A linear-time approximate convex envelope algorithm using the double Legendre–Fenchel transform with application to phase separation

Lorenzo Contento, Alexandre Ern, Rossana Vermiglio

A linear-time approximate convex envelope algorithm using the double Legendre–Fenchel transform with application to phase separation

Lorenzo Contento* Alexandre Ern† Rossana Vermiglio‡

Abstract

We study the double discrete Legendre–Fenchel Transform (LFT) to approximate the convex envelope of a given function. We analyze the convergence of the double discrete LFT in the multivariate case based on previous convergence results for the discrete LFT. We focus our attention on the grid on which the second discrete LFT is computed (dual grid); its choice has great impact on the accuracy of the resulting approximation of the convex envelope. Then, we present an improvement (both in time and accuracy) to the standard algorithm based on a change in the factorization order for the second discrete LFT. This modification is particularly beneficial for bivariate functions. Moreover, we introduce a method for handling functions that are unbounded outside sets of general shape. We also present some situations in which the selection of the dual grid is crucial, and show that it is possible to choose a dual grid of arbitrary size without increasing the memory requirements of the algorithm. Finally, we apply our algorithm to the study of phase separation in non-ideal ionic solutions.

Keywords. Convex envelope; convex hull; Legendre–Fenchel Transform; phase separation.

1 Introduction

In many applications, the equilibrium properties of a thermodynamic system can be studied through the minimization of a certain potential under some macroscopic constraints. In ideal situations, the potential is convex, and solving the constrained minimization problem is straightforward. However, complex modelling taking into account non-ideal effects often leads to non-convex potentials. In this situation, studying the system at equilibrium requires to know the non-convexity region of the potential, i.e., the region where the potential and its convex envelope differ. Depending on the enforced macroscopic constraints, this can lead to phase separation. This behavior was discovered by Maxwell in the study of the Van der Waals equation [15]; he was able to

*Department of Mathematics and Computer Science, University of Udine, via delle Scienze 206, I-33100 Udine, Italy; Graduate School of Advanced Mathematical Sciences, Meiji University 4-21-1 Nakano, Nakano-ku, Tokyo, Japan 164-8525

†Université Paris-Est, CERMICS (ENPC), 77455 Marne la ValléCedex 2, France

‡Department of Mathematics and Computer Science, University of Udine, via delle Scienze 206, I-33100 Udine, Italy

build the correct potential by applying the so-called Maxwell's equal area rule to the derivative of the potential, which, for a univariate potential, is equivalent to finding its convex envelope.

For more complex potentials, e.g., bivariate or multivariate functions, the convex envelope cannot be computed analytically. Since the convex envelope results from a double Legendre–Fenchel Transform (LFT), it can be approximated by a double discrete LFT. This seminal idea can be found in Brenier's work [3]. The main ingredient of the double discrete LFT is obviously the discrete LFT, which can be computed using Lucet's algorithm [13]. This algorithm consists in reducing the transform by means of dimensional factorization to one-dimensional transforms, and the latter can be computed in linear time, that is, in complexity $O(n)$ where n is the number of sample points. A recent example of application of the double discrete LFT to deriving pressure laws in binary mixtures can be found in [10], while a recent review of the state of the art in computational convex analysis and its applications can be found in [14]. We also mention the more sophisticated approach to one-dimensional convex envelope and LFT computation in [8] where piecewise quadratic interpolants are used; unfortunately, the resulting transform is not factorizable, unlike the fast discrete LFT based on piecewise linear interpolation. Nonetheless, the extension to piecewise quadratic functions is very promising, especially in light of recent works on parametric quadratic programming such as [18].

It is also possible to compute the convex envelope without using LFTs but as a solution to a partial differential equation. For example, it is shown in [16] that the convex envelope of a real-valued function is the viscosity solution of a nonlinear obstacle problem; in [17] a numerical method to solve this PDE on a grid is proposed. The basic scheme is obtained by using centered finite differences along directions defined by an associated stencil to approximate the Hessian matrix at the grid point. The resulting discrete solution can be mildly non-convex for small stencils, and the choice of the directions is crucial. Convexity can be achieved by introducing a correction term (which forces the curvature of the solution to be non-zero everywhere) and the convergence of both schemes is proved. A hybrid scheme is also proposed and tested numerically.

Finally, we remark that exact but less simple algorithms for computing the convex hull of a set of points exist; these algorithms can be adapted to the problem of the convex envelope of a function by considering its epigraph. For example, [4] presents an optimal output-sensitive convex hull algorithm with complexity $O(n \log h)$, where h is the output size, while a popular library for computing convex hulls is Qhull [1] which implements an algorithm with worst-case complexity $O(n \log n)$.

The first part of this paper is centered on the convergence properties of the double discrete LFT in the multivariate case. In the literature, most of the papers focus on the computation of the discrete LFT, which has by itself many applications; on the contrary, there are few theoretical results on the double discrete LFT, mainly dealing with convex functions (see for example [6]). Our convergence results on the double discrete LFT use [6] as a starting point. However, in order to prove convergence to the convex envelope and better understand the behavior of the algorithms, it is important to study the *dual grid* on which the second discrete LFT is applied. The choice of the dual grid is one of the main issues regarding an accurate approximation of the convex envelope, an issue which, to our knowledge, has not yet received the attention it deserves in the literature. Incidentally, the choice of the directions stencil in [16] bears similarities to that of the dual grid for the second discrete LFT in our context.

In the second part, we restrict the scope to bivariate functions and present an improvement (both in time and accuracy) of the standard algorithm based on the double

application of Lucet's algorithm. The main idea is to change the factorization order when computing the second discrete LFT. Another improvement consists in merging (through a maximum operation) the results obtained after the two possible changes in the factorization order. We present numerical examples illustrating the benefits of the proposed approach. Moreover, the issue of choosing the dual grid is further investigated, and an efficient handling of dual grids of arbitrary length is devised. We also briefly discuss the treatment of non-rectangular domains.

In the last part, we present a physical application of our algorithm to the study of phase separation in ionic solutions consisting of cations and anions dissolved in a solvent (typically water). Such ionic solutions are encountered in numerous natural and technological contexts (colloids, DNA, electrodes, etc.), largely due to the ability of water to solvate ions and to screen their electrostatic interactions by its high dielectric permittivity. We consider the model presented in [12] where non-ideal effects (due to long-range electrostatic and short-range steric correlations between ions) give rise, under certain conditions, to a non-convex free energy. The free energy has unbounded derivatives for vanishing ionic densities, and, owing to steric correlations, becomes unbounded when the total ionic density reaches a certain finite threshold. We present the shapes of the non-convexity regions, as computed by the present algorithm, for various cases concerning the valences of the ions. Knowledge of the convex envelope of the free energy can be useful to determine the spatial distributions of ions and of the resulting electrostatic potential in confined situations, see [12].

2 Preliminaries

2.1 Legendre–Fenchel Transform

In what follows, we consider functions defined on $\mathbb{R}^n$ and taking values in $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$. Given a function f of this form, we denote by $\underline{f}$ (respectively, $\overline{f}$) the *lower (upper) semi-continuous envelope* of f, i.e., the greatest (smallest) lower (upper) semi-continuous function upper-bounded (lower-bounded) by f, and by $\operatorname{conv} f$ the *convex envelope* of f, i.e., the greatest convex function upper-bounded by f.

Let g be a convex function. We denote by $\operatorname{dom} g$ its *(effective) domain*, i.e., the subset of $\mathbb{R}^n$ where $g < +\infty$; we say that g is *proper* if $\operatorname{dom} g$ is non-empty and g never takes the value $-\infty$. Moreover, we denote by $\operatorname{cl} g$ the *closure* of g, which is $\underline{g}$ if g is proper and the constant function $-\infty$ otherwise; g is said to be *closed* if $g = \operatorname{cl} g$. Given a point $x \in \mathbb{R}^n$ and a vector $\xi \in \mathbb{R}^n$, we say that ξ is a *subgradient* of g at x if the affine function $z \mapsto g(x) + \langle \xi, z - x \rangle$, where $\langle \cdot, \cdot \rangle$ is the scalar product of $\mathbb{R}^n$, is upper-bounded by g; the collection of all subgradients of g at x is called the *subdifferential* of g at x and is denoted by $\partial g(x)$.

Definition 1. *Given $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ not necessarily convex, the* Legendre–Fenchel Transform (LFT) *of f is the function f^* from $\mathbb{R}^n$ to $\overline{\mathbb{R}}$ defined as*

$$f^*(\xi) = \sup_{x \in \mathbb{R}^n} [\langle x, \xi \rangle - f(x)].$$

For an exhaustive treatment of the properties of the LFT, we refer to any introductory book in convex analysis, such as [19]; we recall only the properties which are useful in the development of the algorithm. Namely, f^* is a closed convex function; $f \leq g$ implies $f^* \geq g^*$; $f^{**} = \operatorname{cl}(\operatorname{conv} f)$; for $n \geq 2$, the LFT can be factorized along

each dimension as $f^* = (-(\cdots(-f^{*1})^{*2}\cdots)^{*(n-1)})^{*n}$, where f^{*i} is the LFT along the ith-dimension, i.e.,

$$f^{*i} : (x_{\sim i}, \xi_i) \mapsto \sup_{x_i \in \mathbb{R}} \left[x_i \xi_i - f(x_{\sim i}, x_i) \right], \tag{1}$$

with the notation $\sim i = \{1, \ldots, n\} \setminus \{i\}$.

2.2 Discrete Legendre–Fenchel Transform

In order to define the discrete version of the LFT, we introduce the following notation: given a set $\emptyset \neq \Omega \subseteq \mathbb{R}^n$ and $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, we define f_Ω as

$$f_\Omega(x) = \begin{cases} f(x) & \text{if } x \in \Omega, \\ +\infty & \text{elsewhere.} \end{cases}$$

It is clear that

$$f^*_\Omega(\xi) = \sup_{x \in \Omega} \left[\langle x, \xi \rangle - f(x) \right],$$

and that $f^*_{\Omega'} \geq f^*_\Omega$ when $\Omega \subseteq \Omega'$; moreover, $f^*_{\Omega \cup \Omega'} = \max\left\{ f^*_\Omega, f^*_{\Omega'} \right\}$ for any sets $\Omega, \Omega' \subseteq \mathbb{R}^n$.

Since we are interested in computing the LFT numerically, the case in which Ω is finite is particularly important. In what follows, the subscript N is used to stress the finiteness of the set Ω_N; the notation does not mean that $|\Omega_N| = N$. The set Ω_N is always assumed to be non-empty.

Definition 2. *Given $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ not necessarily convex and a finite set $\Omega_N \subset \mathbb{R}^n$, the* discrete Legendre–Fenchel Transform *of f computed on the set Ω_N is the function $f^*_{\Omega_N}$.*

In what follows, we implicitly assume that

1. $f(x) < +\infty$ for at least one $x \in \Omega_N$, otherwise $f^*_{\Omega_N}$ would be $-\infty$ everywhere;
2. $f(x) > -\infty$ for all $x \in \Omega_N$, otherwise $f^*_{\Omega_N}$ would be $+\infty$ everywhere.

In the one-dimensional case, there is an explicit formula for the discrete LFT; this formula is the foundation for the linear-time one-dimensional algorithm proposed by Lucet in [13]. Moreover, owing to the factorization properties of the LFT, this algorithm is the basic building block of discrete LFT algorithms in higher dimensions.

2.3 Convergence to the Legendre–Fenchel Transform

Our convergence analysis of the double discrete LFT is based on a convergence result for the discrete LFT by Corrias [6], which considers $\Omega = [0,1]^n$ and proves that the discrete LFT converges to the LFT as the set $\Omega_N \subset \Omega$ is refined; the following statement has been adapted to our notation and extended explicitly to more general sets Ω.

Definition 3. *Let $(\Omega_N)_N$ be an increasing sequence of finite subsets of $\Omega \subseteq \mathbb{R}^n$. We say that the sequence converges to Ω if $\min_{x' \in \Omega_N} \|x' - x\| \to 0$ as $N \to \infty$ for all $x \in \Omega$; we denote this as $\Omega_N \to \Omega$. If, additionally, $\sup_{x \in \Omega} \min_{x' \in X_N} \|x' - x\| \to 0$ as $N \to \infty$, we say that the convergence is uniform.*

Theorem 4. *Let $\emptyset \neq \Omega \subseteq \mathbb{R}^n$ and let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be such that $\overline{(f_\Omega)} = \underline{f_\Omega}$. Let $(\Omega_N)_N$ be a sequence of finite subsets of $\mathbb{R}^n$ such that $\Omega_N \to \Omega$. Then, $\overline{f^*_{\Omega_N}}$ converges pointwise to f^*_Ω. Moreover, if $f|_\Omega$ is uniformly continuous and $\Omega_N \to \Omega$ uniformly, then the convergence is uniform on every bounded subset S of* $\operatorname{dom} f^*_\Omega$.

Proof. The proof is essentially the same as that of [6, Theorem 2.1] and can be found in [5, Theorem 2.27]. □ □

The hypothesis $\overline{(f_\Omega)} = \underline{f_\Omega}$ is necessary in order to make Theorem 4 true for every sequence $\Omega_N \to \Omega$, as shown in [6]. A simple setting in which this hypothesis holds is when $\Omega \subseteq \mathrm{cl}\,\mathrm{int}\,\Omega$ and $f|_\Omega$ is upper semicontinuous; the proof can be found in [5, Theorem 2.32]).

3 The double discrete Legendre–Fenchel Transform

3.1 Optimal dual set and grid

Given $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and $\emptyset \neq \Omega \subseteq \mathbb{R}^n$, we consider the problem of computing $\mathrm{conv}\, f_\Omega$. Firstly, we discretize this problem by considering a finite set $\Omega_N \subset \Omega$ and by choosing to approximate $\mathrm{conv}\, f_\Omega$ with the function $\mathrm{conv}\, f_{\Omega_N}$. Secondly, since the application of two successive LFTs to f_{Ω_N} yields its convex envelope $\mathrm{conv}\, f_{\Omega_N}$, we build an approximation of the convex envelope by using two successive discrete LFTs, i.e. taking $\left(f^*_{\Omega_N}\right)^*_{S_M}$, where $S_M \subset \mathbb{R}^n$ is a finite set; we call this operation a *double discrete LFT*, while the set S_M is called the *dual grid*.

A correct choice of the dual grid S_M is essential to obtain an accurate approximation of the convex envelope. For any dual grid S_M, $\left(f^*_{\Omega_N}\right)^*_{S_M} \neq \left(f^*_{\Omega_N}\right)^* = \mathrm{conv}\, f_{\Omega_N}$ as functions defined on $\mathbb{R}^n$; this is true because, by finiteness of Ω_N, the domain of $f^*_{\Omega_N}$ is $\mathbb{R}^n$, and thus, by finiteness of S_M, the convex functions $f^*_{\Omega_N}$ and $\mathrm{conv}\left[\left(f^*_{\Omega_N}\right)_{S_M}\right]$ (and thus their LFTs) differ. Nonetheless, we now show that there exists at least a set S_M which is optimal. In the following statement, the set Ω need not be finite.

Definition 5. *We say that a set $S \subseteq \mathbb{R}^n$ is an* optimal dual set *with respect to Ω if*

$$(f^*_\Omega)^*_S(x) = \mathrm{cl}\,\mathrm{conv}\, f_\Omega(x) \tag{2}$$

for every $x \in \Omega \cap \mathrm{dom}\,\mathrm{cl}\,\mathrm{conv}\, f_\Omega$. Moreover we say that S_M is an optimal dual grid *if it is a finite optimal dual set.*

Remark 6. When the dual set $S = S_M$ is finite, the equality (2) holds nowhere on $\Omega \setminus \mathrm{dom}\,\mathrm{cl}\,\mathrm{conv}\, f_\Omega$, since, by finiteness of S_M, the double discrete LFT is finite everywhere.

Remark 7. If $\Omega = \Omega_N$ is finite, there holds $\mathrm{cl}\,\mathrm{conv}\, f_{\Omega_N} = \mathrm{conv}\, f_{\Omega_N}$, i.e., $\mathrm{conv}\, f_{\Omega_N}$ is closed; see [19, Corollary 19.1.2].

We close this section with two important results concerning dual grids: a sufficient condition for optimality (Theorem 11) and an existence result of finite optimal dual grids when the set Ω_N is itself finite (Theorem 12). The following preliminary Lemma has been inspired by [11, Proposition 2.3], of which it is a direct consequence when Ω is a ball centered at the origin; here, we provide a direct proof for more general sets Ω.

Lemma 8. *Let $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper closed convex function and let $\emptyset \neq \Omega \subseteq \mathbb{R}^n$. Let $\xi_0 \in \mathbb{R}^n$ be such that $\partial g^*(\xi_0) \cap \Omega \neq \emptyset$. Then, $g^*_\Omega(\xi_0) = g^*(\xi_0)$.*

Proof. Let $x_0 \in \partial g^*(\xi_0) \cap \Omega$. Since $x_0 \in \partial g^*(\xi_0)$, by [19, Theorem 23.5 (b,a*)], there holds $g^*(\xi_0) = \langle x_0, \xi_0 \rangle - g(x_0)$. Since $x_0 \in \Omega$, we infer that

$$g^*(\xi_0) \leq \sup_{x \in \Omega} [\langle x, \xi_0 \rangle - g(x)] = g^*_\Omega(\xi_0).$$

By the properties of the LFT, we also obtain $g^*_\Omega(\xi_0) \leq g^*(\xi_0)$, thereby proving the assertion. □ □

Remark 9. We observe that Lemma 8 does not require the finiteness of Ω; if Ω is finite, the converse statement can also be proved [5, Theorem 3.12].

Remark 10. The case in which Ω is a ball centered at the origin has been considered also in [13, Proposition 2] as a corollary of [11] for a general function f. In our opinion, the convexity of the function is an essential assumption to state the result. Moreover, the statement $f^* = f^*_B$ when the ball B is large enough is not true when, for example, the derivative of f^* becomes unbounded.

Theorem 11. *Consider $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and let $\Omega, S \subseteq \mathbb{R}^n$. Let $x \in \mathbb{R}^n$ such that*

$$\partial \operatorname{cl}\operatorname{conv} f_\Omega(x) \cap S \neq \emptyset. \tag{3}$$

Then, $(f^*_\Omega)^*_S(x) = \operatorname{cl}\operatorname{conv} f_\Omega(x)$. As a consequence, if condition (3) holds for every $x \in \Omega \cap \operatorname{dom}\operatorname{cl}\operatorname{conv} f_\Omega$, then S is an optimal dual set with respect to Ω.

Proof. After observing that $(f^*_\Omega)^* = \operatorname{cl}\operatorname{conv} f_\Omega$ by the properties of the LFT, it is sufficient to apply Lemma 8 to the function $g := f^*_\Omega$ and the set S. □ □

Theorem 12. *Consider $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and let $\Omega_N \subset \mathbb{R}^n$ be finite. Then, there exists a finite set $S_M \subset \mathbb{R}^n$ which is an optimal dual grid with respect to Ω_N.*

Proof. Since Ω_N is finite, by [19, Theorem 23.10] we infer $\partial \operatorname{conv} f_{\Omega_N}(x) \neq \emptyset$ for every $x \in \Omega_N \cap \operatorname{dom}\operatorname{conv} f_{\Omega_N}$. Thus, S_M can be built by satisfying the sufficient condition given in Theorem 11; in particular, this yields $|S_M| = \left|\Omega_N \cap \operatorname{dom}\operatorname{conv} f_{\Omega_N}\right|$ which is finite. □ □

3.2 Convergence to the convex envelope

We now return to the double discrete LFT $\left(f^*_{\Omega_N}\right)^*_{S_M}$. We assume that Ω_N has been chosen, while the dual grid S_M has yet to be determined; as seen in the previous section, the best choice for S_M is given by an optimal dual grid. Unfortunately, finding it is a difficult task since we need to know something about $\operatorname{conv} f_{\Omega_N}$. Nonetheless, finding a non-finite optimal dual set S is generally easier (see for example Section 4.1 where a possible optimal dual set S is given in the case $\Omega_N = X_N \times Y_N \subset \mathbb{R}^2$); moreover, since by Theorem 12 we know that a finite dual set exists, we can hopefully find a bounded set S. The following Theorem shows that by considering a sequence of non-optimal dual grids $S_M \to S$, we can achieve convergence of the double discrete LFT to the convex envelope.

Theorem 13. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, let Ω_N be a finite subset of $\mathbb{R}^n$ and let $S \subseteq \mathbb{R}^n$ be an optimal dual set such that $S \subseteq \operatorname{cl}\operatorname{int} S$. Let $(S_M)_M$ be a sequence of finite subsets of $\mathbb{R}^n$ such that $S_M \to S$. Then, for fixed Ω_N, $\left(f^*_{\Omega_N}\right)^*_{S_M}(x) \to \operatorname{conv} f_{\Omega_N}(x)$ for every $x \in \Omega_N \cap \operatorname{dom}\operatorname{conv} f_{\Omega_N}$.*

Proof. The function $f^*_{\Omega_N}$ and the set S satisfy the hypotheses of Theorem 4 since $f^*_{\Omega_N}$ is continuous on $\mathbb{R}^n$ by finiteness of Ω_N. Hence, $\left(f^*_{\Omega_N}\right)^*_{S_M} \to \left(f^*_{\Omega_N}\right)^*_S$. The conclusion follows since, owing to the optimality of S, we infer $\left(f^*_{\Omega_N}\right)^*_S = \operatorname{conv} f_{\Omega_N}$ on $\Omega_N \cap \operatorname{dom}\operatorname{conv} f_{\Omega_N}$. □ □

By repeating the final steps of the proof of Theorem 4, we obtain the following error estimate.

Corollary 14. *Under the hypotheses of Theorem 13, let $x_0 \in \Omega_N \cap \operatorname{dom}\operatorname{conv} f_{\Omega_N}$ and let $\xi_0 \in \partial \operatorname{conv} f_{\Omega_N}(x_0) \cap S$. Then,*

$$|\operatorname{conv} f_{\Omega_N}(x_0) - \left(f^*_{\Omega_N}\right)^*_{S_M}(x_0)| \leq \|\xi_0 - \hat{\xi}_M\| \, \|x_0\| + |f^*_{\Omega_N}(\hat{\xi}_M) - f^*_{\Omega_N}(\xi_0)|,$$

where $\hat{\xi}_M = \arg\min_{\xi \in S_M} \|\xi_0 - \xi\|$.

Proof. Since $\xi_0 \in \partial \operatorname{conv} f_{\Omega_N}(x_0) = \partial \left(f^*_{\Omega_N}\right)^*(x_0)$, by [19, Theorem 23.5 (b,a*)], we infer $\left(f^*_{\Omega_N}\right)^*(x_0) = \langle x_0, \xi_0 \rangle - f^*_{\Omega_N}(\xi_0)$. Moreover, by optimality of S, we also obtain $\left(f^*_{\Omega_N}\right)^*_S(x_0) = \left(f^*_{\Omega_N}\right)^*(x_0)$. We can then write

$$\begin{aligned}
0 &\leq \left(f^*_{\Omega_N}\right)^*_S(x_0) - \left(f^*_{\Omega_N}\right)^*_{S_M}(x_0) \\
&= \langle x_0, \xi_0 \rangle - f^*_{\Omega_N}(\xi_0) - \max_{\xi \in S_M} \left[\langle x_0, \xi \rangle - f^*_{\Omega_N}(\xi)\right] \\
&\leq \langle x_0, \xi_0 \rangle - f^*_{\Omega_N}(\xi_0) - \langle x_0, \hat{\xi}_M \rangle + f^*_{\Omega_N}(\hat{\xi}_M) \\
&\leq \langle \xi_0 - \hat{\xi}_M, x_0 \rangle + f^*_{\Omega_N}(\hat{\xi}_M) - f^*_{\Omega_N}(\xi_0) \\
&\leq \|\xi_0 - \hat{\xi}_M\| \, \|x_0\| + |f^*_{\Omega_N}(\hat{\xi}_M) - f^*_{\Omega_N}(\xi_0)|,
\end{aligned}$$

and the corollary is proved. □ □

Remark 15. Since the function $f^*_{\Omega_N}$ is continuous, Corollary 14 says that the error in the approximation of $\operatorname{conv} f_{\Omega_N}$ at the point x_0 is determined only by the density of the grid S_M near a point of $\partial \operatorname{conv} f_{\Omega_N}(x_0)$; should we be interested in the value of $\operatorname{conv} f_{\Omega_N}$ only in one point $x_0 \in \Omega_N$, we could consider as the set S a neighborhood of a point of $\partial \operatorname{conv} f_{\Omega_N}(x_0)$.

4 An algorithm for the convex envelope

4.1 Standard factorization

The discrete LFT, being a special case of LFT, can be factorized in a similar way. For simplicity, we focus on the case $n = 2$, i.e., bivariate functions; see Remark 18 for the multivariate case. Consider $f : \mathbb{R}^2 \to \overline{\mathbb{R}}$ and $\Omega = X \times Y \subset \mathbb{R}^2$, not necessarily finite. The dimensional factorization (1) yields

$$f^*_\Omega = \left(-f^{*1}_\Omega\right)^{*2}.$$

Since for every $y \in \mathbb{R}$ we can have $f_\Omega(x,y) < +\infty$ only if $x \in X$, we obtain

$$g(\xi, y) := (f_\Omega)^{*1}(\xi, y) = [f_\Omega(\cdot, y)]^*(\xi) = [f(\cdot, y)]^*_X(\xi).$$

In particular, when $y \notin Y$, there holds $f_\Omega(\cdot, y) \equiv +\infty$ and thus $g(\cdot, y) \equiv -\infty$. Finally, we obtain

$$f^*_\Omega(\xi, \eta) = (-g)^{*2}(\xi, \eta) = [-g(\xi, \cdot)]^*(\eta) = [-g(\xi, \cdot)]^*_Y(\eta),$$

where the last equality follows from the fact that, fixed ξ, the function $-g(\xi,\cdot)$ is $+\infty$ outside Y.

In the case of the discrete LFT, i.e., when $\Omega = \Omega_N = X_N \times Y_N$ is finite with $|X_N| = n$ and $|Y_N| = m$, we obtain a factorization of the two-dimensional discrete LFT into one-dimensional discrete LFTs. In particular, in order to compute $f^*_{\Omega_N}(\xi,\eta)$ for a given $(\xi,\eta) \in \mathbb{R}^2$, we need to compute $g(\xi,y_j)$ for every $j = 1,\dots,m$ using m one-dimensional discrete LFTs along the x-direction (each on data whose length is n) and then applying one one-dimensional discrete LFT along the y-direction (on data whose length is m); if we want to compute the transform for $(\xi,\eta) \in S_M$, where S_M is a grid with the same size as Ω_N, we then have in total m discrete LFTs on data of size n and n discrete LFTs on data of size m. Having the one-dimensional algorithm linear complexity, the complexity of a two-dimensional algorithm based on this factorization is $O(nm)$, again linear; this is the algorithm presented in [13]. Finally, by choosing a dual grid of the tensor-product form $S_M = C_M \times D_M \subset \mathbb{R}^2$, we can compute the double discrete LFT $\left(f^*_{\Omega_N}\right)^*_{S_M}$ by applying twice the algorithm just presented; this is the algorithm used in [10], to which we refer henceforth as the "standard factorization" algorithm.

The choice of the dual grid is a crucial issue common to all convex envelope algorithms based on the double discrete LFT and, in our opinion, has not been stressed enough in the literature. Owing to Theorem 13, it is reasonable to choose S_M as a discretization of an optimal dual set S, possibly as small as possible; but also the type of discretization is relevant and we treat it in Section 4.5. In the case $\Omega_N = X_N \times Y_N$ and $f < +\infty$ on Ω_N, a possible choice of S is the set $[\xi^-,\xi^+] \times [\eta^-,\eta^+]$ built as follows. Given $y \in \mathbb{R}$, let $g_y := f_{\Omega_N}(\cdot,y)$, $\xi_y^- := \max \partial \operatorname{conv} g_y(\min X_N)$ and $\xi_y^+ := \min \partial \operatorname{conv} g_y(\max X_N)$; this means that ξ_y^- and ξ_y^+ are, respectively, the right and left derivative of $\operatorname{conv} g_y$ at $\min X_N$ and $\max X_N$. We then define ξ^- as $\min_{y\in Y_N} \xi_y^-$ and ξ^+ as $\max_{y\in Y_N} \xi_y^+$; η^- and η^+ are defined in a similar manner. A proof of this result can be found in [5, Theorem 4.15].

4.2 Alternating factorization

Since the order of factorization is arbitrary, we do not need to factorize both discrete LFTs in the same way. We can decompose the second discrete LFT as

$$\left(f^*_{\Omega_N}\right)^*_{S_M} = \left[-\left(f^*_{\Omega_N}\right)^{*2}_{S_M}\right]^{*1},$$

obtaining that

$$\left(f^*_{\Omega_N}\right)^*_{S_M}(x,y) = \left[-h(\cdot,y)\right]^*_{C_M}(x),$$

where

$$h(\xi,y) = \left[f^*_{\Omega_N}(\xi,\cdot)\right]^*_{D_M}(y) = \left[\left(-g(\xi,\cdot)\right)^*_{Y_N}\right]^*_{D_M}(y).$$

This means that $h(\xi,\cdot)$ is an approximation of the convex envelope of $\left(-g(\xi,\cdot)\right)_{Y_N}$. By substituting h with the true convex envelope (which is easily calculable in one dimension) and by observing that

$$\operatorname{conv}\left[\left(-g(\xi,\cdot)\right)_{Y_N}\right] = \left[\left(-g(\xi,\cdot)\right)^*_{Y_N}\right]^*,$$

we obtain a way to compute the function $\left(f^*_{\Omega_N}\right)^*_{C_M\times\mathbb{R}}$ in linear time, even if the dual set $C_M\times\mathbb{R}$ is not finite. This function is a better approximation of conv f_{Ω_N} than $\left(f^*_{\Omega_N}\right)^*_{S_M}$ since, owing to the properties of the FLT,

$$\left(f^*_{\Omega_N}\right)^*_S \leq \left(f^*_{\Omega_N}\right)^*_{S'} \leq \left(f^*_{\Omega_N}\right)^* = \operatorname{conv} f_{\Omega_N}$$

for every sets $S, S' \subseteq \mathbb{R}^2$ such that $S \subseteq S'$.

Definition 16. *We refer to the algorithm computing* $\left(f^*_{\Omega_N}\right)^*_{C_M\times\mathbb{R}}$ *by the factorization presented above as the* alternating factorization *algorithm.*

A further variant is to consider the two possible changes in the factorization order and merge the results through a maximum operation, that is, we compute $\left(f^*_{\Omega_N}\right)^*_{C_M\times\mathbb{R}}$ and $\left(f^*_{\Omega_N}\right)^*_{\mathbb{R}\times D_M}$ and set

$$\left(f^*_{\Omega_N}\right)^*_{(C_M\times\mathbb{R})\cup(\mathbb{R}\times D_M)} = \max\left\{\left(f^*_{\Omega_N}\right)^*_{C_M\times\mathbb{R}}, \left(f^*_{\Omega_N}\right)^*_{\mathbb{R}\times D_M}\right\}. \tag{4}$$

Definition 17. *We refer to the algorithm computing* $\left(f^*_{\Omega_N}\right)^*_{(C_M\times\mathbb{R})\cup(\mathbb{R}\times D_M)}$ *from* (4) *as the* max-alternating factorization *algorithm.*

The max-alternating factorization algorithm improves the alternating factorization one in terms of accuracy since the dual set has been enlarged again. Our numerical results in Section 4.4 show that the max-alternating factorization algorithm is also more efficient in terms of computational time to achieve a certain accuracy threshold (except for very simple functions). Finally, when applied to functions which are invariant under swap of coordinates, the max-alternating factorization algorithm preserves this property as the exact convex hull does.

The dual sets used for the double discrete LFTs presented in this section are no longer finite. They can be built following the construction at the end of Section 4.1 for an optimal dual set of the form $S = [\xi^-, \xi^+] \times [\eta^-, \eta^+]$. Specifically, we choose C_M as a discretization of $[\xi^-, \xi^+]$ and D_M as a discretization of $[\eta^-, \eta^+]$; in what follows, when speaking of the dual grid in relation to the alternating factorization algorithms, we refer to the finite sets C_M and D_M, and not to the actual dual sets $C_M \times \mathbb{R}$ and $\mathbb{R} \times D_M$ used in the transforms.

Remark 18. The alternating factorization algorithm can be also applied in dimension $n > 2$, but we expect less significant improvements in speed and precision the higher the dimension is, since the number of loops is always reduced by one independently of n.

4.3 Implementation

Let us first briefly recall the standard algorithm for the double discrete LFT. Consider a function $f : \mathbb{R}^2 \to \overline{\mathbb{R}}$, a primal grid $\Omega_N = X_N \times Y_N$, and a dual grid $S_M = C_M \times D_M$. The algorithm consists in evaluating $f^*_{\Omega_N}$ on the grid S_M and then $(f^*_{\Omega_N})^*_{S_M}$ on the grid Ω_N. A simple choice for the dual grid S_M is a uniform discretization of the optimal dual set $S = [\xi^-, \xi^+] \times [\eta^-, \eta^+]$. We notice that $\xi^\pm$ and $\eta^\pm$ can be easily found by computing the interval containing all the natural grids of the one-dimensional discrete

LFTs, respectively, along the rows and along the columns, since in one dimension those grids correspond to the slopes of the convex envelope. More elaborate choices for the dual grid, which are instrumental when approximating the convex envelope of complex functions, are discussed in Sections 4.5 and Appendix A.

In order to implement the alternating factorization algorithm, we can adapt the standard factorization algorithm just presented by condensing the two consecutive one-dimensional discrete LFTs in the same direction into a single application of the convex envelope operation; thus the computation of $\left(f^*_{\Omega_N}\right)^*_{C_M\times\mathbb{R}}$ is not only slightly better, but also slightly faster (see, for instance, the results in Table 3 below). Algorithm 1 presents a pseudo-code description of the alternating factorization algorithm. We denote by `convexenvelope_1d(grid, values)` the subroutine which computes the convex envelope of the piecewise linear interpolant on the nodes `grid` and the values `values` (the row index in `values` corresponds to the first dimension, while the column index to the second one). Moreover, we denote by `fast_dlft(grid, values)` the subroutine which computes the one-dimensional discrete LFT by Lucet's algorithm. Both subroutines return a piecewise linear function, which is represented by an object `pcw` whose components are `pcw.grid` and `pcw.values`; we denote by `pcw(grid)` the evaluation of the piecewise linear function on the points of `grid`. We use an array notation where indexes start from 1 and where the index -1 is a shortcut for the last index; the function `size(array)` gives the size of an array. Moreover, we denote by `linspace(start,stop,N)` the grid which discretizes the interval $[$`start`$,$`stop`$]$ with `N` evenly spaced points. Finally, the keyword `parallel` denotes the loops whose iterations are independent and thus computed in parallel.

The implementation of the max-alternating factorization algorithm follows along the same lines. A practically relevant observation is that it is not necessary to implement a new function when computing $\left(f^*_{\Omega_N}\right)^*_{\mathbb{R}\times D_M}$. Indeed, it suffices to change the order of the dimensions (i.e., transpose the data matrices) before and after Algorithm 1.

Finally, we observe that, owing to finite precision arithmetic, both the standard and the modified algorithms can contain points where the output values are greater than the input values. In order to satisfy the property $\left(f^*_{\Omega_N}\right)^*_{S_M} \leq f_{\Omega_N}$, it is possible to take as the convex envelope the minimum between the input and output values. This test is used in what follows to detect the region where the function differs from its convex envelope.

The readers interested in the codes can request them to the first author.

4.4 Numerical comparisons

In this section we present numerical tests to compare the three above algorithms to approximate the double discrete LFT, which we abbreviate as s-dDLFT (standard algorithm for the double discrete LFT), a-dDLFT (alternating factorization) and ma-dDLFT (max-alternating factorization). In all cases, the dual grid is built from a uniform discretization of the optimal dual set $S = [\xi^-, \xi^+] \times [\eta^-, \eta^+]$, with the same size as the grid Ω_N on which the function is approximated. We first test the algorithms on functions of the form $f(x,y) = f(r)$, where r is the distance to the origin of the point (x,y), i.e., $r = \sqrt{x^2+y^2}$; a non-radial function is tested in Section 4.5. The first test function we consider is

$$f^1_{\text{test}}(x,y) = (r^2-1)^2,$$

Input: $X_N \to \texttt{xgrid}$, $Y_N \to \texttt{ygrid}$, $f(X_N \times Y_N) \to \texttt{values}$

Output: $\left(f^*_{\Omega_N}\right)^*_{C_M \times \mathbb{R}} \to \texttt{values}$

```
! Step 1: compute the discrete LFT along each column
lwb = +Inf
upb = -Inf
parallel do j = 1,size(ygrid)
  pcws(j) = fast_dlft(xgrid, values(:,j))
  lwb = min(lwb, pcws(j).grid(1))
  upb = max(upb, pcws(j).grid(-1))
end do
! Step 2: build the dual grid C_M
!         and evaluate each of the discrete LFTs on it
cgrid = linspace(lwb, upb, size(xgrid))
parallel do j = 1,size(ygrid)
  values(:,j) = -pcws(j)(cgrid)
end do
! Step 3: compute the convex envelope of each row
parallel do i = 1,size(cgrid)
  pcw = convexenvelope_1d(ygrid, values(i,:))
  values(i,:) = -pcw(ygrid)
end do
! Step 4: compute the discrete LFT along each column
parallel do j = 1,size(ygrid)
  pcw = fast_dlft(cgrid, values(:,j))
  values(:,j) = pcw(xgrid)
end do
```

Algorithm 1: Alternating factorization algorithm for the double discrete LFT.

whose convex envelope is

$$\operatorname{conv} f^1_{\text{test}}(x,y) = \begin{cases} (r^2-1)^2 & \text{if } r > 1, \\ 0 & \text{if } 0 \le r \le 1. \end{cases}$$

The grid Ω_N consists of 1000×1000 points uniformly distributed on the square $[-1.5, 1.5]^2$. As a first assessment of the quality of the numerical results, we compare the regions flagged as non-convex by the different algorithms, i.e., where the computed convex envelope differs from the value of the function; in the present case the exact region of non-convexity is the unit disk. The regions detected by the s-dDLFT, a-dDLFT and ma-dDLFT are shown in white in Figure 1; the loss of invariance under swap of coordinates for the a-dDLFT is visible in the different shape of the region around the two axes. In Figure 2, the error between the computed and exact convex envelopes is shown, whereas in Table 1, quantitative information about this error is reported. It is clear that the change of factorization order employed in the a-dDLFT improves the approximation of the convex envelope, and that the ma-dDLFT further improves the results. These successive improvements are also visible from Figure 3 which displays the

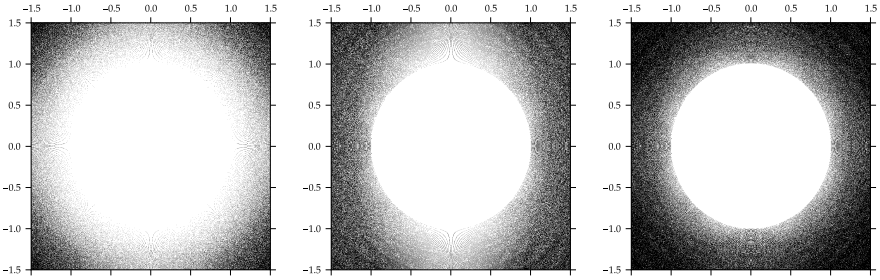

Figure 1: Non-convexity region of the function f^1_{test} on the domain $[-1.5,1.5]^2$, computed from left to right by the s-dDLFT, a-dDLFT and ma-dDLFT; the points where the computed convex envelope differs from the function f^1_{test} are shown in white.

Table 1: Errors between the exact convex envelope of f^1_{test} and its computed value. The values for the a-dDLFT and ma-dDLFT are expressed as the relative error reduction with respect to the s-dDLFT.

inside $[-1.5,1.5]^2$	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0297	-29.30%	-29.30%
Mean value	0.0041	-23.73%	-37.03%
Standard deviation	0.0068	-21.64%	-32.55%
inside the unit circle	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0297	-29.30%	-29.30%
Mean value	0.0117	-23.55%	-36.76%
Standard deviation	0.0068	-18.24%	-25.29%
outside the unit circle	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0026	-2.77%	-98.14%
Mean value	$2.883\cdot 10^{-5}$	-64.12%	-96.96%
Standard deviation	$8.565\cdot 10^{-5}$	-37.99%	-95.87%

restriction of the transforms to the axes. Finally, to illustrate Theorem 11, we have also included in Figure 2 the case in which the value 0 is inserted manually into each one-dimensional dual grid used by ma-dDLFT; being $(0,0)$ the value of the gradient in the non-convex region, we obtain an almost exact result inside the region of non-convexity.

The second test function (see Figure 4) is

$$f^2_{\text{test}}(x,y) = \exp r + 25\cdot\sin(2.5-r)\cdot\exp\left[-(2.5-r)^2\right],$$

and, by observing that $\operatorname{conv} f^2_{\text{test}}(x,y) = \operatorname{conv} f^2_{\text{test}}(r)$, we can build a very accurate approximation of the convex envelope of f^2_{test} by a one-dimensional computation (we denote by s the value of the one-dimensional derivative inside the region of non-convexity). We treat this convex envelope as the "exact" one in order to test the two-dimensional algorithms. The domain is the square $[-3.75,3.75]^2$ discretized by a 1000×1000 uniform grid. The exact region of non-convexity is an annulus with radii 0.47 and 2.87, whereas the regions computed numerically are presented in Figure 5. In Figure 6 we plot the errors produced by the various algorithms; for the fourth

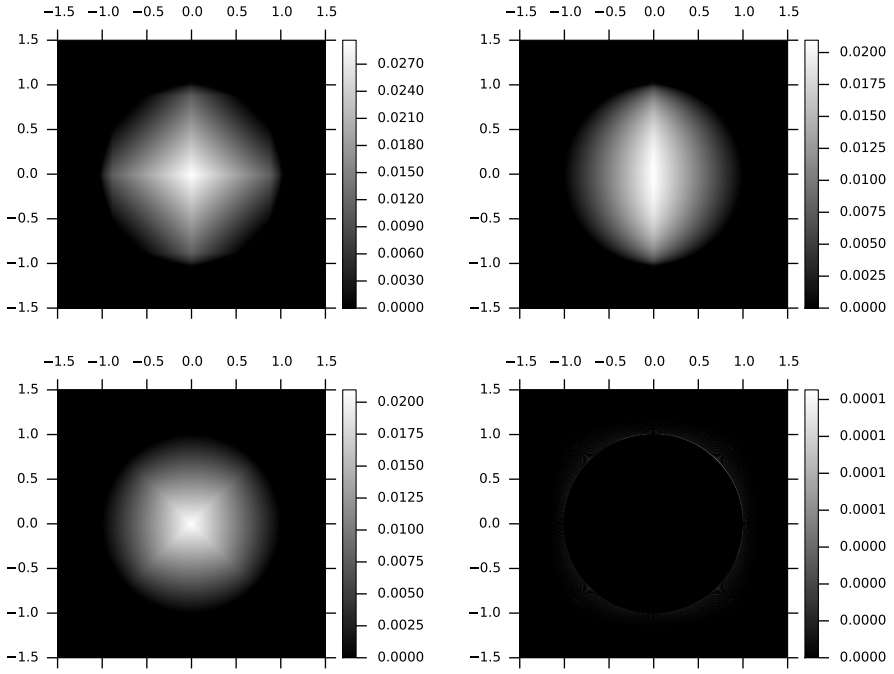

Figure 2: Error between the exact convex envelope of f^1_{test} and its value computed from left to right and from top to bottom by the s-dDLFT, a-dDLFT, ma-dDLFT and ma-dDLFT using a dual grid containing 0.

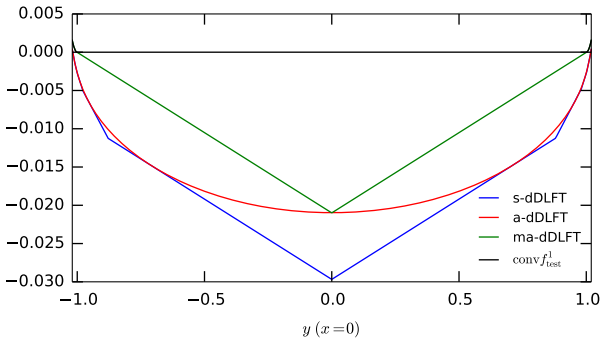

Figure 3: Restriction to the y-axis of the convex envelope of f^1_{test} computed respectively by s-dDLFT (blue line), a-dDLFT (red line) and ma-dDLFT (green line); the exact convex hull is shown in black.

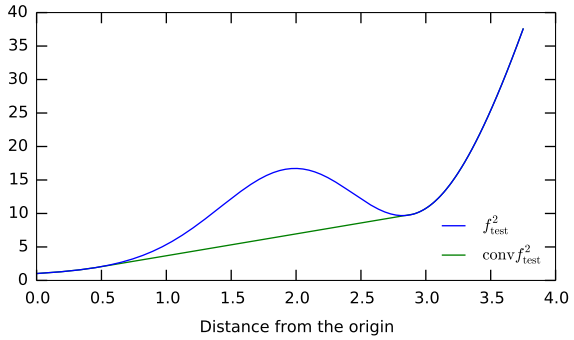

Figure 4: The graph of the test function f^2_{test} (blue line) and of its convex envelope (green line).

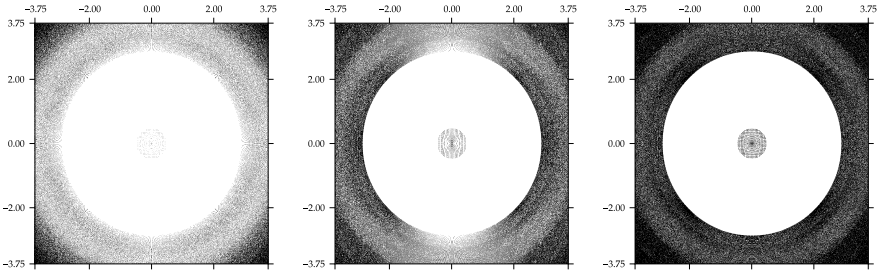

Figure 5: Non-convexity region of the function f^2_{test} on the domain $[-3.75, 3.75]^2$, computed from left to right by the s-dDLFT, a-dDLFT and ma-dDLFT; the points where the computed convex envelope differs from the function f^2_{test} are shown in white.

plot, we inserted manually into the grids the value s (positive), but in this case the region where the resulting transform is exact is much smaller since the gradient in the region of non-convexity is constant only in norm. By studying Table 2, we observe that the improvement achieved by the a-dDLFT over the s-dDLFT is much larger for f^2_{test} than for f^1_{test}. Moreover, there is a large reduction in the maximum error in the non-convexity region when we use the ma-dDLFT; this does not happen for f^1_{test} since the maximum error is reached at the origin which is invariant under swap of coordinates. We have also tested grids with increasing size (both in primal and dual space) in order to check numerically the convergence to the exact convex hull; the maximum and mean error as functions of the grid size for the three variants of the double DLFT algorithm are reported in Figure 7.

Finally, we compare the running time of the algorithms. The a-dDLFT is as expected faster than the s-dDLFT with a 32% reduction in computational time, while computing the ma-dDLFT takes 49% more time than the s-dDLFT. However, in order to properly compare the a-dDLFT and ma-dDLFT algorithms, we have to account not only for the computational time, but also for the quality of the results. A possible way to do so is to compute the ma-dDLFT on a 1000×1000 grid and the a-dDLFT on grids having size $(1000+i) \times (1000+i)$ with $i = 1, 2, \ldots$; then, we choose i as the first value for which the error on the convex envelope measured in a certain metric (maximum,

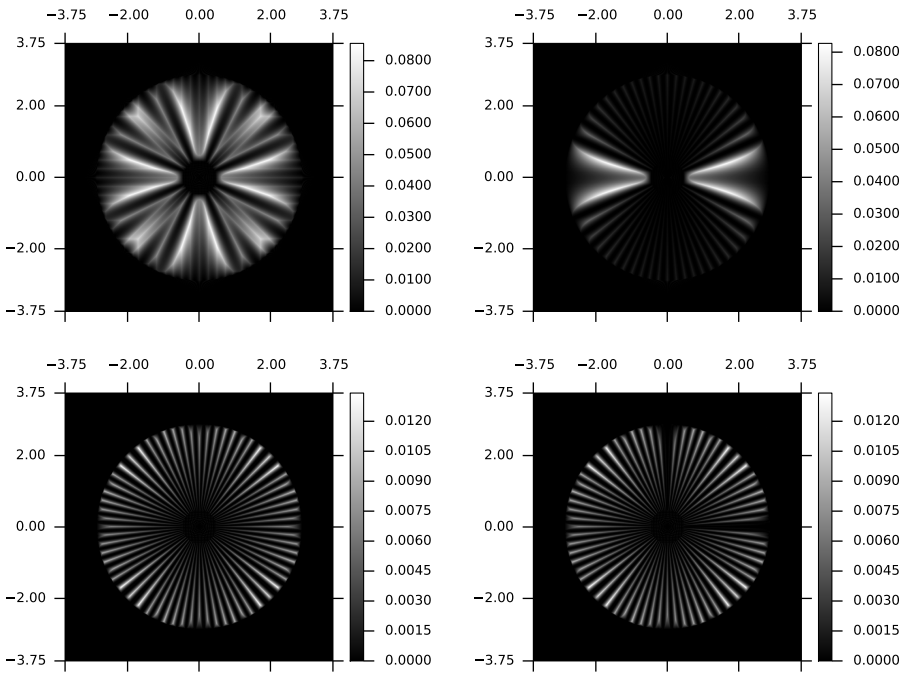

Figure 6: Errors between the exact convex envelope of f_{test}^2 and its value computed respectively from left to right and from top to bottom by the s-dDLFT, a-dDLFT, ma-dDLFT and ma-dDLFT using dual grids containing s.

Table 2: Errors between the exact convex envelope of f_{test}^2 and its computed value. The values for the a-dDLFT and ma-dDLFT are expressed as the relative error reduction with respect to the s-dDLFT.

on $[-3.75, 3.75]^2$	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0855	-3.29%	-84.32%
Mean value	0.0125	-65.31%	-92.34%
Standard deviation	0.0182	-39.18%	-89.75%
inside the annulus	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0855	-3.29%	-84.32%
Mean value	0.0276	-65.35%	-92.31%
Standard deviation	0.0181	-17.08%	-87.24%
outside the annulus	s-dDLFT	a-dDLFT	ma-dDLFT
Maximum value	0.0081	-0.00%	-80.97%
Mean value	$2.219 \cdot 10^{-4}$	-60.58%	-95.45%
Standard deviation	$3.725 \cdot 10^{-4}$	-33.79%	-84.17%

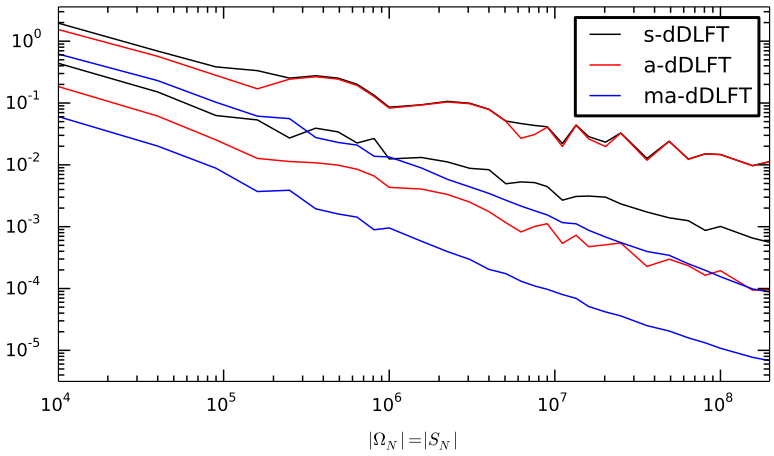

Figure 7: The maximum and mean errors for the convex hull of f^2_{test} as functions of the grid size $|\Omega_N| = |S_N|$

Table 3: Values of i at which the a-dDLFT produces a better result than the ma-dDLFT in each of the three error measures. The time comparisons do not include input generation.

	f^1_{test} (odd i's are excluded)	f^2_{test}
Maximum error	$i = 2$ Time: -50% of ma-dDLFT	$i > 3000$
Mean error	$i = 212$ Time: -27% of ma-dDLFT	$i = 1279$ Time: $+173\%$ of ma-dDLFT
Standard deviation	$i = 162$ Time: -33% of ma-dDLFT	$i = 2245$ Time: $+472\%$ of ma-dDLFT

mean, or standard deviation) becomes smaller than the reference value obtained with the ma-dDLFT. In this way, we can compare the time the two algorithms require to reach a result of comparable quality. Some care is needed in this assessment, since the convergence of the error measures to zero with increasing i is not always monotone; for example, the error for f^1_{test} is always very small when i is odd since the dual grid contains the slope 0. The results are reported in Table 3. We conclude that while the a-dDLFT outperforms the ma-dDLFT for the simple test function f^1_{test}, it is much slower when applied to the slightly more complex f^2_{test}. The time taken by the ma-dDLFT is about 0.78 s for both test functions; the algorithms have been implemented in Fortran (with parallelization provided by OpenMP) and are called from a Python interpreter under Mac OS X 10.8 on a 2.93 GHz Intel Core i7 (4 cores, HT).

4.5 The choice of the dual grid

In the rest of the paper we always consider the more accurate ma-dDLFT, but the problems and possible solutions presented in this section concerning the dual grid are relevant to the other algorithms as well.

The main issue in the accurate computation of the convex envelope is the choice of

the dual grids C_M and D_M; as we have seen in Theorem 13, a discretization $C_M \times D_M$ of the optimal dual set $S = [\xi^-, \xi^+] \times [\eta^-, \eta^+]$ represents a reasonable choice. Until now, we have used a uniform discretization of S; however, when the rate of change of the derivatives varies significantly over the domain, this simple choice does not work well. This is the case of both the test functions f^1_{test} and f^2_{test} when enlarging the domain on which the convex envelope is computed. Figure 8 illustrates this fact for the second test function. It is interesting to observe that the plots are composed of black curves whose spacing is the greater the nearer they are to the origin: each of these curves corresponds to a point in the dual grid. Let us consider for example $\xi \in C_M$; being $C_M \times \mathbb{R}$ the actual set on which the second transform is computed, we infer from Theorem 11 that for every point $(x,y) \in \Omega_N$, such that $(\xi,\eta) \in \partial \operatorname{conv} f_{\Omega_N}(x,y)$ for a certain $\eta \in \mathbb{R}$, there holds $\operatorname{conv} f_{\Omega_N}(x,y) = \left(f^*_{\Omega_N}\right)^*_{C_M \times \mathbb{R}}$. Since in our case the test functions and their convex envelopes are differentiable, the locus of the points where the first component of the gradient is ξ, corresponds to one of the black curves, i.e., is composed of points where the algorithm yields the exact result. By observing that the gradient is always directed radially and that its module depends only on the distance to the origin and increases with it, we can easily explain the shape and the origin of the black curves, which are asymptotically tangent to the y-axis; a similar remark also holds for the curves asymptotically tangent to the x-axis, which are related to the points $\eta \in D_M$. Since in our tests the second derivative increases with the distance to the origin, we can explain the reason they are denser and the result of the algorithm better far away from the origin by the fact that the distance between successive points of the dual grids is always the same, but the distance in primal space needed for the same increase in the gradient becomes smaller as we move away from the origin.

Remark 19. If the rate of change of the derivatives of f is constant, then it is satisfactory to use a large set S uniformly discretized. Consider for example the convex function $f(x,y) = Kr^2$ on the domain $[-1,1]^2$; by increasing the constant K we can enlarge the optimal dual set S to be discretized uniformly. Since the derivatives of f grow linearly with r, the distance between the black curves corresponding to the points of the dual grid is the same everywhere. If we take a dual grid which is at least as large as the primal grid, then there are enough curves to cover the entire space and thus the result is (almost) exact.

A first solution to this problem consists in increasing the size of the dual grid. It is possible to use a dual grid of arbitrary length without increasing memory usage by partitioning the dual grid; this method can be applied to the three algorithms presented and also to their higher-dimensional variants. Given an arbitrarily large dual grid S_L, we partition it as $\bigcup_{j=1,\dots,s} S^{(j)}_M$, where $\left|S^{(j)}_M\right| = |\Omega_N|$ for all $j = 1,\dots,s$; the results obtained using the various dual grids $S^{(j)}_M$ are then merged using $\left(f^*_{\Omega_N}\right)^*_{S_L} = \max_{j=1,\dots,s} \left(f^*_{\Omega_N}\right)^*_{S^{(j)}_M}$. We can build an algorithm which has memory requirements independent of $|S_L|$, as shown in Algorithm 2; moreover, we can perform the first step of discrete LFTs only once and use this result in the calculation of each $\left(f^*_{\Omega_N}\right)^*_{S^{(j)}_M}$, $j = 1,\dots,s$.

We can use Algorithm 2 to compute the double discrete LFT with a uniform dual grid of arbitrary size. However, this approach is not very efficient (with respect to computational time) since it requires dual grids of extremely large size (see for example the bottom right plot in Figure 9). As discussed in Remark 19, the difficulty

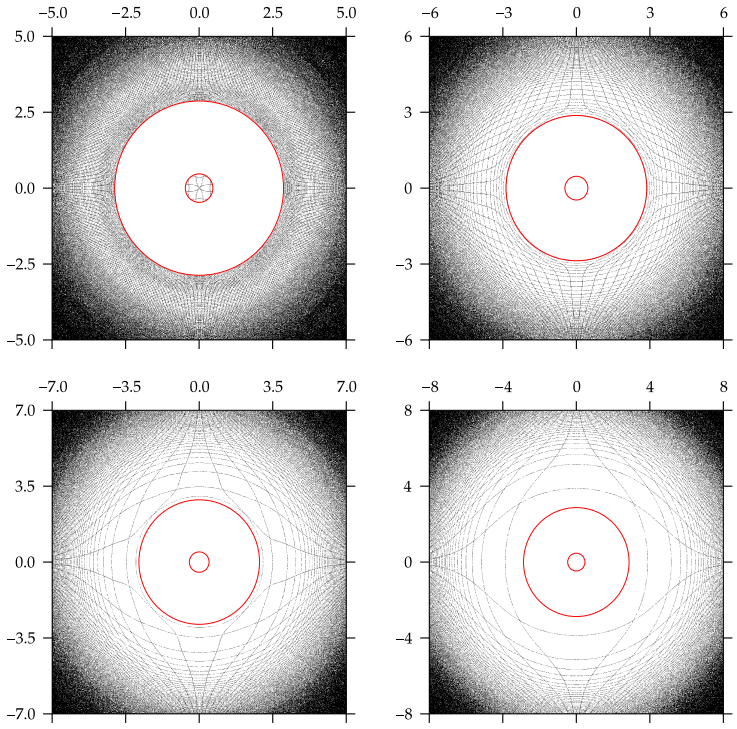

Figure 8: Non-convexity region of the function f^2_{test} respectively from left to right and from top to bottom on the domains $[-5,5]^2$, $[-6,6]^2$, $[-7,7]^2$ and $[-8,8]^2$; the points where the computed convex envelope differs from the function f^2_{test} are shown in white, while the boundary of the exact non-convexity region is shown in red.

does not lie in the density of the dual grid but in its uniformity. A non-uniform distribution of points has to be considered since intervals of equal length in dual space can correspond to intervals of varying length in primal space. A possible heuristic approach uses as the dual grid S_L the union of all the grids `pcws(:).grid` of the first step of one-dimensional discrete LFTs. This large grid in dual space can be partitioned naturally in correspondence to each line on which the one-dimensional LFT has been computed: this means taking `S_M(j)=pcws(j).grid` and `s=size(ygrid)` in Algorithm 2. Incidentally, we observe that the resulting dual grid S_L can be non-optimal, so that the computed envelope is not necessarily well-approximated. In practice, accurate results are achieved by considering only few lines; for example, we can fix $2 \leq \mathtt{s} \leq \mathtt{size(ygrid)}$ and take `S_M(j)=pcws(indexes(j)).grid`, where `indexes=floor(linspace(1,size(ygrid),s))`.

In order to test if the choice $S_M^{(j)}$ from the grids `pcws(:).grid` delivers more accurate results compared to a uniform grid, we consider the test function

$$f^3_{\text{test}}(x,y) = f^1_{\text{test}}(x,y) + \sum_{i=1}^{7} \left[f^1_{\text{test}}(0,10i) \cdot e^{-10(r-10i)^2} \right].$$

In Figure 9 the computed non-convexity region of f^3_{test} is plotted for a heuristic non-uniform dual grid with $s = 2,3,10,20$. The primal grid has size 1000×1000, while the

Input: $X_N \to$ `xgrid`, $Y_N \to$ `ygrid`,

$f(X_N \times Y_N) \to$ `values`, $S_M^{(j)}, j = 1, \ldots, s \to$ `S_M(s)`

Output: $\left(f^*_{\Omega_N}\right)^*_{S_L} \to$ `cur_values`

```
! Step 1 (common for all algorithm variants)
parallel do j = 1,size(ygrid)
  pcws(j) = fast_dlft(xgrid, values(:,j))
end do
! iterate over the dual grids S_M
cur_values = -Inf
do j = 1,s
  ! complete the computation of the double discrete LFT
  ! using the dual grid S_M(j)
  temp_values = ... (depending on the factorization used)
  ! merge the two solutions by a max operation
  cur_values = max(cur_values, temp_values)
end do
```

Algorithm 2: Convex envelope algorithm with dual grid partition.

dual grid has size $(1000s) \times (1000s)$. As a comparison, we have included the results obtained with uniform dual grids of size 1000×1000 (upper left) and $(1000s) \times (1000s)$ with $s = 100$ (bottom right); for the latter plot, Algorithm 2 has been used with $s = 100$. Like f^2_{test}, it is possible to compute with great accuracy the exact convex envelope of f^3_{test} and thus its exact non-convexity region; although this region is different from the non-convexity region of $\left(f^3_{\text{test}}\right)_{\Omega_N}$ (which is the one we are actually approximating), we compare our results against it. Then, the distance between the computed and "exact" non-convexity region (mapped on the same finite grid), measured as the percentage of points where they differ, is respectively 44.1%, 5.23%, 1.36%, 0.40%, 0.32% and 1.17%. Thus, for a non-uniform dual grid with $s = 10$, the result is of the same quality as with a uniform dual grid with $s = 100$.

As a final test, we consider the following function introduced in [17]:

$$g_{\alpha,\theta}(x,y) = \frac{1}{2}(\cos^2\theta + \alpha\sin^2\theta)x^2 + (1-\alpha)\cos\theta\sin\theta xy + \frac{1}{2}(\alpha\cos^2\theta + \sin^2\theta)y^2.$$

We take $\alpha = -\frac{1}{2}$, $\theta = \frac{1}{2}\arctan\frac{1}{3}$ and choose as the domain the square $[-1,1]^2$ discretized by a uniform 1000×1000 grid; the resulting function and its convex envelope computed by the Qhull library using the same primal grid are shown in Figure 10. We report in Table 4 the differences between the various algorithms and the solution given by Qhull. We use the suffixes xy and yx to denote the two different orderings for the alternating factorization algorithm. Moreover, we denote by the suffix hN with some integer N (respectively uN) the application of Algorithm 2 with $s = N$ and a heuristic (respectively uniform) grid; we have included only one value of s since in this case higher values do not improve the results significantly. We observe that since the test function $g_{-\frac{1}{2},\frac{1}{2}\arctan\frac{1}{3}}$ is not invariant under exchange of the axes, different orderings produce different results; which ordering works best depends on the particular func-

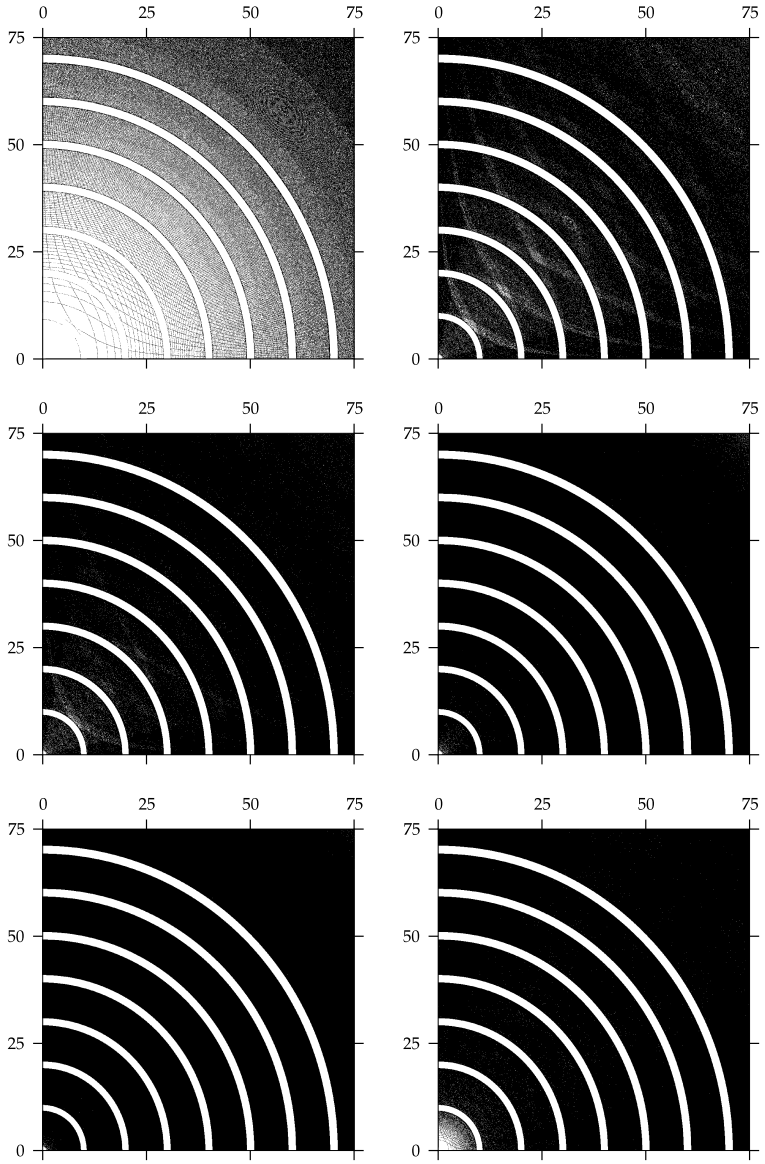

Figure 9: Plots of the non-convexity regions of the test function f^3_{test} on the domain $[-75,75]^2$ covered by a uniform 1000×1000 grid. The upper left plot is produced with a uniform dual grid of size 1000×1000, while for the bottom right plot a uniform dual grid of size $(1000s) \times (1000s)$ with $s = 100$ is used; for the other plots, a heuristic non-uniform dual grid is used with s, respectively from left to right and from top to bottom, equal to 2, 3, 10 and 20.

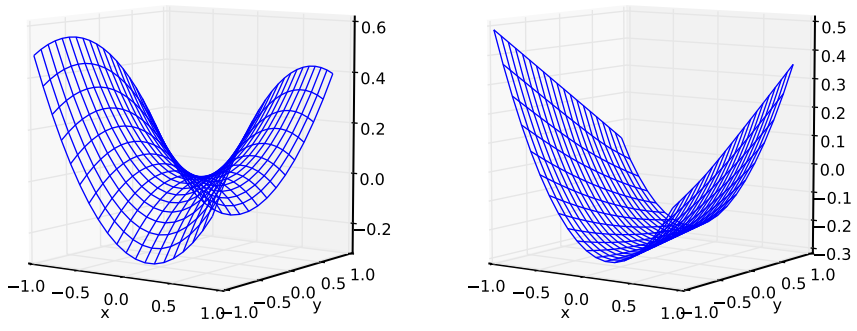

Figure 10: Plots of the function $g_{-\frac{1}{2},\frac{1}{2}\arctan\frac{1}{3}}$ (left) and its convex hull (right) on $[-1,1]^2$.

tion considered. In the general case in which the size of the dual grid differs from that of the primal grid, also the computational speed is not the same for the two orderings. However, in our case we have chosen to take the two grids with the same size and thus in theory there is no difference in computational time; the time difference observed in practice is also negligible. Furthermore, as expected both alternating factorization algorithms perform better than the standard one, but they are outperformed by the max-alternating one. Using Algorithm 2 leads to better results at the expense of computational time. In this case, the gain due to a uniform grid is modest, while the heuristic approach performs extremely well even for s very small. This is due to the fact that the heuristic algorithm is able to build a dual grid which is extremely close to the optimal one. However, this is not what happens in general, and the present observation results from the particular shape of the test function. Moreover, the choice of the factorization direction affects the heuristic algorithm more than the uniform one. In this test case, the high precision observed in the symmetrized heuristic algorithm is due to the xy ordering. In the yx case, the convex hull of the sections along the y direction are all affine functions, which means that their derivative is constant and thus the dual grid associated is composed by only one point; even by examining s sections, we can at most build a dual grid with s points, resulting in an inaccurate approximation.

5 Application to phase separation in ionic solutions

Ionic solutions consist of a solvent, usually water, in which one or more ionic species are dissolved. The behavior of ionic solutions interacting with, or confined by, solid objects carrying surface charges is of interest in many natural and technological contexts. One of the first steps to elucidate the behavior of such solutions is the study of bulk ionic solutions in the absence of external forcing. The non-ideal behavior of bulk ionic solutions results mainly from two types of effects, which both play a major role as the ionic density increases: long-range electrostatic correlations due to screening by counter-ions and short-range steric correlations due to excluded volume effects (see [2]

Table 4: Differences for the convex envelope of $g_{-\frac{1}{2},\frac{1}{2}\arctan\frac{1}{3}}$ between the result produced by the Qhull library and those of the algorithms presented in this paper. Variations in computational time are expressed as fraction of the time taken by the ma-dDLFT algorithm.

	s-dDLFT	a-dDLFT-xy	a-dDLFT-yx
Maximum value	10^{-4}	$1.2\cdot10^{-6}$	$8.6\cdot10^{-5}$
Mean value	$2.3\cdot10^{-5}$	$3.2\cdot10^{-7}$	$9.8\cdot10^{-7}$
Standard deviation	$1.6\cdot10^{-5}$	$2.6\cdot10^{-7}$	$4.6\cdot10^{-6}$
Time	-34.5%	-49.9%	-50.1%
	ma-dDLFT	ma-dDLFT-h3	ma-dDLFT-u10
Maximum value	$1.2\cdot10^{-6}$	$6.8\cdot10^{-16}$	$1.2\cdot10^{-7}$
Mean value	$2.2\cdot10^{-7}$	$5.0\cdot10^{-17}$	$1.7\cdot10^{-8}$
Standard deviation	$1.9\cdot10^{-7}$	$5.3\cdot10^{-17}$	$1.7\cdot10^{-8}$
Time	0.544 s	$+92.6\%$	$+527.2\%$

and references therein). These non-ideal terms result from ion-ion interactions in the solution; ion-solvent interactions are neglected. Liquid-vapor transition and criticality in bulk ionic solutions due to non-ideal behavior have been extensively investigated over the past decades; see, e.g., [9].

We consider two dissolved ionic species, a cation and an anion; we denote their valences by $Z_+ > 0$ and $Z_- < 0$, respectively. Within the so-called Primitive model considered herein, both ions have the same diameter σ. We denote by $\rho = (\rho_+, \rho_-)$ the ionic densities, and we introduce the reduced ionic densities $\hat{\rho}_\pm = \sigma^3\rho_\pm$. The bulk free energy of the ionic solution splits into $f(\rho) = f_{\rm id}(\rho) + f_{\rm corr}(\rho)$, where the ideal term is given by

$$f_{\rm id}(\rho) = \frac{1}{\beta\sigma^3}\sum_{i=\pm}\hat{\rho}_i(\log(\hat{\rho}_i) - 1),$$

with $\beta := (k_{\rm B}T)^{-1}$ where $k_{\rm B}$ is the Boltzmann constant and T the temperature. The non-ideal term $f_{\rm corr}(\rho)$ is detailed in [12], which is devoted to the study of phase separation in ionic solutions in confined situations. For the present purpose, it suffices to know that $f_{\rm corr}(\rho)$ only depends on the non-dimensional reduced temperatures

$$T^*_\pm = k_{\rm B}T\frac{4\pi\varepsilon\sigma}{Z^2_\pm e^2},$$

where ε is the solvent dielectric permittivity and e the elementary charge. Moreover, due to excluded volume effects, $f_{\rm corr}(\rho)$ becomes unbounded whenever the packing number $\xi(\rho) = \frac{\pi}{6}\hat{\rho}_{\rm tot}$, with reduced total ionic density $\hat{\rho}_{\rm tot} = \hat{\rho}_+ + \hat{\rho}_-$, reaches unity. Therefore, the domain of f is $\{\rho \in \mathbb{R}^2_+; \xi(\rho) < 1\}$. When the reduced temperatures $T^*_\pm$ are large enough, the bulk free energy density f is a convex function of the ionic densities. The minimization of the free energy under canonical constraints fixing the mean-values of the ionic densities in this regime has been investigated in [7] in confined situations.

Herein, we are interested in the regime where at least one of the reduced temperatures $T^*_\pm$ falls below the critical value $T^*_{\rm crit} \approx 0.07857$, so that f is no longer convex in ρ. We observe that the convexity properties of the univariate restrictions $f_+(\rho_+) = f(\rho_+, 0)$ and $f_-(\rho_-) = f(0, \rho_-)$ are solely determined by the value of the corresponding reduced temperature $T^*_\pm$. For instance, f_+ (resp., f_-) is convex in ρ_+ (resp., ρ_-)

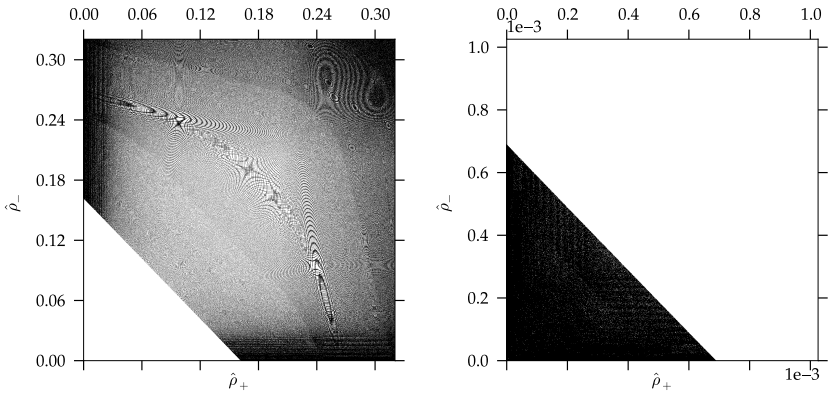

Figure 11: Non-convexity region of the bulk free energy for $Z_\pm = \pm 2$ and $T^* = 0.07$; the black points are where $f = \operatorname{conv} f$. The right panel provides a zoom of the left panel near the origin.

if $T_+^* \geq T_{\text{crit}}^*$ (resp., $T_-^* \geq T_{\text{crit}}^*$) and non-convex otherwise. Another relevant property is that, in general (see below for a counter-example), the bulk free energy density f is convex for high enough ionic densities because steric correlations, which always yield a convex contribution, become dominant.

Firstly, we study the case $Z_\pm = \pm 2$. In the symmetric case where $Z_+ + Z_- = 0$, it turns out that f_{corr} only depends on $\hat{\rho}_{\text{tot}}$, which ensures that f and its convex envelope are invariant when swapping ionic densities. Moreover, the two reduced temperatures $T_\pm^*$ are equal, and we denote by T^* their common value. We show the shape of the non-convexity region in the case $T^* = 0.07$ (below the critical value $T_{\text{crit}}^* \approx 0.07857$) in Figure 11; as expected this region is invariant when swapping ionic densities. We notice that since f is convex for sufficiently large ionic densities, we can restrict the computation of the convex envelope to a set smaller than the actual domain of f (in this case we have chosen the set $[0, 0.032]^2$ for the reduced ionic densities). Moreover, the dual grid is uniform.

Now, we consider the non-symmetric case $Z_+ = +2$ and $Z_- = -1$. The non-convexity regions are plotted in Figure 12 for $T_+^* = 0.07$ and $T_+^* = 0.0196$. As expected, both regions are no longer invariant when swapping ionic densities. Moreover, for $T_+^* = 0.07$, the non-convexity region does not divide the state space into two connected components (contrary to the above symmetric case). The reason for this is easily understood by considering the univariate restrictions f_+ and f_- defined above. Indeed, $T_+^* = 0.07$ falls below the critical value T_{crit}^* so that the non-convexity region of f touches the ρ_+-axis, whereas $T_-^* = 4T_+^*$ lies above the critical value so that the non-convexity region does not intersect the ρ_--axis. Instead, for $T_+^* = 0.0196$, T_-^* also falls below the critical value so that the non-convexity region also intersects the ρ_--axis. This is illustrated by the zoom near the origin provided by the plots in the second row of Figure 12. As a further illustration, we consider the case $Z_+ = +3$ and $Z_- = -2$. The non-convexity region is presented in Figure 13 for $T_+^* = 0.03499$ (left panel) and $T_+^* = 0.03491$ (right panel). In both cases, T_*^+ falls below the critical value T_{crit}^*, but only in the second case, T_-^* falls (slightly) below T_{crit}^*.

Finally, we consider the case $Z_+ = +3$, $Z_- = -1$, and $T_+^* = 0.0266$. For these values of the parameters, the region of non-convexity covers an area so large that it

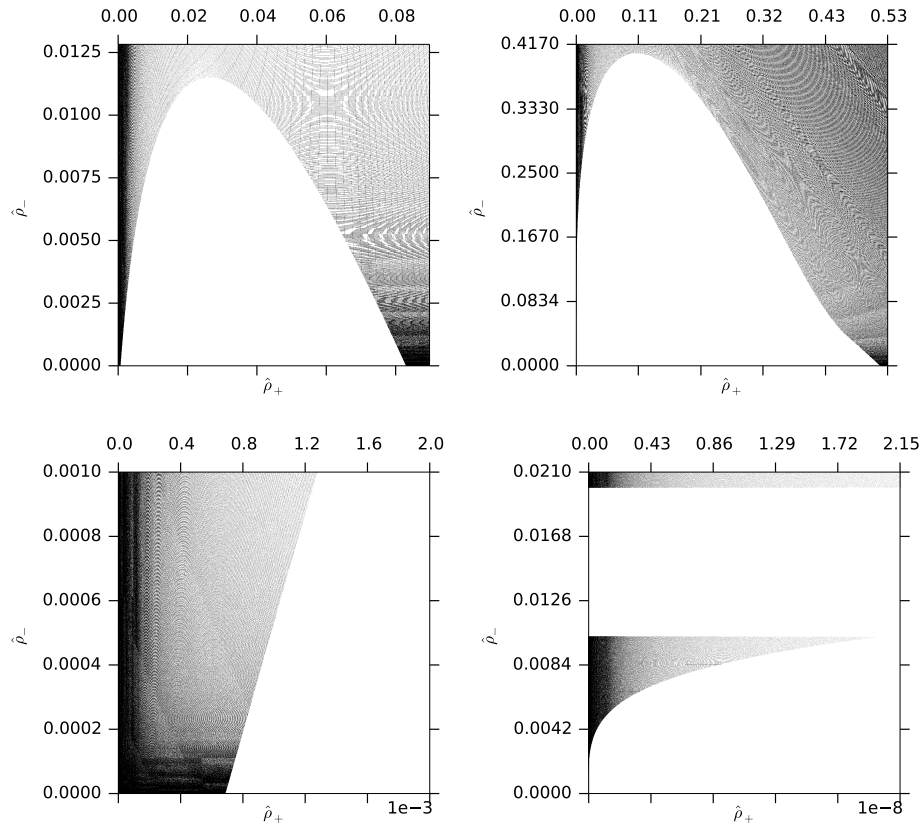

Figure 12: Non-convexity region of the bulk free energy for $Z_+ = +2$, $Z_- = -1$, and $T_+^* = 0.07$ (left) or $T_+^* = 0.0196$ (right). The second row provides a zoom of the corresponding panels from the upper row near the origin.

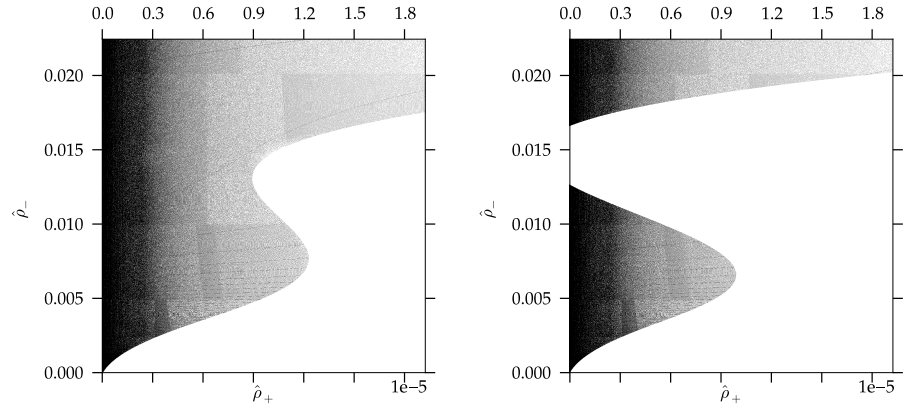

Figure 13: Non-convexity region of the bulk free energy for $Z_+ = +3$, $Z_- = -2$, and $T_+^* = 0.03499$ (left panel) or $T_+^* = 0.03491$ (right panel).

is close to the boundary of the domain of the free energy density, where its values and derivatives go rapidly to infinity. Thus, a uniform discretization to build the dual grid fails to produce any meaningful result; even when using Algorithm 2 to deal with dual grids of extremely large size (we have tried $s = 1000$ with a primal grid of size 5000×5000, i.e., with a dual grid of size $(5 \times 10^6) \times (5 \times 10^6)$), the results are not satisfactory. The heuristic non-uniform dual grid constructed in Section 4.5 yields instead a more accurate approximation, where the border of the non-convexity region is better resolved (see Figure 14).

6 Conclusions

In this work, we have studied convex envelope algorithms based on the double discrete LFT. In addition to the standard factorization algorithm presented in [13] and [10], we have proposed the alternating factorization algorithm which we have shown to be more efficient than the standard factorization one for bivariate functions. Additional improvements are achieved by the max-alternating factorization and by the correct handling of domains with arbitrary shape (see Appendix A for the latter). Moreover, we have emphasized theoretically and numerically that the quality of the resulting approximations given by both the standard and alternating algorithms depends on the choice of the dual grid. This grid is not given as an input and must be generated. The simplest solution to this problem, i.e., a uniform discretization, is not always sufficient to obtain accurate results. In order to address this important issue, we have presented an efficient method to handle dual grids of arbitrary length and we have devised a computationally-effective approach for generating non-uniform dual grids. Finally, we have applied the present algorithms to the practically-relevant study of phase separation in ionic solutions consisting of cations and anions dissolved in a solvent, confirming the validity of the approach.

With an eye toward future developments, we observe that in certain situations it could be necessary to obtain a characteristic function for the non-convexity region. This could allow a more efficient storage of the envelope data in the non-convexity region, whereas for the values outside the region, the analytical expression of the function f could be used. The non-convexity regions resulting from the application of a threshold to the difference $(f - \operatorname{conv} f)$ have often a shrinking and slightly different shape. In order to avoid this, if the region can be already recognized well from the plot of the binary matrix $f \neq \operatorname{conv} f$ (as it has been the case up to now), it is possible to use the closing operation from mathematical morphology (see [20]): the closing is applied to the binary matrix with structuring elements of increasing size until the resulting matrix has the correct number of connected components.

A Treatment of non-rectangular domains

The double discrete LFT algorithms operate naturally on rectangular computational domains discretized by a Cartesian grid. But since every function can be extended by setting its value to $+\infty$ where it is not defined, we should be able to treat inside this framework functions defined on sets of arbitrary shape. Unfortunately, as observed in Remark 6, the double discrete LFT is always finite everywhere and thus every information about the domain shape is lost.

The problem lies in the fact that the information about the domain shape is encoded

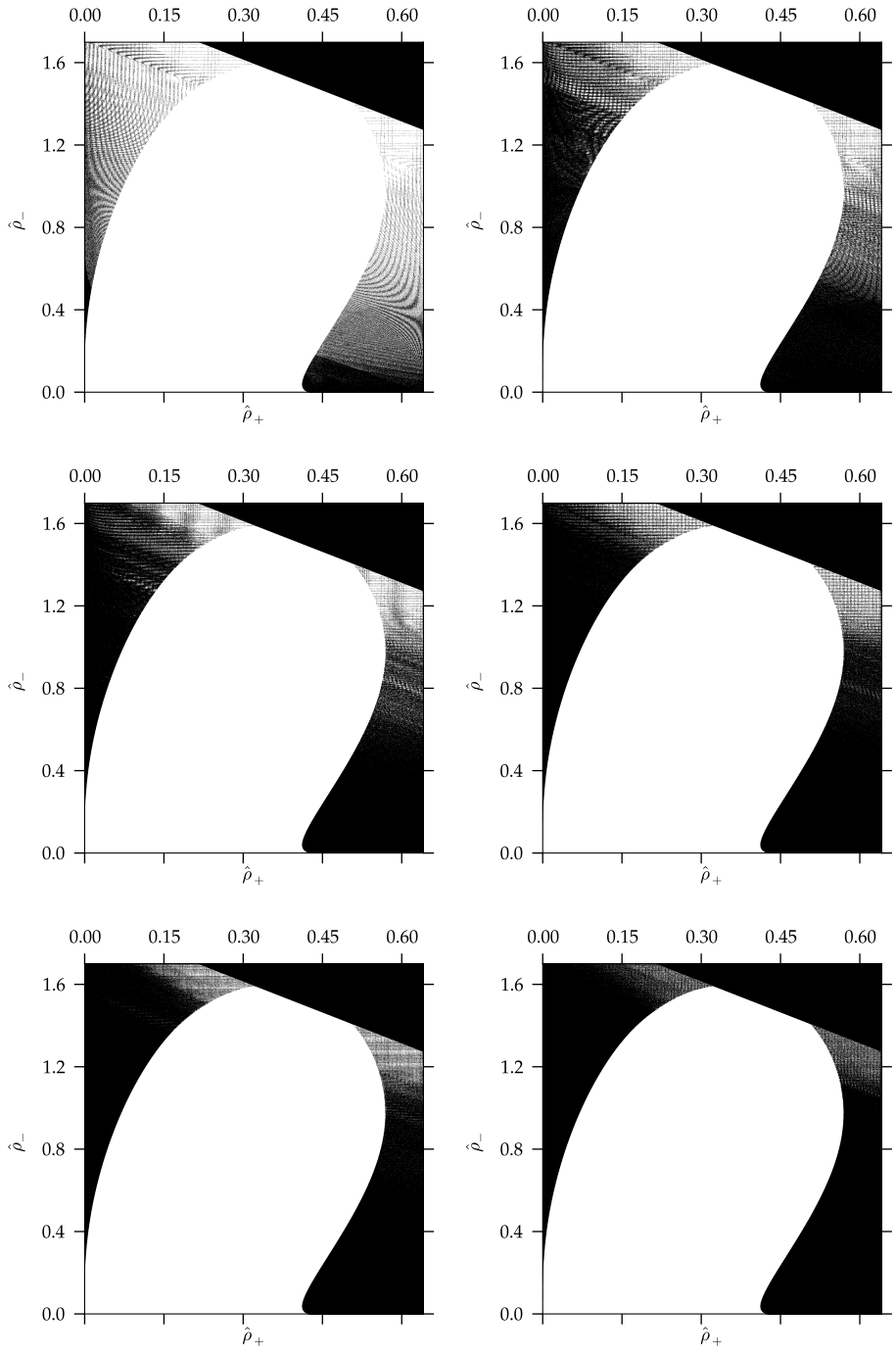

Figure 14: Non-convexity region of the bulk free energy for $Z_+ = +3$, $Z_- = -1$, and $T_+^* = 0.0266$. The values of s used are, respectively from left to right and from top to bottom, 2, 5, 10, 20, 30 and 100. The black triangular zone in the upper right corner of each plot lies outside the domain of the bulk free energy.

in the behavior at infinity of the LFT. Consider for example a one-dimensional function f which is finite on the interval $[l,r]$ and $+\infty$ elsewhere. Its discrete LFT $f^*_{X_N}$ is a piecewise linear function whose domain is the whole real line and whose first and last slopes, which extend all the way to $\mp\infty$, are respectively l and r. If we apply the continuous LFT to $f^*_{X_N}$, we obtain again a function which is defined on $[l,r]$ and $+\infty$ elsewhere. On the other hand, by applying a discrete LFT to $f^*_{X_N}$, we use its values in only a finite number of points and suppose that it is $+\infty$ elsewhere, which produces again a function defined on the whole real line. In order to reconstruct the correct domain, it is sufficient to keep in memory the values of the external slopes of $f^*_{X_N}$.

In the two-dimensional case, we similarly have to preserve the external slope information of the first discrete LFT in order to obtain the correct domain after the second discrete LFT. Using the modified algorithm simplifies the matter because we only have one set of slopes to consider since there are only two passes of one-dimensional discrete LFTs; however, because of the pass of convex envelopes between them, these slopes change before the last pass. This is reasonable since the convex envelope can make the domain of finiteness of a function grow.

Using the notation of Section 4.2, in order to compute the slopes after the convex envelope pass, we have to determine which points of Y_N belong asymptotically for $\xi \to \pm\infty$ to the convex envelope $\operatorname{conv}\left[(-g(\xi,\cdot))_{Y_N}\right]$. This can be done in a similar way to the computation of the standard one-dimensional convex envelope by taking into account also the slopes obtained in the first pass of discrete LFTs. We explain explicitly only the case $\xi \to \infty$; the other case can be treated in exactly the same way by inverting the ξ-axis. Let y_i, $i = 1,\dots,N$ be the points of the grid Y_N. Fixed i, the function $-g(\cdot,y_i)$ is a piecewise linear function whose slope has a constant value s_i after a certain value ξ_i on the ξ-axis. Take $\xi_0 = \max_i \xi_i$. For simplicity, given $\xi > \xi_0$, denote the value $-g(\xi,y_i)$ by $z_i(\xi)$; we have that $z_i(\xi) = z_i(\xi_0) + s_i(\xi - \xi_0)$ and that the slopes $c_{ij}(\xi)$ between the nodes x_i and x_j are

$$c_{ij}(\xi) = \frac{z_j(\xi) - z_i(\xi)}{x_j - x_i} = \frac{z_j(\xi_0) - z_i(\xi_0) + (s_j - s_i)(\xi - \xi_0)}{x_j - x_i}. \tag{5}$$

The convex hull of the piecewise linear function having nodes y_i and values $z_i(\xi)$ with $i = 1,\dots,N$ is given by interpolating only a subset of the nodes. This subset depends on ξ, but we will now show that there exists $\xi_r \geq \xi_0$ such that the subset remains constant for $\xi \geq \xi_r$; we will denote by i^*_k, $k = 1,\dots,M$, with $M \leq N$, these nodes.

We have that $i^*_1 = 1$ since the first node always belongs to the convex hull nodes for every ξ (similarly we also have $i^*_M = N$). Given i^*_k, the next value i^*_{k+1} can be computed in the following way. Neglecting the grid points x_i with $i < i^*_k$ and supposing that $x_{i^*_k}$ is in the convex hull, the next node in the convex hull at ξ is $\arg\min_{j=i^*_k+1,\dots,N} c_{i^*_k j}(\xi)$. Since $c_{i^*_k j}(\xi)$ are affine as functions of ξ, there is $\xi^*_{k+1} \geq \xi_0$ such that the minimizer will be the same for all $\xi > \xi^*_{k+1}$. This value is i^*_{k+1} and from (5) we have that $i^*_{k+1} = \arg\min_{j=i^*_k+1,\dots,N} \frac{s_j - s_{i^*_k}}{x_j - x_{i^*_k}}$. In the case of the minimum being assumed by more than one value of j we choose the one for which $\frac{z_j(\xi_0) - z_{i^*_k}(\xi_0)}{x_j - x_{i^*_k}}$ is the smallest; in case of a further draw, we can take the point for which x_j is larger. If $i^*_{k+1} = N$, then $M = k+1$ and all values have been computed.

The value ξ^*_{k+1} is the first value of ξ for which

$$c_{i^*_k i^*_{k+1}}(\xi) \le c_{i^*_k j}(\xi) \quad \text{for every } j > i^*_k;$$

by substituting (5) and solving these linear inequalities we obtain that

$$\xi^*_{k+1} = \xi_0 + \max_{\substack{j>i^*_k \\ j\neq i^*_{k+1}}} \frac{\frac{z_{i^*_{k+1}}(\xi_0)-z_{i^*_k}(\xi_0)}{x_{i^*_{k+1}}-x_{i^*_k}} - \frac{z_j(\xi_0)-z_{i^*_k}(\xi_0)}{x_j-x_{i^*_k}}}{\frac{s_j-s_{i^*_k}}{x_j-x_{i^*_k}} - \frac{s_{i^*_{k+1}}-s_{i^*_k}}{x_{i^*_{k+1}}-x_{i^*_k}}}.$$

Taking $\xi_r = \max_{k=1,\dots,M} \xi^*_k$ (where we put $\xi^*_1 = \xi^*_M = \xi_0$), we have that the convex hull nodes are given by $(i^*_k)_{k=1,\dots,M}$ for all $\xi > \xi_r$. The functions $\operatorname{conv}\left[(-g(\xi,\cdot))_{Y_N}\right](y_i)$, $i = 1,\dots,N$, when $\xi > \xi_r$ are given by the formulas

$$\begin{cases} z_i(\xi) & \text{if } i = i^*_k \text{ for some } k, \\ \alpha z_{i^*_k}(\xi) + (1-\alpha) z_{i^*_{k+1}}(\xi) & \text{if } i^*_k < i < i^*_{k+1} \text{ for some } k; \end{cases}$$

by substituting the z_i, we can easily get their slopes as functions of s_i.

In this way we obtain the interval $[\xi_l, \xi_r]$ outside of which the points of Y_N belonging to the convex envelope do not change anymore and the external slopes of $\operatorname{conv}\left[(-g(\xi,\cdot))_{Y_N}\right](y)$ as a function of ξ for every $y \in Y_N$. Then, by discretizing $[\xi_l, \xi_r]$ and using the obtained slopes, we can compute the final pass of discrete LFTs recovering the correct shape of the domain. We finally observe that this improvement to the algorithm can also be seen as a further enlargement of the dual set; in particular, the function we are computing is $\left(f^*_{\Omega_N}\right)^*_{(]-\infty,\xi_l[\cup C_M\cup]\xi_r,+\infty[)\times\mathbb{R}}$, where C_M is a discretization of $[\xi_l, \xi_r]$; by applying the symmetrization, we can then compute the double LFT with a dual set S_M such that $\mathbb{R}^2 \setminus S_M$ is bounded.

References

[1] Barber, C., Dobkin, D., Huhdanpaa, H.: The Quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software **22**(4), 469–483 (1996). DOI 10.1145/235815.235821. URL http://www.qhull.org

[2] Blum, L.: Mean spherical model for asymmetric electrolytes. Molecular Physics **30**(5), 1529–1535 (1975). DOI 10.1080/00268977500103051

[3] Brenier, Y.: Un algorithme rapide pour le calcul de transformées de Legendre-Fenchel discrètes. Comptes Rendus des Séances de l'Académie des Sciences. Série I. Mathématique **308**(20), 587–589 (1989)

[4] Chan, T.: Optimal output-sensitive convex hull algorithms in two and three dimensions. Discrete & Computational Geometry **16**(4), 361–368 (1996). DOI 10.1007/BF02712873

[5] Contento, L.: The Discrete Legendre-Fenchel Transform and its application to phase separation in electrolytes. Master's thesis, University of Udine (2012). URL http://hal.archives-ouvertes.fr/hal-00806085

[6] Corrias, L.: Fast Legendre–Fenchel Transform and Applications to Hamilton–Jacobi Equations and Conservation Laws. SIAM Journal on Numerical Analysis **33**(4), 1534–1558 (1996). DOI 10.1137/S0036142993260208

[7] Ern, A., Joubaud, R., Lelièvre, T.: Mathematical study of non-ideal electrostatic correlations in equilibrium electrolytes. Nonlinearity **25**(6), 1635–1652 (2012). DOI 10.1088/0951-7715/25/6/1635

[8] Gardiner, B., Lucet, Y.: Convex Hull Algorithms for Piecewise Linear-Quadratic Functions in Computational Convex Analysis. Set-Valued and Variational Analysis **18**(3-4), 467–482 (2010). DOI 10.1007/s11228-010-0157-5

[9] Groh, B., Evans, R., Dietrich, S.: Liquid-vapor interface of an ionic fluid. Physical Review E **57**(6), 6944–6954 (1998). DOI 10.1103/PhysRevE.57.6944

[10] Helluy, P., Mathis, H.: Pressure laws and fast Legendre transform. Mathematical Models and Methods in Applied Sciences **21**(4), 745–775 (2011). DOI 10.1142/S0218202511005209

[11] Hiriart-Urruty, J.B.: Lipschitz *r*-continuity of the approximate subdifferential of a convex function. Mathematica Scandinavica **47**, 123–134 (1980)

[12] Joubaud, R., Bernard, O., Contento, L., Ern, A., Rotenberg, B., Turq, P.: Phase separation in highly charged confined ionic solutions. Tech. rep., University Paris-Est, CERMICS (2013). URL http://hal.archives-ouvertes.fr/hal-00806218

[13] Lucet, Y.: Faster than the Fast Legendre Transform, the Linear-time Legendre Transform. Numerical Algorithms **16**(2), 171–185 (1997). DOI 10.1023/A:1019191114493

[14] Lucet, Y.: What Shape Is Your Conjugate? A Survey of Computational Convex Analysis and Its Applications. SIAM Review **52**(3), 505–542 (2010). DOI 10.1137/100788458

[15] Maxwell, J.: On the Dynamical Evidence of the Molecular Constitution of Bodies. Nature **11**, 357–359 (1875). DOI 10.1038/011357a0

[16] Oberman, A.: The convex envelope is the solution of a nonlinear obstacle problem. Proceedings of the American Mathematical Society **135**(6), 1689–1694 (2007). DOI 10.1090/S0002-9939-07-08887-9

[17] Oberman, A.: Computing the convex envelope using a nonlinear partial differential equation. Mathematical Models and Methods in Applied Sciences **18**(5), 759–780 (2008). DOI 10.1142/S0218202508002851

[18] Patrinos, P., Sarimveis, H.: Convex parametric piecewise quadratic optimization: Theory and algorithms. Automatica **47**(8), 1770–1777 (2011). DOI 10.1016/j.automatica.2011.04.003

[19] Rockafellar, R.: Convex Analysis. Princeton University Press (1970)

[20] Serra, J.: Image Analysis and Mathematical Morphology. Academic Press (1983)