximate Stationary Point). *A point $\boldsymbol{x} \in X$ is a weak ε-approximate stationary point, or simply a weak ε-stationary point, for $\varepsilon > 0$ if*

$$||\nabla f(\boldsymbol{x})||_\infty = M_f^{(1)}(\boldsymbol{x}) \leq \varepsilon.$$

▶ **Definition 2.2** (Strong Approximate Stationary Point). *A point $\boldsymbol{x} \in X$ is a strong ε-approximate stationary point, or simply a strong ε-stationary point, for $\varepsilon > 0$ if there exists $\boldsymbol{x}^* \in X$ such that*

$$\nabla f(\boldsymbol{x}^*) = 0 \quad \textit{and} \quad ||\boldsymbol{x} - \boldsymbol{x}^*||_\infty \leq \varepsilon.$$

Taylor Expansion. The Taylor expansion of f at a point $\boldsymbol{x}$, with a k-th order remainder term, with respect to a point $\widehat{\boldsymbol{x}} \in X$, assuming $\{\widehat{\boldsymbol{x}} + t(\boldsymbol{x} - \widehat{\boldsymbol{x}}) \mid t \in [0, 1]\} \subseteq X$, is

$$f(\boldsymbol{x}) = \sum_{|\boldsymbol{s}|<k} \frac{(D^{\boldsymbol{s}} f)(\widehat{\boldsymbol{x}})}{\boldsymbol{s}!}(\boldsymbol{x} - \widehat{\boldsymbol{x}})^{\boldsymbol{s}} + \sum_{|\boldsymbol{s}|=k} R_{\boldsymbol{s}}(\boldsymbol{x})(\boldsymbol{x} - \widehat{\boldsymbol{x}})^{\boldsymbol{s}}, \tag{1}$$

where $R_{\boldsymbol{s}}(\boldsymbol{x})$ is the remainder. There are several equivalent formulations of $R_{\boldsymbol{s}}(\boldsymbol{x})$; we are primarily interested in bounding it and will make use of the following inequality based on Lagrange's formulation

$$|R_{\boldsymbol{s}}(\boldsymbol{x})| \leq \frac{1}{\boldsymbol{s}!} \max_{t\in[0,1]} \max_{|\boldsymbol{s}'|=|\boldsymbol{s}|} |(D^{\boldsymbol{s}'} f)(\widehat{\boldsymbol{x}} + t(\boldsymbol{x} - \widehat{\boldsymbol{x}}))| = \frac{1}{\boldsymbol{s}!} \max_{t\in[0,1]} M_f^{(|\boldsymbol{s}|)}(\widehat{\boldsymbol{x}} + t(\boldsymbol{x} - \widehat{\boldsymbol{x}})). \tag{2}$$

The Electrostatic Potential (Equilibrium Points)

We have a configuration of n charges in the d-dimensional Euclidean space: the i-th charge, $i \in [n]$, has a strength of $q_i \in \mathbb{R}_{\neq 0}$ and is located at $\boldsymbol{a}_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,d}) \in \mathbb{R}^d$. The electrostatic potential at a point $\boldsymbol{x} \in \mathbb{R}^d$ due to this configuration of charges is

$$f(\boldsymbol{x}) = \sum_{i\in[n]} \frac{q_i}{||\boldsymbol{x} - \boldsymbol{a}_i||}, \tag{3}$$

where $||\boldsymbol{x} - \boldsymbol{a}_i|| = \sqrt{\sum_{j\in[d]}(x_j - a_{i,j})^2}$ is the Euclidean norm. A point $\boldsymbol{x}$ is an equilibrium point (or a critical point) if $\nabla f(\boldsymbol{x}) = 0$ and $\boldsymbol{x} \neq \boldsymbol{a}_i$ for all $i \in [n]$; essentially, $\boldsymbol{x}$ is a stationary point of f, except that it also avoids the points of singularity $(\boldsymbol{a}_i)_{i\in[n]}$. With a physics interpretation, an equilibrium point is where the force $-\nabla f(\boldsymbol{x}) = 0$. The ℓ-th component of $\nabla f(\boldsymbol{x})$ is

$$\frac{\partial f}{\partial x_\ell}(\boldsymbol{x}) = \sum_{i\in[n]} \frac{-q_i(x_\ell - a_{i,\ell})}{(\sum_{j\in[d]}(x_j - a_{i,j})^2)^{3/2}}, \qquad \text{for all } \ell \in [d], \tag{4}$$

where the i-th charge contributes $\frac{-q_i(x_\ell - a_{i,\ell})}{(\sum_{j\in[d]}(x_j - a_{i,j})^2)^{3/2}}$ to $\nabla f(\boldsymbol{x})$. Note that the case of $d = 3$ corresponds to the real-life electrostatic (Coulomb) force in the 3-dimensional physical space.

We make the following normalizations without loss of generality: let $\min_{i\in[n]} |q_i| = 1$ and let $\min_{i,i'\in[n],i\neq i'} \max_{j\in[d]} |a_{i,j} - a_{i',j}| = 1$. Essentially, we scale the charges to have the minimum strength of the charges to be 1 and scale the distances to have the minimum distance among the charges for at least one coordinate to be 1, which also implies that the minimum distance among the charges is at least 1. Let $q_{max} = \max_i |q_i|$ and $a_{max} = \max_{i,i'\in[n],j\in[d]} |a_{i,j} - a_{i',j}|$.

3 Computing Approximate Solutions in Constant Dimension

In this section, we prove that we can efficiently compute approximate stationary points of the electrostatic potential, as long as the dimension d is constant.

▶ **Theorem 3.1.** *Fix any $d \geq 1$. Given n charges in d-dimensional space, and a bounded convex set X (given as a system of linear inequalities), as well as error parameters $\varepsilon > \delta > 0$, we can output in polynomial time*

- *either a point $\boldsymbol{x} \in X$ with $\|\nabla f(\boldsymbol{x})\|_\infty \leq \varepsilon$,*
- *or that there is no point $\boldsymbol{x} \in X$ such that $\|\nabla f(\boldsymbol{x})\|_\infty \leq \delta$.*

Here f denotes the electrostatic potential created by the charges.

Our proof proceeds in two steps.

Step 1: We show that, in constant dimension, we can efficiently solve systems of inequalities involving *well-behaved* functions, a notion which we introduce. The main idea is to use the well-behavedness property to subdivide the domain into a polynomial number of regions, and then to apply a Taylor approximation in each region. The regions need to be chosen very carefully, and their sizes vary exponentially.

Step 2: We show that the electrostatic potential is well-behaved, as defined in step 1. At a very high level, the regions where some derivative is large can be determined efficiently and their size can be controlled.

3.1 Step 1: Well-behaved Functions

In this section, we formally define the notion of well-behaved functions that we use, and we state and prove our result about solving systems of inequalities involving such functions, in constant dimension.

We study functions of type $f : X \to \mathbb{R}$ defined on compact domains $X \subseteq \mathbb{R}^d$, where X is defined using a system of linear inequalities. We assume the functions to be *smooth*, i.e., ∞-continuously differentiable in a suitably large open set that contains X. Furthermore, we assume the functions to be *well-behaved*, as defined below.

For a rational number x, let $\mathsf{size}(x)$ denote the bit-length of the representation of x as an irreducible fraction, where the numerator and denominator are represented in binary. Similarly, for a rational vector $\boldsymbol{x} \in \mathbb{R}^d$, we let $\mathsf{size}(\boldsymbol{x})$ be the length of its representation, where each entry is represented as above. We use $a = \mathsf{poly}(b_1, \ldots, b_n)$ to denote that there exist $k, C \in \mathbb{N}$ such that $a \leq C \cdot b_1^k \cdots b_n^k$.

▶ **Definition 3.2.** *A family $\mathcal{F}$ of efficiently representable smooth functions in d-dimensional space is a mapping that assigns to every bitstring $r \in \{0,1\}^*$ a corresponding smooth function $f_r : X_r \to \mathbb{R}$, where $X_r \subseteq \mathbb{R}^d$ is bounded. Furthermore:*

- *There is a polynomial-time algorithm that, given $r \in \{0,1\}^*$ as input, outputs the bounded domain $X_r \subseteq \mathbb{R}^d$ of the function f_r as a system of linear inequalities.*
- *There is a polynomial-time algorithm that, given as input the representation $r \in \{0,1\}^*$, a rational point $\boldsymbol{x} \in X_r$, a multi-index $\boldsymbol{s}$, and $\varepsilon > 0$, outputs $v \in \mathbb{R}$ such that $|(D^{\boldsymbol{s}} f_r)(\boldsymbol{x}) - v| \leq \varepsilon$. To be precise, the algorithm runs in polynomial time in $|r|, \mathsf{size}(\boldsymbol{x}), |\boldsymbol{s}|$, and $\log(1/\varepsilon)$.*

▶ **Definition 3.3** (Well-Behaved)**.** *A family $\mathcal{F} = (f_r)_r$ of efficiently representable smooth functions in d-dimensional space is* well-behaved *if, for every $r \in \{0,1\}^*$, there exist integers $B = \mathsf{poly}(|r|)$ and $C = \mathsf{poly}(|r|)$, such that, for every $\beta > 0$, there exist $n_\beta \in \mathbb{Z}_{\geq 0}$ and $((L_{\beta,i,j}, H_{\beta,i,j})_{j\in[d]})_{i\in[n_\beta]} \in \mathbb{R}^{2n_\beta d}$ such that*

$$n_\beta = \mathsf{poly}(|r|, \mathsf{size}(\beta)),$$

$$\left\{ \boldsymbol{x} \in X_r \;\middle|\; M_{f_r}^{(k)}(\boldsymbol{x}) > C^k 2^B \frac{k!}{\beta^k} \text{ for some } k \in \mathbb{Z}_{\geq 0} \right\} \subseteq \bigcup_{i\in[n_\beta]} \left(\bigtimes_{j\in[d]} [L_{\beta,i,j}, H_{\beta,i,j}] \right),$$

$$|H_{\beta,i,j} - L_{\beta,i,j}| \leq \beta, \qquad \text{for all } i \in [n_\beta], j \in [d].$$

Additionally, we assume that $M_{f_r}^{(k)}(\boldsymbol{x}) \leq C^k 2^B \frac{k!}{\beta_{min}^k}$ for all $\boldsymbol{x} \in X_r$ and $k \in \mathbb{Z}_{\geq 0}$, for some value $\beta_{min} > 0$ with $\mathsf{size}(\beta_{min}) = \mathsf{poly}(|r|)$. Furthermore, given r and β, we can compute n_β and $((L_{\beta,i,j}, H_{\beta,i,j})_{j\in[d]})_{i\in[n_\beta]}$ in time polynomial in $|r|$ and $\mathsf{size}(\beta)$.

Our main result in this section is the following.

▶ **Theorem 3.4.** *Fix any $d \geq 1$. For any well-behaved family $\mathcal{F}$ of efficiently representable smooth functions in d-dimensional space, we can solve the following problem in polynomial time. Given functions $f_1, \ldots, f_m \in \mathcal{F}$ (given by their representations $r_1, \ldots, r_m$), an error parameter $\varepsilon > 0$, and a bounded domain $X \subset \mathbb{R}^d$ (given as a system of linear inequalities) that satisfies $X \subseteq \cap_{i=1}^m X_{r_i}$, output:*

- *either a point $\boldsymbol{x} \in X$ that satisfies $f_\ell(\boldsymbol{x}) \leq 0$ for all $\ell \in [m]$,*
- *or that there is no point $\boldsymbol{x} \in X$ that satisfies $f_\ell(\boldsymbol{x}) \leq -\varepsilon$ for all $\ell \in [m]$.*

Before providing the proof of this theorem, we give a brief proof sketch. The starting point of our algorithm is to use Taylor expansions to approximate the functions $f_1, \ldots, f_m$. Assuming that we could do this, we could then use existing tools to solve systems of polynomial inequalities in constant dimension [11, 20].

The challenge comes from the fact that the Taylor expansion is only a good approximation of the function close to the point around which it is created. The notion of well-behavedness that we define can be used to ensure that the domain can be subdivided into a polynomial number of regions, such that in each of them there is a good Taylor approximation. These regions are not a nice regular grid, but instead the grid is finer where the functions f_ℓ change rapidly (derivatives are large) and coarser where the functions f_ℓ change slowly.

Proof of Theorem 3.4. Fix the dimension $d \geq 1$ and let $\mathcal{F} = (f_r)_r$ be any well-behaved family of efficiently representable smooth functions in d-dimensional space. Recall that we are given functions $f_1, \ldots, f_m \in \mathcal{F}$ (given by their representations $r_1, \ldots, r_m$), an error parameter $\varepsilon > 0$, and a bounded domain $X \subset \mathbb{R}^d$ (given as a system of linear inequalities) that satisfies $X \subseteq \cap_{i=1}^m X_{r_i}$.

Our goal is to provide an algorithm that runs in time polynomial in m, $|r_1|, \ldots, |r_m|$, $\log(1/\varepsilon)$, and the size of the representation of X and which outputs:

- either a point $\boldsymbol{x} \in X$ that satisfies $f_\ell(\boldsymbol{x}) \leq 0$ for all $\ell \in [m]$,
- or that there is no point $\boldsymbol{x} \in X$ that satisfies $f_\ell(\boldsymbol{x}) \leq -\varepsilon$ for all $\ell \in [m]$.

Note that the algorithm can take super-polynomial time in d, as we are assuming d to be a constant.

Our algorithm uses a piecewise Taylor expansion. We divide the search space into an axis-aligned grid. For each cell in the grid, we construct m polynomials that approximate the m functions $(f_\ell)_{\ell \in [m]}$. We then solve a system of polynomial inequalities for each cell in the grid to find an approximate solution, if any. The grid is finer where the functions f_ℓ change rapidly (derivatives are large) and coarser where the functions f_ℓ change slowly. This ensures that our polynomial approximation has a small error. We use the well-behavedness property of the family $\mathcal{F}$ to give a polynomial bound on the number of required grid cells.

Grid. We construct the grid using a set of axis-aligned hyperplanes. Fix an $\ell \in [m]$ and focus on the function f_ℓ; we repeat the same process for each $\ell \in [m]$. Note that f_ℓ comes from the family $\mathcal{F}$ of well-behaved functions. We can thus compute the parameters B_ℓ, C_ℓ, and $\beta_{\ell,min}$ associated with f_ℓ as introduced in Definition 3.3. Furthermore, we also compute, for each $j \in [d]$

$$L_j = \min_{\boldsymbol{x} \in X} x_j, \quad H_j = \max_{\boldsymbol{x} \in X} x_j$$

and $t_{\ell,max}$, which is defined as the smallest $t \geq 1$ such that

$$\beta_{\ell,min} 2^t \geq \max_{j \in [d]} |H_j - L_j|.$$

Finally, we let $\beta_{\ell,max} = \beta_{\ell,min} 2^{t_{\ell,max}}$. Since X is bounded and given as a system of linear inequalities, we can compute these quantities in polynomial time. Furthermore, $t_{\ell,max}$ is an integer bounded by a polynomial.

For $t = 0$ to $t_{\ell,max}$:

- Let $\beta = \beta_{\ell,min} 2^t$.
- Compute the $n_{\ell,\beta}$ hypercuboids $((L_{\ell,\beta,i,j}, H_{\ell,\beta,i,j})_{j \in [d]})_{i \in [n_{\ell,\beta}]}$ satisfying the conditions set out in Definition 3.3. Note that by the well-behavedness of $\mathcal{F}$, these are guaranteed to exist and can be computed efficiently.
- For every such hypercuboid $(L_{\ell,\beta,i,j}, H_{\ell,\beta,i,j})_{j \in [d]}$, where $i \in [n_{\ell,\beta}]$, add the following hyperplanes:

$$\left\{ \boldsymbol{x} \in \mathbb{R}^d \mid x_j = L_{\ell,\beta,i,j} + s \frac{H_{\ell,\beta,i,j} - L_{\ell,\beta,i,j}}{4C_\ell d} \right\}, \text{ for } j \in [d] \text{ and } s \in \{0\} \cup [4C_\ell d].$$

In other words, we split each hypercuboid into $4C_\ell d$ uniform pieces along every axis. We also add the hyperplanes $\{\boldsymbol{x} \in \mathbb{R}^d \mid x_j = L_j\}$ and $\{\boldsymbol{x} \in \mathbb{R}^d \mid x_j = H_j\}$ for all $j \in [d]$, and also split each of those hypercuboids into $4C_\ell d$ uniform pieces along every axis.

Let us count the total number of axis-aligned hyperplanes we added along each axis $j \in [d]$. We added two hyperplanes $x_j = L_j$ and $x_j = H_j$ at the ends, as well as $4C_\ell d - 1$ additional hyperplanes evenly spaced between them. For each $\ell \in [m]$, we iterated over

$t_{\ell,max}+1$ values of β; for each β, we iterated over $n_{\ell,\beta}$ hypercuboids; for each hypercuboid, we added $4C_\ell d+1$ hyperplanes. Overall, we have at most

$$\sum_{\ell\in[m]}(1+4C_\ell d)\left(1+\sum_{t=0}^{t_{\ell,max}} n_{\ell,\beta_{\ell,min}2^t}\right),$$

hyperplanes along each axis, which is polynomial in the input size.

Polynomial Approximation. The set of hyperplanes specified above divides the $\bigtimes_{j\in[d]}[L_j,H_j]$ hypercuboid that covers X into a grid. For each cell in the grid, where a cell is a minimal hypercuboid created by these hyperplanes that does not contain another hypercuboid, we can find an approximate solution, if it exists, by Taylor-approximating each function f_ℓ for $\ell\in[m]$.

Fix a cell in the grid – denoted by $C=\bigtimes_{j\in[d]}[L'_j,H'_j]$ and let $X'=C\cap X$. If X' is empty, then we ignore this cell. Otherwise, pick an arbitrary point $\widehat{\boldsymbol{x}}\in X'$. We approximate f_ℓ using a polynomial of degree $k_\ell-1$, where $k_\ell=\lceil B_\ell+\log(8/\varepsilon)\rceil$. In particular, we compute the $(k_\ell-1)$-th order Taylor expansion of f_ℓ with respect to the point $\widehat{\boldsymbol{x}}$, denoted by $g_\ell(\boldsymbol{x})$

$$g_\ell(\boldsymbol{x})=\sum_{|\boldsymbol{s}|<k_\ell}\frac{(D^{\boldsymbol{s}}f_\ell)(\widehat{\boldsymbol{x}})}{\boldsymbol{s}!}(\boldsymbol{x}-\widehat{\boldsymbol{x}})^{\boldsymbol{s}}. \tag{5}$$

Using the fact that the family of functions is well-behaved, we obtain the following crucial claim, which we prove at the end of this section.

▷ Claim 3.5. The $(k_\ell-1)$-th Taylor approximation of f_ℓ with respect to some arbitrary point $\widehat{\boldsymbol{x}}\in X'$, denoted by g_ℓ, has an error of at most $\varepsilon/8$, i.e.,

$$|f_\ell(\boldsymbol{x})-g_\ell(\boldsymbol{x})|\le\varepsilon/8,\ \text{for all } \ell\in[m] \text{ and } \boldsymbol{x}\in X'.$$

In general, we might not be able to compute the values $(D^{\boldsymbol{s}}f_\ell)(\widehat{\boldsymbol{x}})$ exactly. However, by Definition 3.2, we know that we can approximate these values arbitrarily well. Thus, by computing the coefficients with sufficient precision, we obtain a polynomial $\widehat{g}_\ell$ of degree $k_\ell-1$ that satisfies $|\widehat{g}_\ell(\boldsymbol{x})-g_\ell(\boldsymbol{x})|\le\varepsilon/8$ for all $\boldsymbol{x}\in X'$. As a result, together with the claim, we have that

$$|f_\ell(\boldsymbol{x})-\widehat{g}_\ell(\boldsymbol{x})|\le\varepsilon/4,\ \text{for all } \ell\in[m] \text{ and } \boldsymbol{x}\in X'. \tag{6}$$

Next, we will make use of the following theorem [11] (an algorithm with improved complexity is given in [20]). Note that here it is crucial that the dimension d is constant.

▶ **Theorem 3.6** (e.g. [11]). *Fix $d\ge1$. Given $\delta>0$ and a system of polynomial inequalities in d variables, $p_i(\boldsymbol{x})\le0$ for all $i\in[m]$, we can output in polynomial time*

- *either a point $\boldsymbol{x}\in\mathbb{R}^d$ such that there exists $\boldsymbol{x}^*\in\mathbb{R}^d$ with $\|\boldsymbol{x}-\boldsymbol{x}^*\|\le\delta$ and $p_i(\boldsymbol{x}^*)\le0$ for all $i\in[m]$,*
- *or that there is no point $\boldsymbol{x}\in\mathbb{R}^d$ that satisfies $p_i(\boldsymbol{x})\le0$ for all $i\in[m]$.*

We apply this theorem to the following system of polynomial inequalities

$$\begin{aligned}\widehat{g}_\ell(\boldsymbol{x})+\varepsilon/2\le0,\qquad&\text{for } \ell\in[m],\\ \boldsymbol{x}\in X',&\end{aligned}$$

where the last constraint represents the list of linear inequalities defining the feasible region X'. We set $\delta = \varepsilon/4L$, where L is such that all the $\widehat{g}_\ell$ are L-Lipschitz-continuous over some sufficiently big compact set containing X'.

If the solver outputs that the system is not satisfiable, then, since $|\widehat{g}_\ell(\boldsymbol{x}) - f_\ell(\boldsymbol{x})| \leq \varepsilon/4$ by (6), it follows that there is no $\boldsymbol{x} \in X'$ such that $f_\ell(\boldsymbol{x}) \leq -\varepsilon$ for all $\ell \in [m]$.

If the solver outputs that the system is satisfiable, then it also outputs a point $\boldsymbol{x} \in \mathbb{R}^d$ such that there exists $\boldsymbol{x}^* \in X'$ with $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta$ and $\widehat{g}_\ell(\boldsymbol{x}^*) + \varepsilon/2 \leq 0$ for all $\ell \in [m]$. By replacing x by its projection[9] onto the convex set X', we have that $x \in X'$ and $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta$. As a result, it follows that

$$\widehat{g}_\ell(\boldsymbol{x}) \leq \widehat{g}_\ell(\boldsymbol{x}^*) + L\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq -\varepsilon/2 + L\delta \leq -\varepsilon/2 + \varepsilon/4 \leq -\varepsilon/4.$$

Since $|\widehat{g}_\ell(\boldsymbol{x}) - f_\ell(\boldsymbol{x})| \leq \varepsilon/4$ by (6), we thus have $f_\ell(\boldsymbol{x}) \leq 0$ for all $\ell \in [m]$.

After running the solver on each cell, we either have found a point $\boldsymbol{x} \in X$ with $f_\ell(\boldsymbol{x}) \leq 0$ for all $\ell \in [m]$, or, we can confidently output that there exists no $\boldsymbol{x} \in X$ such that $f_\ell(\boldsymbol{x}) \leq -\varepsilon$ for all $\ell \in [m]$.

Running Time. Since the number of axis-aligned hyperplanes along each axis is polynomial, and since the dimension d is constant, the number of cells is polynomial. For each such cell we run the polynomial inequality solver, which runs in polynomial time, because the dimension d is constant. Thus, the total running time of the algorithm is polynomial. ◀

Proof of Claim 3.5. Let us fix an $\ell \in [m]$ and focus on f_ℓ; the same argument applies for every ℓ. We suppress the notation ℓ to decrease clutter.

From (1) and (2), we know that for any $\boldsymbol{x} \in X'$ the error is bounded by

$$\begin{aligned}
|f(\boldsymbol{x}) - g(\boldsymbol{x})| &\leq \left| \sum_{|\boldsymbol{s}|=k} R_{\boldsymbol{s}}(\boldsymbol{x})(\boldsymbol{x} - \widehat{\boldsymbol{x}})^{\boldsymbol{s}} \right| \leq \sum_{|\boldsymbol{s}|=k} |R_{\boldsymbol{s}}(\boldsymbol{x})||\boldsymbol{x} - \widehat{\boldsymbol{x}}|^{\boldsymbol{s}} \\
&\leq \sum_{|\boldsymbol{s}|=k} \frac{1}{\boldsymbol{s}!} \left(\max_{t \in [0,1]} M_f^{(|\boldsymbol{s}|)}(\widehat{\boldsymbol{x}} + t(\boldsymbol{x} - \widehat{\boldsymbol{x}})) \right) |\boldsymbol{x} - \widehat{\boldsymbol{x}}|^{\boldsymbol{s}} \\
&\leq \max_{t \in [0,1]} M_f^{(k)}(\widehat{\boldsymbol{x}} + t(\boldsymbol{x} - \widehat{\boldsymbol{x}})) \sum_{|\boldsymbol{s}|=k} \frac{1}{\boldsymbol{s}!} |\boldsymbol{x} - \widehat{\boldsymbol{x}}|^{\boldsymbol{s}}.
\end{aligned}$$

Let $\alpha = \max_{\boldsymbol{y} \in X'} M_f^{(k)}(\boldsymbol{y})$. Since the family of functions is well-behaved (Definition 3.3), we have that $M_f^{(k)}(\boldsymbol{x}) \leq C^k 2^B \frac{k!}{\beta_{min}^k}$ for all $\boldsymbol{x} \in X$. So it must be that $\alpha \leq C^k 2^B \frac{k!}{\beta_{min}^k}$. In the algorithm, β is incremented by a factor of 2, starting from β_{min} and going to β_{max}. Let β' be the smallest value among the set of β-values used in the algorithm such that $\alpha > C^k 2^B \frac{k!}{\beta'^k}$. If there is no such value, then it must be that $\alpha \leq C^k 2^B \frac{k!}{\beta_{max}^k}$, and we set $\beta' = \beta_{max}$. Notice that by the definition of β', we have

$$\alpha = \max_{\boldsymbol{y} \in X'} M_f^{(k)}(\boldsymbol{y}) \leq C^k 2^B \frac{k!}{(\beta'/2)^k}. \tag{7}$$

In the case where $\alpha > C^k 2^B \frac{k!}{\beta_{max}^k}$, we have $\alpha > C^k 2^B \frac{k!}{\beta'^k}$. But this means that a point inside the cell X' (namely the one where the value α is achieved) must be covered by a hypercuboid corresponding to β'. Since X' is a cell, all of X' must thus be covered by that hypercuboid,

[9] The projection can be computed in polynomial time, since X' is given as a system of linear inequalities.

which has sides of length at most β'. Further, in the algorithm, this hypercuboid was cut into smaller pieces by splitting each side into $4Cd$ parts. Therefore, we must have

$$\max_{j\in[d],\boldsymbol{x}\in X'} |x_j - \widehat{x}_j| \leq \frac{\beta'}{4Cd}. \tag{8}$$

In the case where $\alpha > C^k 2^B \frac{k!}{\beta_{max}^k}$, we have $\beta' = \beta_{max}$, and (8) again holds, since the whole domain X fits within a box of side-length β_{max}, and we further subdivided that box into $4Cd$ parts in a uniform manner.

Furthermore, by the multinomial theorem

$$d^k = (1+1+\ldots+1)^k = \sum_{|\boldsymbol{s}|=k} \frac{k!}{\boldsymbol{s}!} \iff \sum_{|\boldsymbol{s}|=k} \frac{1}{\boldsymbol{s}!} = \frac{d^k}{k!}. \tag{9}$$

Plugging (7), (8), and (9) into the error bound, we get

$$\begin{aligned}
|f(\boldsymbol{x}) - g(\boldsymbol{x})| &\leq \max_{t\in[0,1]} M_f^{(k)}(\widehat{\boldsymbol{x}} + t(\boldsymbol{x}-\widehat{\boldsymbol{x}})) \sum_{|\boldsymbol{s}|=k} \frac{1}{\boldsymbol{s}!} |\boldsymbol{x}-\widehat{\boldsymbol{x}}|^{\boldsymbol{s}} \\
&\leq \max_{\boldsymbol{y}\in X'} M_f^{(k)}(\boldsymbol{y}) \sum_{|\boldsymbol{s}|=k} \frac{1}{\boldsymbol{s}!} \left(\frac{\beta'}{4Cd}\right)^k \\
&\leq C^k 2^B \frac{k!}{(\beta'/2)^k} \frac{d^k}{k!} \left(\frac{\beta'}{4Cd}\right)^k = \frac{2^B}{2^k} \leq \varepsilon/8,
\end{aligned}$$

where the last inequality holds because $k = \lceil B + \log(8/\varepsilon)\rceil \geq B + \log(8/\varepsilon)$. ◁

3.2 Step 2: Application to Electrostatic Potentials

In this section, we apply the tools introduced in step 1 to the setting of electrostatic potentials. For this, we need to show that the potential f is well-behaved, as defined in the previous section. However, notice that the function f and its derivatives are unbounded near the charges; in particular

$$\left|\frac{\partial f}{\partial x_\ell}(\boldsymbol{x})\right| = \left|\sum_{i\in[n]} \frac{-q_i(x_\ell - a_{i,\ell})}{(\sum_{j\in[d]}(x_j - a_{i,j})^2)^{3/2}}\right| \to \infty \text{ as } \boldsymbol{x} \to \boldsymbol{a}_i \text{ for any } i \in [n].$$

This means that the function is not well-behaved in its entire domain. However, we observe that a small region near each charge, where the derivatives become unbounded, never has an equilibrium point. The following lemma is proved in the full version of the paper.

▶ **Lemma 3.7.** *For every $i \in [n]$, there is no ε-equilibrium point in the hypercube $\bigtimes_{j\in[d]}[a_{i,j} - \rho, a_{i,j} + \rho]$ around the charge q_i located at $\boldsymbol{a}_i$, where $\rho = \frac{1}{d\sqrt{d}(4nq_{max}+\varepsilon)}$.*

Given Lemma 3.7, we can safely avoid searching for an equilibrium point inside the hypercube centered around each charge of side length 2ρ. The region left after cutting out these hypercubes from the domain X is not a convex set, which means that we cannot immediately apply our Taylor approximation algorithm (Theorem 3.4). We can easily resolve this by splitting this non-convex set into a polynomial number of convex sets by creating a grid as follows: Along each axis $j \in [d]$ and around each charge $i \in [n]$, add the hyperplanes $a_{i,j} \pm \rho$. Overall, along each axis, we have introduced at most $2n$ hyperplanes. These $2n$ hyperplanes create $2n+1$ segments. So, we have at most $(2n+1)^d$ cells in the grid, which

are polynomially many because d is assumed to be a constant. Our convex sets are simply the intersections of these cells with X, where we ignore the cells within ρ distance from each charge. Since X is bounded, each of those cells intersected with X will also be bounded.

Given this, it now remains to prove that the electrostatic potential is well-behaved in each such bounded set that does not contain the regions close to the charges. It suffices to prove this for the potential induced by one charge (since, as shown in the full version, we can take linear combinations of well-behaved functions to extend it to multiple charges). Namely, we need the following lemma, which is proved in the full version.

▶ **Lemma 3.8.** *Fix $d \geq 1$. The family of electrostatic potentials induced by a single charge*

$$\mathcal{F} = \left\{ f : X' \to \mathbb{R}, \boldsymbol{x} \mapsto \frac{q}{||\boldsymbol{x} - \boldsymbol{a}||_2} : q \in \mathbb{Q}_{\neq 0}, \boldsymbol{a} \in \mathbb{Q}^d, X' \subset \mathbb{R}^d \text{ bounded with } \boldsymbol{a} \notin X' \right\}$$

is well-behaved.[10]

As mentioned above, after eliminating the regions close to the singularities, where solutions cannot occur, we obtain a polynomial number of convex domains. Let X' denote any one of them. Given error parameters $\varepsilon > \delta > 0$, we write down the system of equations

$$\begin{aligned} \frac{\partial f}{\partial x_j}(\boldsymbol{x}) &\leq \varepsilon, && \text{for } j \in [d], \\ -\frac{\partial f}{\partial x_j}(\boldsymbol{x}) &\leq \varepsilon, && \text{for } j \in [d], \\ \boldsymbol{x} &\in X', \end{aligned}$$

As f is well-behaved, using the tools in the appendix of the full version (derivatives of well-behaved functions), we know that $\frac{\partial f}{\partial x_j}$ is also well-behaved for all $j \in [d]$. So, by using the algorithm presented in the previous section (Theorem 3.4) with error parameter $\varepsilon - \delta$, we obtain in polynomial time

- either a point $\boldsymbol{x} \in X'$ such that $\|\nabla f(\boldsymbol{x})\| \leq \varepsilon$,
- or the guarantee that there is no point $\boldsymbol{x} \in X'$ with $\|\nabla f(\boldsymbol{x})\| \leq \delta$.

By performing this for each of the polynomially many sets X', we obtain the desired output. Namely, either we find an ε-stationary point in X, or we can guarantee that there is no δ-stationary point in X.

4 Computing Strong Approximate Solutions

In Section 3, we presented an algorithm for computing approximate stationary points of an electrostatic potential. However, these points are not guaranteed to lie close to exact solutions. In this section, we show how to compute *strong* approximate stationary points, namely points that are guaranteed to lie close to exact solutions. Our algorithm requires an additional assumption, that we call *strong non-degeneracy*.

A point $\boldsymbol{x}$ is a degenerate stationary point of f if $(\nabla f)(\boldsymbol{x}) = 0$ and $(\nabla^2 f)(\boldsymbol{x}) = 0$, i.e., both the gradient and the Hessian of f are 0 at $\boldsymbol{x}$.[11] On the other hand, $\boldsymbol{x}$ is a non-degenerate stationary point if $(\nabla f)(\boldsymbol{x}) = 0$ but $(\nabla^2 f)(\boldsymbol{x}) \neq 0$. A function f is non-degenerate if all its stationary points are non-degenerate. Most *generic* functions are non-degenerate, i.e., parameterized functions are non-degenerate for *almost all* instantiations of the parameters

[10] As before, we assume that X' is a bounded polytope given as a system of linear inequalities.

[11] Note that ∇^2 is the Hessian $\nabla \times \nabla$ operator and not the Laplacian $\nabla \cdot \nabla$ operator. The Laplacian of the real-life electrostatic potential is 0 everywhere, but the Hessian is not (for non-trivial instances).

(degenerate only for a measure 0 subspace of the parameters). For example, among all degree-k polynomials, parameterized by the coefficients of the monomials, the polynomials that are degenerate correspond to a subset of the coefficients of measure 0. Another example, the electrostatic potential function in (3) is degenerate only for a measure 0 subspace of the possible locations and strengths of the charged particles. In other words, if we slightly perturb (even the location or strength of only one of) the charges, the electrostatic potential will *almost surely* be non-degenerate [19, Chapter 6, 32].

We assume a stronger form of non-degeneracy in our analysis: We call a function f δ-strongly non-degenerate in a domain X if there exists a stationary point $\boldsymbol{x} \in X$ where the determinant of the Hessian at $\boldsymbol{x}$ has magnitude at least $\delta > 0$, i.e., $(\nabla f)(\boldsymbol{x}) = 0$ and $|\det((\nabla^2 f)(\boldsymbol{x}))| \geq \delta$. With this assumption, we provide an algorithm to compute a strong ε-stationary point of f that runs in time polynomial in $\log(1/\varepsilon)$ and $\log(1/\delta)$, so we could set ε and δ exponentially small.

▶ **Theorem 4.1.** *Fix any $d \geq 1$ and any well-behaved family $\mathcal{F}$ of efficiently representable smooth functions in d-dimensional space. There exists an algorithm that takes as input*

- *a function $f \in \mathcal{F}$ (given by its representation r, and defined on domain $X_r \subset \mathbb{R}^d$),*
- *an error parameter $\varepsilon > 0$,*
- *and a bounded domain $X \subset \mathbb{R}^d$ (given as a system of linear inequalities) with $X \subseteq X_r$,*

and which has the following guarantee. Whenever there exist $\boldsymbol{x}^* \in X$ and $\delta > 0$ such that $(\nabla f)(\boldsymbol{x}^*) = 0$ and $|\det((\nabla^2 f)(\boldsymbol{x}^*))| \geq \delta$, the algorithm runs in time polynomial in $|r|$, $\log(1/\varepsilon)$, $\log(1/\delta)$, and the size of the representation of X, and outputs a point $\boldsymbol{x} \in X$ such that there exists $\boldsymbol{x}' \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \varepsilon$ and $\nabla f(\boldsymbol{x}') = 0$.

The rest of this section provides a proof sketch of the theorem. We will assume that we are given $\delta > 0$ that satisfies the condition in the theorem. If we are not given δ, we can run the algorithm that we present here for values of δ decreasing exponentially $1, 1/2, 1/4, 1/8, \ldots$ and obtain the desired result.

As the family $\mathcal{F}$ is well-behaved, $\frac{\partial f}{\partial x_j}$ and $\frac{\partial^2 f}{\partial x_j \partial x_{j'}}$ also satisfy the well-behavedness condition for all $j, j' \in [d]$ (see the appendix of the full version). Further, again as shown in the appendix of the full version, $\det(\nabla^2 f)$ is also well-behaved.

As a result, we can use Theorem 3.4 to solve the following system of inequalities over X, where $\varepsilon' > 0$ is some sufficiently small value that we fix later.

$$\begin{aligned} \frac{\partial f}{\partial x_j}(\boldsymbol{x}) \leq \varepsilon' \quad \text{and} \quad \frac{\partial f}{\partial x_j}(\boldsymbol{x}) \geq -\varepsilon', \qquad & \text{for } j \in [d], \\ \det(\nabla^2 f(\boldsymbol{x})) \geq \delta - \varepsilon' \quad \text{or} \quad \det(\nabla^2 f(\boldsymbol{x})) \leq -\delta + \varepsilon'. & \end{aligned} \tag{10}$$

The "or" condition among the last two inequalities can be implemented by solving the system of inequalities once with the first condition, and once with the second condition. Notice that the point $\boldsymbol{x}^* \in X$ satisfies the above inequalities with at least ε' slack. As a result, using Theorem 3.4 with error parameter ε', we obtain a point $x \in X$ that satisfies the system of inequalities (10). In the full version, we show how to pick ε' such that any point $\boldsymbol{x} \in X$ that satisfies (10) is an ε-strong approximate equilibrium.

5 Computational Complexity of Generalizations

In Section 3, we gave an efficient algorithm for computing stationary points of well-behaved functions in a constant dimensional space. There are two natural generalizations: First, we can consider well-behaved functions in a high-dimensional space. Second, we can restrict to constant dimensions but consider functions that are not well-behaved. We study these two generalizations in this section.

5.1 Well-Behaved Functions in a High-Dimensional Space

In this section we show that our approach cannot be generalized to settings where the dimension of the space is not fixed, but can grow polynomially. Namely, we prove that computing an approximate stationary point of a well-behaved function is NP-hard in higher dimension.

The class of functions that we consider are simply the polynomials of degree 3 over d variables, for all $d \geq 1$. In the appendix of the full version, we show that this class of functions is well-behaved. Ahmadi and Zhang [1] have shown that it is NP-hard to decide whether such a polynomial admits a stationary point over $\mathbb{R}^d$. We use their reduction to show a stronger result, namely that a so-called gap version of the problem remains hard, which implies that our result for fixed dimension cannot be generalized to higher dimension. The proof can be found in the full version.

▶ **Theorem 5.1.** *Given a polynomial $p : \mathbb{R}^d \to \mathbb{R}$ of degree 3, it is NP-hard to distinguish between the following two cases:*

- *p has a stationary point lying in $[-1, 1]^d$,*
- *p has no $1/d^2$-stationary point in $\mathbb{R}^d$.*

5.2 Non-Well-Behaved Functions in a Constant-Dimensional Space

In this section we prove CLS-hardness of computing equilibrium points of a generalized potential function in a two-dimensional space with only two charges. Each charge creates a potential that is broadly similar to the electrostatic potential. In particular, the potential function due to a given charge is continuously differentiable, has a singularity at the location of the charge, and is monotonically decreasing as we move away from the charge. The overall potential due to the two charges is the sum of the potentials due to each charge, as with the electrostatic potential.

▶ **Theorem 5.2.** *Given two locations $a_1, a_2 \in \mathbb{R}^2$ and two potentials[12] $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$ that satisfy for $i \in \{1, 2\}$*

- *f_i has a singularity at location a_i, but is continuously differentiable on $\mathbb{R}^2 \setminus \{a_i\}$,*
- *f_i is monotonically decreasing as we move away from the location a_i,*

it is CLS-hard to compute a weak ε-approximate equilibrium point of $f_1 + f_2$.

We prove our hardness result by a reduction from the 2D-MIN-MAX problem defined below, which is known to be CLS-hard [13].

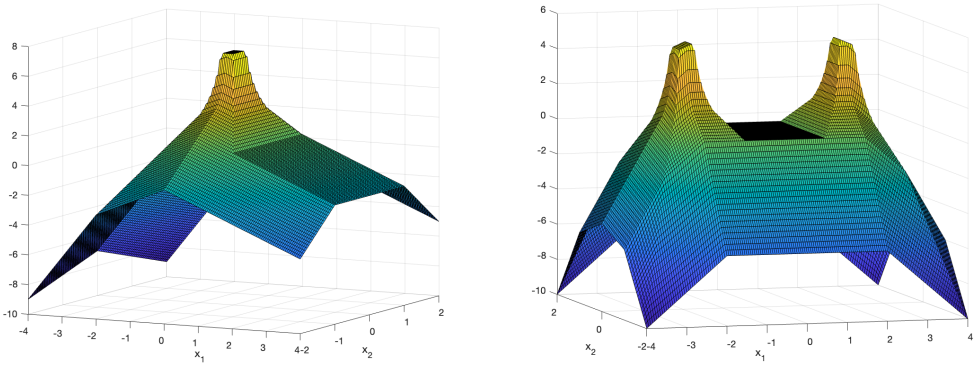
▶ **Definition 5.3.** *In the 2D-MIN-MAX problem we are given a continuously differentiable function[13] $f : [-1, 1]^2 \to \mathbb{R}$, an approximation factor $\varepsilon > 0$, and a value $L > 0$, such that ∇f is L-Lipschitz-continuous over $[-1, 1]^2$. The goal is to find an ε-KKT point of the optimization problem*

$$\min_{x_1 \in [-1,1]} \max_{x_2 \in [-1,1]} f(x_1, x_2)$$

i.e., a point $(x_1, x_2) \in [-1, 1]^2$ that satisfies

- *if $x_1 \neq -1$, then $\frac{\partial f}{\partial x_1} \leq \varepsilon$,*

[12] The functions and their derivatives are given by arithmetic circuits, see [6].
[13] The function and its derivative are again given by arithmetic circuits.



(a) Potential induced by a single charge. This corresponds to function h in the proof.

(b) Potential induced by two charges. This corresponds to the net potential due to the two charges, assuming $\widehat{f} = 0$ in the proof.

Figure 4 Generalized potential function due to two charges (non-smoothed).

- *if* $x_1 \neq 1$, *then* $\frac{\partial f}{\partial x_1} \geq -\varepsilon$,
- *if* $x_2 \neq -1$, *then* $\frac{\partial f}{\partial x_2} \geq -\varepsilon$,
- *if* $x_2 \neq 1$, *then* $\frac{\partial f}{\partial x_2} \leq \varepsilon$.

▶ **Theorem 5.4** ([13]). *The 2D-MIN-MAX problem is* CLS*-hard. Further, this hardness result holds even if we assume that the function does not have an ε-KKT point at the boundary, and additionally $|f| \leq 1$ and $||\nabla f||_2 \leq 1$ over $[-1, 1]^2$.*

The proof of Theorem 5.2 can be found in the full version. Here, we give a brief overview of our approach. We first construct two piecewise linear potentials that are decreasing around their respective charge locations. These are carefully constructed such that, when they are added together, a flat region appears between the two singularities. See Figure 4a for the potential associated to a single charge, and Figure 4b for the sum of the two potentials and the flat region.

Next, we modify one of the potentials very carefully, such as to embed an instance of the 2D-MIN-MAX problem in that flat region. As a result, we argue that any stationary point must correspond to a KKT point of the 2D-MIN-MAX instance. Importantly, we make sure that our two potentials remains monotonically decreasing, despite the modifications. Finally, we apply a box filter to make our two potentials continuously differentiable.

References

1 Amir Ali Ahmadi and Jeffrey Zhang. Complexity aspects of local minima and related notions. *Advances in Mathematics*, 397:108119, 2022. doi:10.1016/j.aim.2021.108119.

2 Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1480–1498, 2015. doi:10.1109/focs.2015.94.

3 Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. Finding a Nash equilibrium is no easier than breaking Fiat-Shamir. In *Proceedings of the 51st ACM Symposium on Theory of Computing (STOC)*, pages 1103–1114, 2019. doi:10.1145/3313276.3316400.

4 Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization. SIAM, 2000. doi:10.1137/1.9780898719857.

5 Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 790–804, 2011. doi:10.1137/1.9781611973082.62.

6 John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: CLS = PPAD ∩ PLS. *Journal of the ACM*, 70(1):7:1–7:74, 2022. doi:10.1145/3568163.

7 Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

8 Andrei Gabrielov, Dmitry Novikov, and Boris Shapiro. Mystery of point charges. *Proceedings of the London Mathematical Society*, 95(2):443–472, 2007.

9 G Giorgadze and G Khimshiashvili. On equilibrium points of three point charges. *Bulletin Georgian National Acad. Sci*, 15(3):7–13, 2021.

10 Paul W. Goldberg and Alexandros Hollender. The hairy ball problem is ppad-complete. *J. Comput. Syst. Sci.*, 122:34–62, 2021. doi:10.1016/J.JCSS.2021.05.004.

11 D. Yu. Grigor'ev and N.N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1):37–64, 1988. doi:10.1016/S0747-7171(88)80005-1.

12 Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. In *Proceedings of the 53rd ACM Symposium on Theory of Computing (STOC)*, pages 708–721, 2021. doi:10.1145/3406325.3451055.

13 Alkis Kalavasis, Panagiotis Patsilinakos, Stratis Skoulakis, and Manolis Zampetakis. The Complexity of 2D Min-max Optimization. Manuscript, 2025.

14 Kenneth Killian. A remark on Maxwell's conjecture for planar charges. *Complex Variables and Elliptic Equations*, 54(12):1073–1078, December 2009.

15 Tsung-Lin Lee and Ya-Lun Tsai. Nine equilibrium points of four point charges on the plane. *Applied Mathematics Letters*, 132:108207, 2022. doi:10.1016/J.AML.2022.108207.

16 Ziming Liu, Di Luo, Yilun Xu, Tommi S. Jaakkola, and Max Tegmark. Genphys: From physical processes to generative models. *arXiv*, abs/2304.02637, 2023. doi:10.48550/arXiv.2304.02637.

17 James Clerk Maxwell. *A Treatise on Electricity and Magnetism, Vol. I*. Macmillan and Co., 1873.

18 Dhagash Mehta. Finding all the stationary points of a potential-energy landscape via numerical polynomial-homotopy-continuation method. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 84(2):025702, 2011.

19 Marston Morse and Stewart S Cairns. *Critical point theory in global analysis and differential topology: an introduction*. Academic Press, 1969.

20 James Renegar. On the computational complexity and geometry of the first-order theory of the reals, part I: introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *J. Symb. Comput.*, 13(3):255–300, 1992. doi:10.1016/S0747-7171(10)80003-3.

21 Alexei Yu Uteshev and Marina V Yashina. Stationary points for the family of fermat–torricelli–coulomb-like potential functions. In *International Workshop on Computer Algebra in Scientific Computing*, pages 412–426. Springer, 2013. doi:10.1007/978-3-319-02297-0_34.

22 Alexei Yu Uteshev and Marina V Yashina. On maxwell's conjecture for coulomb potential generated by point charges. In *Transactions on Computational Science*, pages 68–80. Springer, 2016. doi:10.1007/978-3-662-50412-3_5.

23 Yilun Xu, Ziming Liu, Max Tegmark, and Tommi S. Jaakkola. Poisson flow generative models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/6ad68a54eaa8f9bf6ac698b02ec05048-Abstract-Conference.html.

24 Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, and Tommi S. Jaakkola. PFGM++: unlocking the potential of physics-inspired generative models. In *International Conference on Machine Learning, ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 38566–38591. PMLR, 2023. URL: `https://proceedings.mlr.press/v202/xu23m.html`.

25 Vladimir Zolotov. Upper bounds for the number of isolated critical points via the Thom–Milnor theorem. *Analysis and Mathematical Physics*, 13(5):81, 2023.