



UNIVERSITÀ  
DEGLI STUDI  
DI UDINE

## Università degli studi di Udine

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

*Original*

*Availability:*

This version is available <http://hdl.handle.net/11390/865092> since 2021-03-18T12:37:28Z

*Publisher:*

*Published*

DOI:10.1007/s10589-014-9666-8

*Terms of use:*

The institutional repository of the University of Udine (<http://air.uniud.it>) is provided by ARIC services. The aim is to enable open access to all the world.

*Publisher copyright*

(Article begins on next page)

# An efficient algorithm for the double Legendre–Fenchel transform with application to phase separation

Lorenzo Contento,<sup>\*</sup> Alexandre Ern,<sup>†</sup> Rossana Vermiglio<sup>\*</sup>

March 31, 2013

## Abstract

We study the double discrete Legendre–Fenchel Transform (LFT) to approximate the convex hull 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 by Corrias. 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 hull. 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. 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 ionic solutions where non-ideal effects due to long-range electrostatic and short-range steric correlations between ions play an important role.

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, the system at equilibrium is not in a state lying in the non-convexity region of the potential, i.e., the region where the potential and its convex hull 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 ([9]); he was able to 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 hull.

For more complex potentials, e.g., bivariate or multivariate functions, the convex hull cannot be computed analytically. Since the convex hull results

---

<sup>\*</sup>Department of Mathematics and Computer Science, University of Udine, Italy

<sup>†</sup>Université Paris-Est, CERMICS, Ecole des Ponts ParisTech, 77455 Marne la Vallée Cedex 2, France. ern@cermics.enpc.fr

from a double Legendre–Fenchel Transform (LFT), it can be approximated by a double discrete LFT. This approach has been considered in [5] in view of deriving pressure laws in binary mixtures. The main ingredient of the double discrete LFT is obviously the discrete LFT, which can be computed, as in [5], using Lucet’s algorithm [8]. 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.

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 [2]). Our convergence results on the double discrete LFT are based on [2] which addresses the convergence of the discrete LFT. However, in order to prove convergence to the convex hull and better understand the behavior of the algorithms, it is important to study the grid on which the second discrete LFT is applied, which we call *dual grid*. The choice of the dual grid is one of the main issues regarding an accurate approximation of the convex hull, an issue which is often only briefly discussed, as for example happens in [5].

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. The same change could be made in the multivariate case, but the gain would be marginally inferior. 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 again highlighted, and an efficient handling of dual grids of arbitrary length is investigated.

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). We consider the model presented in [7] 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.

## 1 Preliminaries

### 1.1 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 hull* of  $f$ , i.e., the greatest (smallest) lower (upper) semi-continuous function upper-bounded (lower-bounded) by  $f$ , and by  $\text{conv } f$  the *convex hull* of  $f$ , i.e., the greatest convex function upper-bounded by

$f$ .

Let  $g$  be a convex function. We denote by  $\text{dom } g$  its (*effective*) *domain*, i.e., the subset of  $\mathbb{R}^n$  where  $g < +\infty$ ; we say that  $g$  is *proper* if  $\text{dom } g$  is non-empty and  $g$  never takes the value  $-\infty$ . Moreover, we denote by  $\text{cl } g$  the *closure* of  $g$ , which is  $g$  if  $g$  is proper and the constant function  $-\infty$  otherwise;  $g$  is said to be *closed* if  $g = \text{cl } g$ . Given a point  $x \in \mathbb{R}^n$  and a vector  $\xi \in \mathbb{R}^n$ , we say that  $\xi$  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  $\partial g(x)$ .

Lucet's algorithm for the convex hull is based on the *Legendre–Fenchel Transform (LFT)*, defined as follows.

**Definition 1.** Given  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  not necessarily convex, the 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 [10]; 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  $f^* \geq g^*$ ;  $f^{**} = \text{cl}(\text{conv } f)$ ; for  $n \geq 2$ , the LFT can be factorized along each dimension as  $f^* = (-\dots(-f^{*1})^{*2}\dots)^{*(n-1)*n}$ , where  $f^{*i}$  is the LFT along the  $i$ th-dimension, i.e.,

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

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

## 1.2 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  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ , we define  $f_\Omega$  as

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

It is clear that

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

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

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

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

In what follows, we implicitly assume that

- (i)  $f(x) < +\infty$  for at least one  $x \in \Omega_N$ , otherwise  $f_{\Omega_N}^*$  would be  $-\infty$  everywhere;
- (ii)  $f(x) > -\infty$  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 [8]. Moreover, owing to the factorization properties of the LFT, this algorithm is the basic building block of discrete LFT algorithms in higher dimensions.

### 1.3 Convergence to the Legendre-Fenchel Transform

Our convergence analysis of the double discrete LFT is based on a convergence result of the discrete LFT by Corrias [2], 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 present statement has been adapted to our notation and extended explicitly to more general sets  $\Omega$ .

**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  $\Omega$  if  $\min_{x' \in \Omega_N} \|x' - x\| \rightarrow 0$  as  $N \rightarrow \infty$  for all  $x \in \Omega$ ; we denote this fact by the notation  $\Omega_N \rightarrow \Omega$ . If, additionally,  $\sup_{x \in \Omega} \min_{x' \in \Omega_N} \|x' - x\| \rightarrow 0$  as  $N \rightarrow \infty$ , we say that the convergence is uniform.

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

*Proof.* The proof is essentially the same as that of [2, Theorem 2.1] and can be found in [1, Theorem 3.27].  $\square$

The hypothesis  $\overline{(f_\Omega)} = f_\Omega$  is necessary in order to make Theorem 4 true for every sequence  $\Omega_N \rightarrow \Omega$ , as shown in [2]. A couple of simple settings in which this hypothesis holds is

- (i)  $\Omega$  is open and  $f|_\Omega$  is upper semi-continuous;
- (ii)  $\Omega \subseteq \text{cl int } \Omega$  and  $f|_\Omega$  is continuous.

The proofs for these two sufficient conditions can be found respectively in Corollary 3.30 and Proposition 3.33 of [1].

## 2 The double Discrete Legendre-Fenchel Transform

### 2.1 Optimal dual set and grid

Given  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  and  $\emptyset \neq \Omega \subseteq \mathbb{R}^n$ , we consider the problem of computing  $\text{conv } f_\Omega$ . Firstly, we discretize this problem by considering a finite set  $\Omega_N \subset \Omega$ ;

the function we actually want to compute is then  $\text{conv } f_{\Omega_N}$ . Secondly, since the application of two successive LFT's to such a function yields its convex hull, we build an approximation of the convex hull by applying two successive discrete LFT's, i.e., taking  $(f_{\Omega_N}^*)_{S_M}^*$ , 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 hull. For any dual grid  $S_M$ ,  $(f_{\Omega_N}^*)_{S_M}^* \neq (f_{\Omega_N}^*)^* = \text{conv } f_{\Omega_N}$  as functions defined on  $\mathbb{R}^n$ ; this is true because by finiteness of  $\Omega_N$ , the domain of  $f_{\Omega_N}^*$  is  $\mathbb{R}^n$ , and by finiteness of  $S_M$ , the convex functions  $f_{\Omega_N}^*$  and  $\text{conv} \left[ (f_{\Omega_N}^*)_{S_M} \right]$  (and thus their LFT's) differ. Nonetheless, we now show that there exists at least a set  $S_M$  which is optimal. In the following statement, the set  $\Omega$  need not be finite.

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

$$(f_{\Omega}^*)_S^*(x) = \text{cl conv } f_{\Omega}(x) \quad (2)$$

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

*Remark 6.* If  $S_M$  is an optimal dual grid, the equality (2) never holds on  $\Omega \setminus \text{dom cl conv } f_{\Omega}$ , since, in that case, the double discrete LFT is finite everywhere.

*Remark 7.* If  $\Omega = \Omega_N$  is finite, there holds  $\text{cl conv } f_{\Omega_N} = \text{conv } f_{\Omega_N}$ , i.e.,  $\text{conv } f_{\Omega_N}$  is closed; see [10, Corollary 19.1.2].

The following Lemma is inspired by [6, Proposition 2.3], from which it follows immediately when  $\Omega$  is a ball centered at the origin; here, we provide a direct proof for more general sets  $\Omega$ .

**Lemma 8.** Let  $g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  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 [10, 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.  $\square$

*Remark 9.* We observe that Lemma 8 does not require the finiteness of  $\Omega$ ; if  $\Omega$  is finite, the converse statement can also be proved [1, Lemma 4.10].

*Remark 10.* The case in which  $\Omega$  is a ball centered at the origin has been considered also in [8, Proposition 2] as a corollary of [6] 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.

We close this section with two important results concerning dual grids: a sufficient condition for this property (Theorem 11) and an existence result of finite optimal dual grids when the set  $\Omega_N$  is itself finite (Theorem 12).

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

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

Then,  $(f_{\Omega}^*)_S^*(x) = \text{cl conv } f_{\Omega}(x)$ . Moreover, if (3) holds for every  $x \in \Omega \cap \text{dom cl conv } f_{\Omega}$ , then  $S$  is an *optimal dual set* with respect to  $\Omega$ .

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

**Theorem 12.** *Consider  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  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  $\Omega_N$ .*

*Proof.* Since  $\Omega_N$  is finite, by [10, Theorem 23.10] we infer  $\partial \text{conv } f_{\Omega_N}(x) \neq \emptyset$  for every  $x \in \Omega_N \cap \text{dom conv } f_{\Omega_N}$ . Thus,  $S_M$  can be built by satisfying the sufficient condition given in Theorem 11; in particular, this yields  $|S_M| = |\Omega_N \cap \text{dom conv } f_{\Omega_N}|$  which is finite.  $\square$

## 2.2 Convergence to the convex hull

We now return to the double discrete LFT  $(f_{\Omega_N}^*)_{S_M}^*$ . We assume that  $\Omega_N$  has been chosen, while the dual grid  $S_M$  has yet to be determined; as seen in the previous section, the best choice is given by an optimal dual grid  $S_M$ . Unfortunately, this is a difficult task since we need to know something about  $\text{conv } f_{\Omega_N}$ . Nonetheless, finding a non-finite optimal dual set  $S$  is generally easier (see for example Section 3.1 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  $S_M \rightarrow S$ , we can achieve convergence of the double discrete LFT to the convex hull.

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

*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  $\Omega_N$ . Hence,  $(f_{\Omega_N}^*)_{S_M}^* \rightarrow (f_{\Omega_N}^*)_S^*$ . The thesis follows since, owing to the optimality of  $S$ , we infer  $(f_{\Omega_N}^*)_S^* = \text{conv } f_{\Omega_N}$  on  $\Omega_N \cap \text{dom conv } f_{\Omega_N}$ .  $\square$

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  $x_0 \in \Omega_N \cap \text{dom conv } f_{\Omega_N}$  and let  $\xi_0 \in \partial \text{conv } f_{\Omega_N}(x_0) \cap S$ . Then,*

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

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

*Proof.* Since  $\xi_0 \in \partial \text{conv } f_{\Omega_N}(x_0) = \partial (f_{\Omega_N}^*)^*(x_0)$ , owing to [10, Theorem 23.5 (b,a\*)], we infer  $(f_{\Omega_N}^*)^*(x_0) = \langle x_0, \xi_0 \rangle - f_{\Omega_N}^*(\xi_0)$ . Moreover, by optimality of

$S$ , we also obtain  $(f_{\Omega_N}^*)^*_S(x_0) = (f_{\Omega_N}^*)^*(x_0)$ . We can then write

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

and the thesis is proved.  $\square$

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

## 3 An algorithm for the convex hull

### 3.1 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  $f : \mathbb{R}^2 \rightarrow \overline{\mathbb{R}}$  and  $\Omega = X \times Y \subset \mathbb{R}^2$ , not necessarily finite. The dimensional factorization (1) yields

$$f_{\Omega}^* = (-f_{\Omega}^{*1})^{*2}.$$

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  $\xi$ , the function  $-g(\xi, \cdot)$  is  $+\infty$  outside of  $Y$ .

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 LFT's. 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 LFT's 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  $\Omega_N$ , we then have in total  $m$  discrete LFT's on data of size  $n$  and  $n$  discrete LFT's 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 [8]. 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  $(f_{\Omega_N}^*)_{S_M}^*$  by applying twice the algorithm just presented; this is the algorithm used in [5], to which we refer henceforth as the “standard factorization” algorithm.

The choice of the dual grid is a crucial issue common to all convex hull 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 3.5. In the case  $\Omega_N = X_N \times Y_N$ , a possible choice of  $S$  is the set  $[\xi^-, \xi^+] \times [\eta^-, \eta^+]$  built as follows. Given  $y \in \mathbb{R}$ , let  $g_y := f_{\Omega_N}(\cdot, y)$ ,  $\xi_y^- := \max \partial \text{conv } g_y(\min X_N)$  and  $\xi_y^+ := \min \partial \text{conv } g_y(\max X_N)$ ; this means that  $\xi_y^-$  and  $\xi_y^+$  are, respectively, the right and left derivative of  $\text{conv } g_y$  at  $\min X_N$  and  $\max X_N$ . We then define  $\xi^-$  as  $\min_{y \in Y_N} \xi_y^-$  and  $\xi^+$  as  $\max_{y \in Y_N} \xi_y^+$ ;  $\eta^-$  and  $\eta^+$  are defined in a similar manner. A proof of this result can be found in [1, Theorem 5.15].

### 3.2 Alternating factorization

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

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

obtaining that

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

where

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

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

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

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

$$(f_{\Omega_N}^*)_S^* \leq (f_{\Omega_N}^*)_{S'}^* \leq (f_{\Omega_N}^*)^* = \text{conv } f_{\Omega_N}$$

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

**Definition 16.** We refer to the algorithm computing  $(f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^*$  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  $(f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^*$  and  $(f_{\Omega_N}^*)_{\mathbb{R} \times D_M}^*$  and set

$$(f_{X_N^2}^*)_{(C_M \times \mathbb{R}) \cup (\mathbb{R} \times C_M)}^* = \max \left\{ (f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^*, (f_{\Omega_N}^*)_{\mathbb{R} \times D_M}^* \right\}. \quad (4)$$

**Definition 17.** We refer to the algorithm computing  $\left(f_{X_N^2}^*\right)_{(C_M \times \mathbb{R}) \cup (\mathbb{R} \times C_M)}^*$  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 3.4 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.

The dual sets used for the double discrete LFT's presented in this section are no longer finite. They can be built following the construction at the end of Section 3.1 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.

### 3.3 Implementation

Let us first briefly recall the standard algorithm for the double discrete LFT. Consider a function  $f : \mathbb{R}^2 \rightarrow \overline{\mathbb{R}}$ , 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  $\Omega_N$ . If the grids  $\Omega_N$  and  $S_M$  have the same size, the matrix which initially contains the values of  $f$  on the grid  $\Omega_N$  can be used for all the successive computations: we substitute each row with its discrete LFT and then do the same for the columns of the matrix so obtained. If the sizes of the grids differ, it is still possible to use one matrix by choosing it sufficiently large to accommodate both grids. 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 LFT's, respectively, along the rows and along the columns, since in one dimension those grids correspond to the slopes of the convex hull. More elaborate choices for the dual grid, which are instrumental when approximating the convex hull of complex functions, are discussed in Section 3.5.

In order to implement the alternating factorization algorithm, we can adapt the standard algorithm just presented by condensing the two consecutive one-dimensional discrete LFT's in the same direction into a single application of the convex hull operation; thus the computation of  $(f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^*$  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 `convexhull_1d(grid, values)` the subroutine which computes the convex hull 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

---

**Algorithm 1** Alternating factorization algorithm for the double discrete LFT

---

**Input:**  $X_N \rightarrow \text{xgrid}$ ,  $Y_N \rightarrow \text{ygrid}$ ,  $f(X_N \times Y_N) \rightarrow \text{values}$

**Output:**  $(f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^* \rightarrow \text{values}$

```

! 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 LFT's 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 hull of each row
parallel do i = 1, size(cgrid)
  pcw = convexhull_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

```

---

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  $(f_{\Omega_N}^*)_{\mathbb{R} \times D_M}^*$ . 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  $(f_{\Omega_N}^*)_{S_M}^* \leq f_{\Omega_N}$ , it is possible to take as the convex hull 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 hull.

*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  $n$ .

### 3.4 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  $\Omega_N$  on which the function is approximated. We 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.,  $r = \sqrt{x^2 + y^2}$ . The first test function we use is

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

whose convex hull is

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

The grid  $\Omega_N$  consists of  $1000 \times 1000$  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 exact convex hull with the regions flagged as non-convex by the different algorithms, i.e., where the computed convex hull differs from the value of the function. 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 hulls 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 hull, and that the ma-dDLFT further improves the results. These successive improvements are also visible from Figure 3 which displays the restriction of the transforms to the axes. Finally, to illustrate Corollary 11, we have also included in Figure 2 the case in which the value 0 is inserted manually into the dual grid; being 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

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

and, by observing that  $\text{conv } f_{\text{test}}^2(x, y) = \text{conv } f_{\text{test}}^2(r)$ , we can build a very accurate approximation of the convex hull of  $f_{\text{test}}^2$  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 hull 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  $1000 \times 1000$  uniform grid. The exact region of non-convexity is an

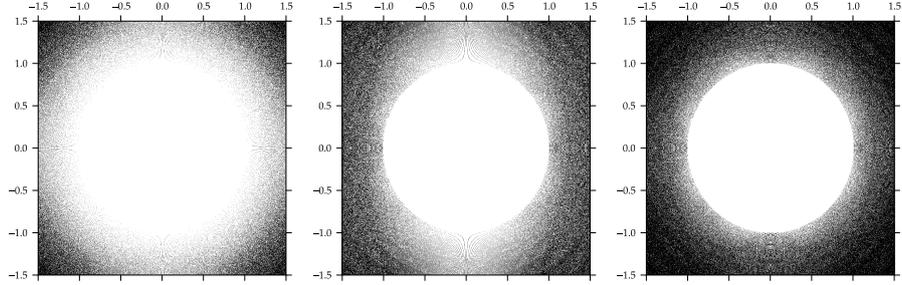


Figure 1: Non-convexity region of the function  $f_{\text{test}}^1$  on the domain  $[-1.5, 1.5]^2$ , computed from left to right by the s-dDLFT, a-dDLFT and ma-dDLFT; the points where the computed convex hull differs from the function  $f_{\text{test}}^1$  are shown in white.

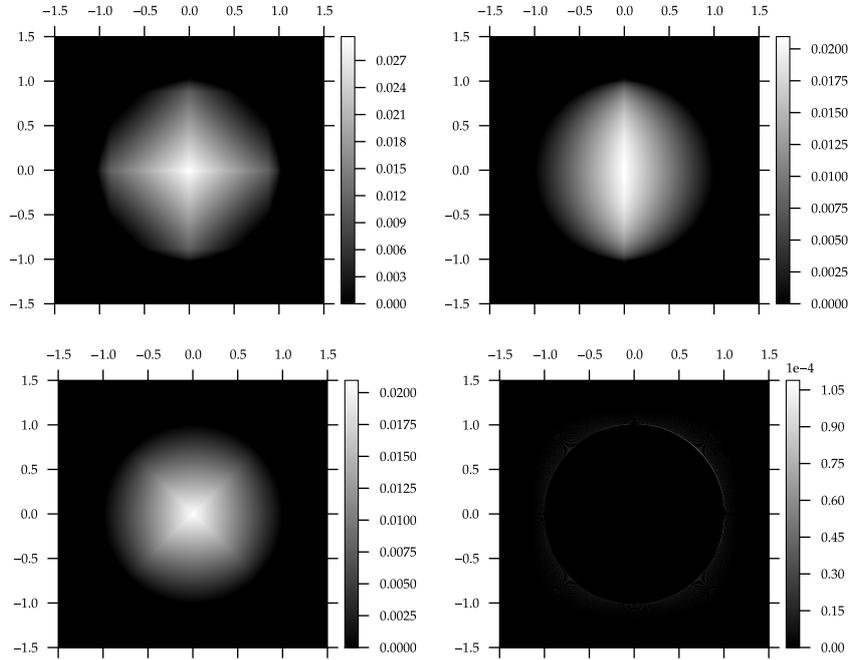


Figure 2: Error between the exact convex hull of  $f_{\text{test}}^1$  and its value computed from left to right and from top to bottom by the s-dDLFT, a-dDLFT, ma-dDLFT and ma-dDLFT using a dual grid containing 0.

on $[-1.5, 1.5]^2$	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%
on 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%

Table 1: Distribution of the error between the exact convex hull of  $f_{\text{test}}^1$  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.

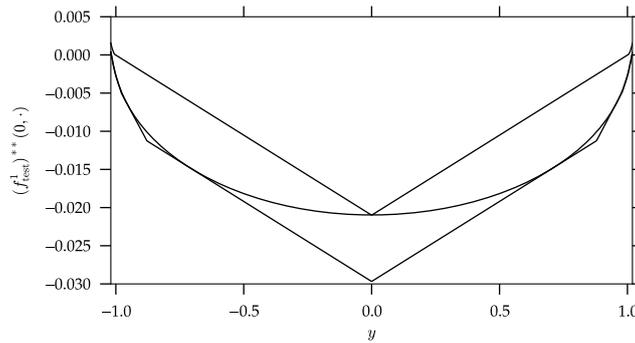


Figure 3: Restriction to the axes of the convex hull of  $f_{\text{test}}^1$  computed respectively by s-dDLFT (topmost line), a-dDLFT (middle line) and ma-dDLFT (bottom line).

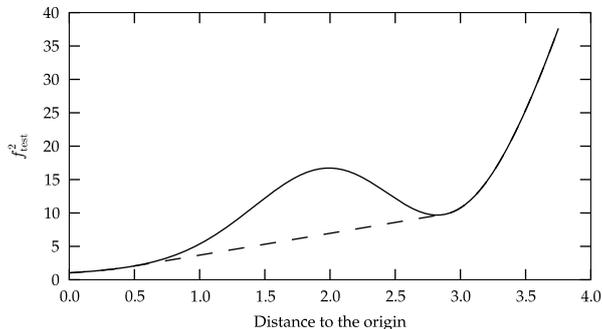


Figure 4: The graph of the test function  $f_{\text{test}}^2$  (solid line) and of its convex hull (dashed line).

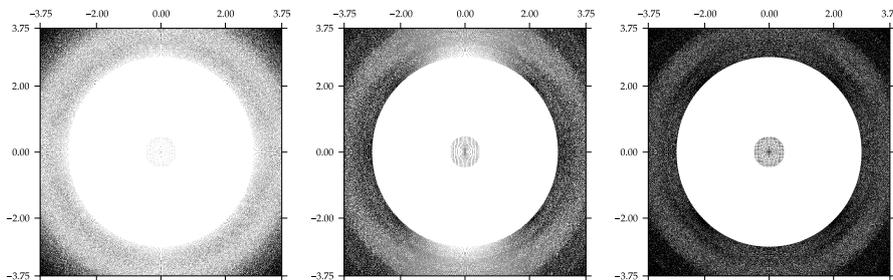


Figure 5: Non-convexity region of the function  $f_{\text{test}}^2$  on the domain  $[-3.75, 3.75]^2$ , computed from left to right through the s-dDLFT, a-dDLFT and ma-dDLFT; the points where the computed convex hull differs from the function  $f_{\text{test}}^2$  are shown in white.

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 plot, we inserted manually into the grid 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  $f_{\text{test}}^2$  than for  $f_{\text{test}}^1$ . 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  $f_{\text{test}}^1$  since the maximum error is reached at the origin which is invariant under swap of coordinates.

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  $1000^2$  grid and the a-dDLFT on grids having size  $(1000 + i)^2$  with  $i = 1, 2, \dots$ ; then, we choose  $i$  as the first value for which the error on the convex hull measured

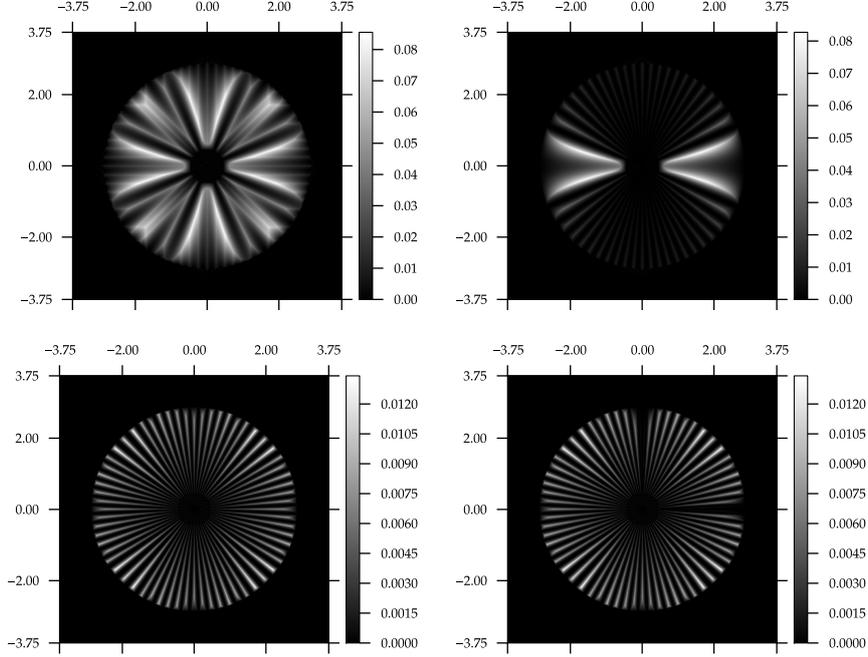


Figure 6: Errors between the exact convex hull of  $f_{\text{test}}^2$  and its value computed respectively from left to right and from top to bottom by the s-dDLFT, a-dDLFT, ma-dDLFT and ma-dDLFT using a dual grid containing  $s$ .

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%
on 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%

Table 2: Distribution of the error between the exact convex hull of  $f_{\text{test}}^2$  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.

	$f_{\text{test}}^1$ (odd $i$ 's are excluded)	$f_{\text{test}}^2$
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

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.

in a certain metric (maximum, 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  $f_{\text{test}}^1$  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  $f_{\text{test}}^1$ , it is much slower when applied to the slightly more complex  $f_{\text{test}}^2$ . 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).

### 3.5 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 hull 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  $f_{\text{test}}^1$  and  $f_{\text{test}}^2$  when enlarging the domain on which the convex hull is computed. Figure 7 illustrates this fact for the second test function. It is interesting to observe that the plots are composed of black curves whose spacing is 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, by Corollary 11 for every point  $(x, y) \in \Omega_N$ , such that  $(\xi, \eta) \in \partial \text{conv } f_{\Omega_N}(x, y)$  for a certain  $\eta \in \mathbb{R}$ , there holds  $\text{conv } f_{\Omega_N}(x, y) = (f_{\Omega_N}^*)_{C_M \times \mathbb{R}}^*$ . Since in our case the test functions and their convex hulls are differentiable, the locus of the points where the first component of the gradient is  $\xi$ , 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

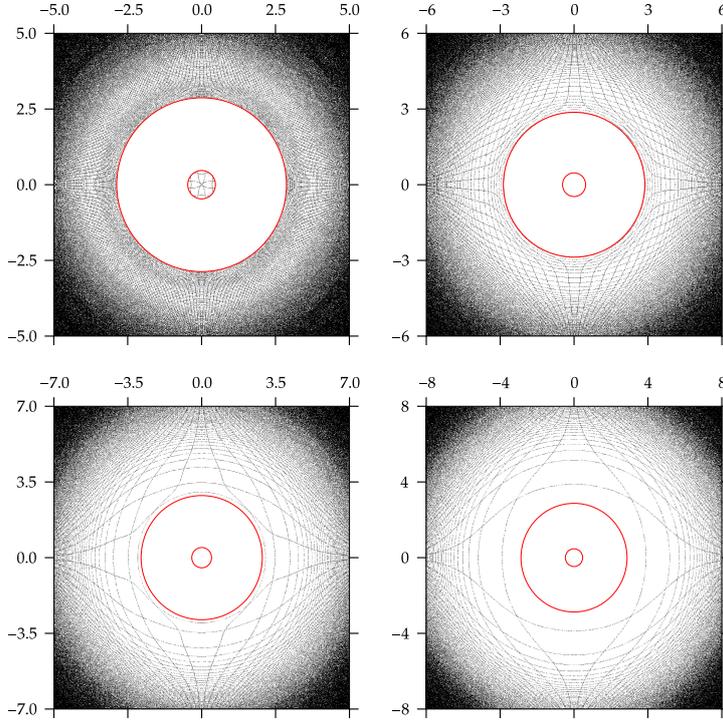


Figure 7: Non-convexity region of the function  $f_{\text{test}}^2$  respectively from left to right and from top to bottom on the domains  $[-5, 5]^2$ ,  $[-6, 6]^2$ ,  $[-7, 7]^2$  and  $[-8, 8]^2$ ; the points where the computed convex hull differs from the function  $f_{\text{test}}^2$  are shown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

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 is 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

---

**Algorithm 2** Convex hull algorithm with dual grid partition.

---

**Input:**  $X_N \rightarrow \text{xgrid}, Y_N \rightarrow \text{ygrid},$

$f(X_N \times Y_N) \rightarrow \text{values}, S_M^{(j)}, j = 1, \dots, s \rightarrow \text{S\_M}(s)$

**Output:**  $(f_{\Omega_N}^*)_{S_L}^* \rightarrow \text{cur\_values}$

```

! 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

```

---

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  $\bigcup_{j=1, \dots, s} S_M^{(j)}$ , where  $|S_M^{(j)}| = |\Omega_N|$  for all  $j = 1, \dots, s$ ; the results obtained using the various dual grids  $S_M^{(j)}$  are then merged using  $(f_{\Omega_N}^*)_{S_L}^* = \max_{j=1, \dots, s} (f_{\Omega_N}^*)_{S_M^{(j)}}^*$ . 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 LFT's only once and use this result in the calculation of each  $(f_{\Omega_N}^*)_{S_M^{(j)}}^*, j = 1, \dots, s$ .

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 8). As discussed in Remark 19, the difficulty 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, herein referred to as adaptive, uses as dual grid  $S_L$  all the slopes of the functions obtained in the first step of one-dimensional convex hulls, i.e., the union of all the grids `pcws(:).grid` of one-dimensional discrete LFT's. This large grid in dual space can be partitioned naturally in correspondence to each line on which the one-dimensional hull 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 hull is not necessarily well-approximated. In practice, accurate results are achieved by considering only few lines; for example,

we can fix  $2 \leq s \leq \text{size}(\text{ygrid})$  and take  $S\_M(j)=\text{pcws}(\text{indexes}(j)).\text{grid}$ , where  $\text{indexes}=\text{floor}(\text{linspace}(1,\text{size}(\text{ygrid}),s))$ .

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

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

In Figure 8 the computed non-convexity region of  $f_{\text{test}}^3$  is plotted for an adaptive non-uniform dual grid with  $s = 2, 3, 10, 20$ . The primal grid has size  $1000 \times 1000$ , while the dual grid has size  $(1000s) \times (1000s)$ . As a comparison, we have included the results obtained with uniform dual grids of size  $1000^2$  (upper left) and  $(1000s) \times (1000s)$  with  $s = 100$  (bottom right); for the latter plot, Algorithm 2 has been used with  $s = 100$ . As for  $f_{\text{test}}^2$ , it is possible to compute with great accuracy the exact convex hull of  $f_{\text{test}}^3$  and thus its exact non-convexity region; although this region is different from the non-convexity region of  $(f_{\text{test}}^3)_{\Omega_N}$  (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 an uniform dual grid with  $s = 100$ .

*Remark 20.* We can build a hierarchical version of Algorithm 2 by considering a sequence of grids  $S_L^{(1)} \subset S_L^{(2)} \subset \dots$ ; for example, we can take  $S_L^{(i)}$  as the union of  $\text{pcws}(j).\text{grid}$  with  $j$  varying in  $\text{floor}(\text{linspace}(1,\text{size}(\text{ygrid}),2**i+1))$ . At each step  $i$ , we compute the double discrete LFT for the grid  $S_L^{(i)} \setminus S_L^{(i-1)}$  using Algorithm 2 and then merge it with the one obtained for  $S_L^{(i-1)}$  by a maximum operation, thus obtaining the double discrete LFT for  $S_L^{(i)}$ . We observe that devising a suitable stopping criterion is, however, not straightforward. We can stop the computation when a certain distance between the current result and the preceding one is below a prescribed tolerance. A possible problem in this approach is that the distance between successive results may remain constant and small if the lines which are added at each step are not close to non-convexity regions which have not been explored yet; this happens frequently if the non-convexity region has many connected components which are small compared to full domain.

## 4 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

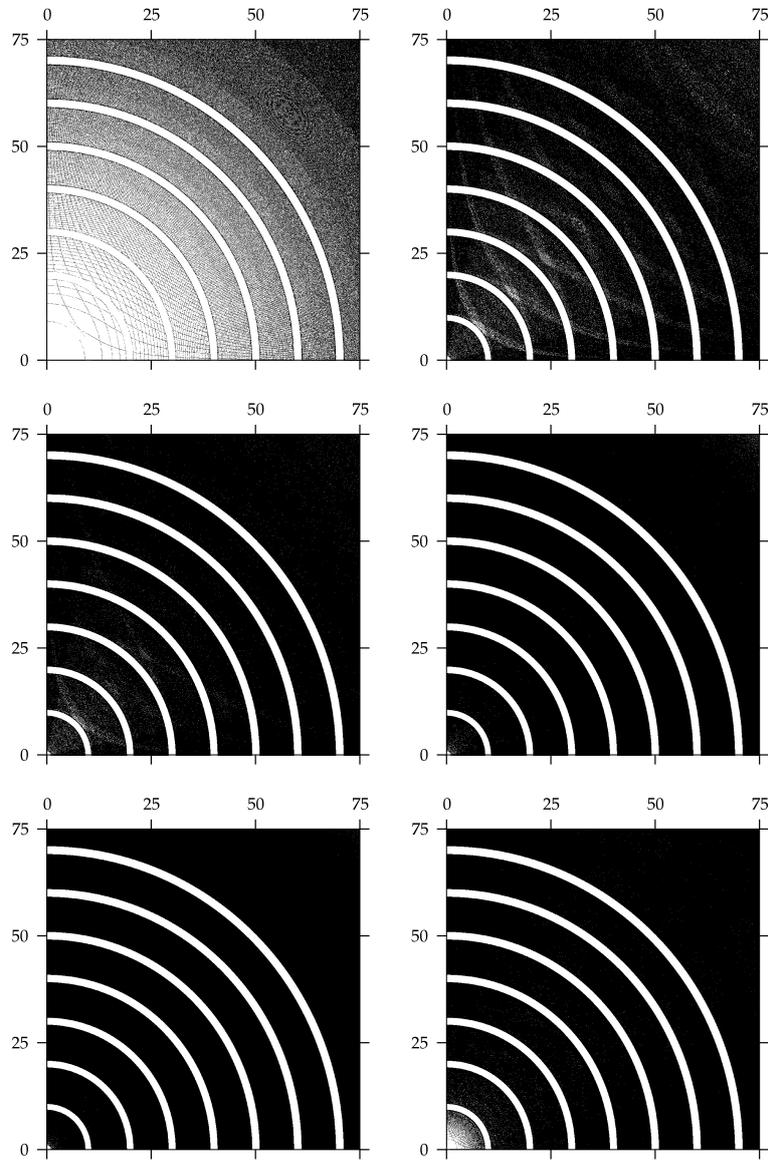


Figure 8: Plots of the non-convexity regions of the test function  $f_{\text{test}}^3$  on the domain  $[-75, 75]^2$  covered by an uniform  $1000 \times 1000$  grid. The upper left plot is produced with an uniform dual grid of size  $1000 \times 1000$ , while for the bottom right plot a uniform dual grid of size  $(1000s) \times (1000s)$  with  $s = 100$  is used; for the other plots, an adaptive 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.

steric correlations due to excluded volume effects. 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., [4].

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  $\sigma$ . 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  $f(\rho) = f_{\text{id}}(\rho) + f_{\text{corr}}(\rho)$ , where the ideal term is given by

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

with  $\beta := (k_{\text{B}}T)^{-1}$  where  $k_{\text{B}}$  is the Boltzmann constant and  $T$  the temperature. The non-ideal term  $f_{\text{corr}}(\rho)$  is detailed in [7], which is devoted to the study of phase separation in ionic solutions in confined situations. For the present purpose, it suffices to know that  $f_{\text{corr}}(\rho)$  only depends on the non-dimensional reduced temperatures

$$T_\pm^* = k_{\text{B}}T \frac{4\pi\varepsilon\sigma}{Z_\pm^2 e^2},$$

where  $\varepsilon$  is the solvent dielectric permittivity and  $e$  the elementary charge. Moreover, due to excluded volume effects,  $f_{\text{corr}}(\rho)$  becomes unbounded whenever the packing number  $\xi(\rho) = \frac{\pi}{6} \hat{\rho}_{\text{tot}}$ , with reduced total ionic density  $\hat{\rho}_{\text{tot}} = \hat{\rho}_+ + \hat{\rho}_-$ , 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 [3] 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  $T_{\text{crit}}^* \approx 0.07857$ , so that  $f$  is no longer convex in  $\rho$ . 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  $\rho_+$  (resp.,  $\rho_-$ ) 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  $f_{\text{corr}}$  only depends on  $\hat{\rho}_{\text{tot}}$ , which ensures that  $f$  and its convex hull 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 9; 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 hull 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.

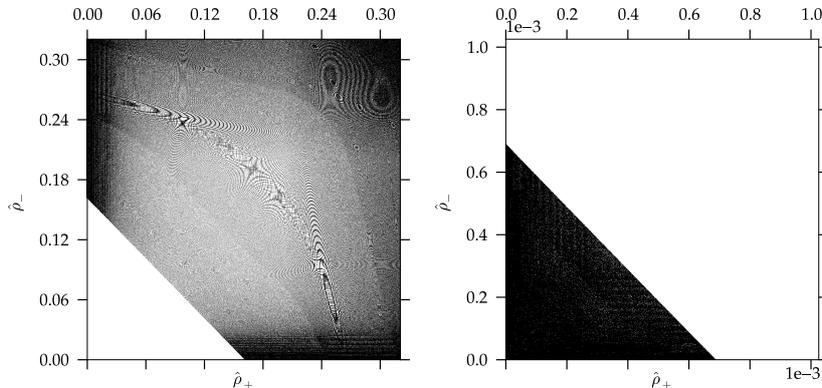


Figure 9: Non-convexity region of the bulk free energy for  $Z_{\pm} = \pm 2$  and  $T^* = 0.07$ ; the black points are where  $f = \text{conv } f$ . The right panel provides a zoom of the left panel near the origin.

Now, we consider the non-symmetric case  $Z_+ = +2$  and  $Z_- = -1$ . The non-convexity regions are plotted in Figure 10 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  $T_{\text{crit}}^*$  so that the non-convexity region of  $f$  touches the  $\rho_+$ -axis, whereas  $T_-^* = 4T_+^*$  lies above the critical value so that the non-convexity region does not intersect the  $\rho_-$ -axis. Instead, for  $T_+^* = 0.0196$ ,  $T_-^*$  also falls below the critical value so that the non-convexity region also intersects the  $\rho_-$ -axis. This is illustrated by the zoom near the origin provided by the plots in the second row of Figure 10. As a further illustration, we consider the case  $Z_+ = +3$  and  $Z_- = -2$ . The non-convexity region is presented in Figure 11 for  $T_+^* = 0.03499$  (left panel) and  $T_+^* = 0.03491$  (right panel). In both cases,  $T_+^*$  falls below the critical value  $T_{\text{crit}}^*$ , but only in the second case,  $T_-^*$  falls (slightly) below  $T_{\text{crit}}^*$ .

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 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  $5000 \times 5000$ , i.e., with a dual grid of size  $(5 \times 10^6) \times (5 \times 10^6)$ ), the results are not satisfactory. The adaptive non-uniform dual grid constructed in Section 3.5 yields instead a more accurate approximation, where the border of the non-convexity region is better resolved (see Figure 12).

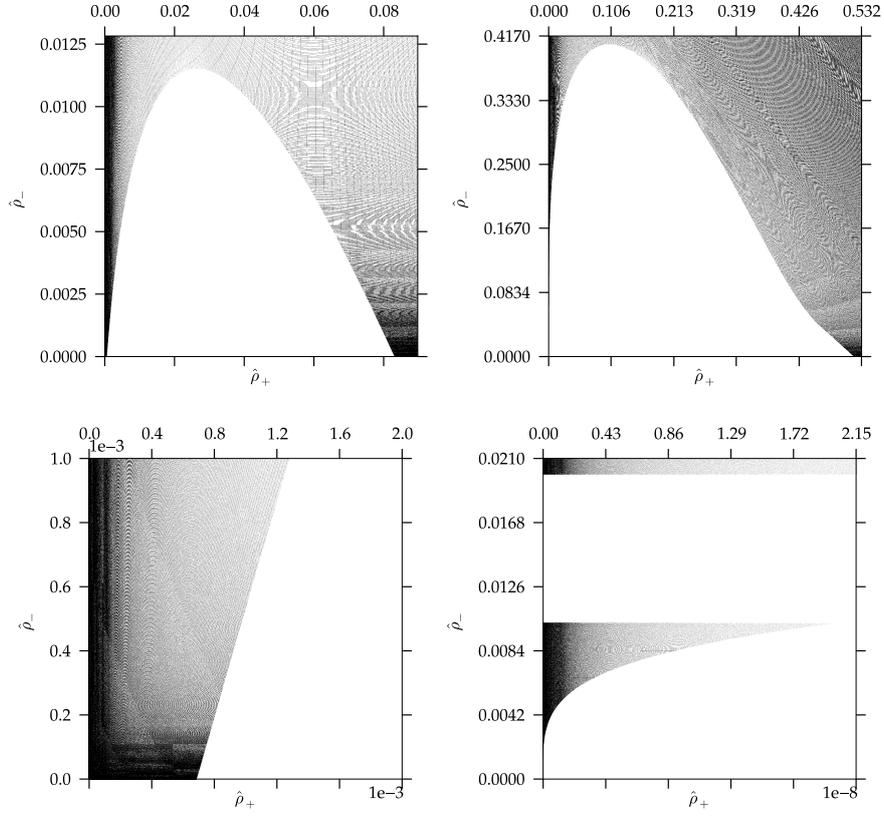


Figure 10: Non-convexity region of the bulk free energy for  $Z_+ = +2$ ,  $Z_- = -1$ , and  $T_+^* = 0.07$  (left) or  $T_+^* = 0.0196$  (right). The second row provides a zoom of the corresponding panels from the upper row near the origin.

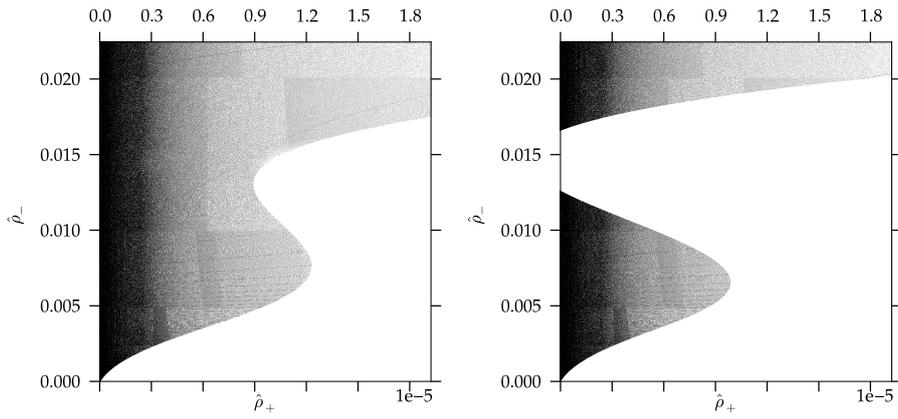


Figure 11: Non-convexity region of the bulk free energy for  $Z_+ = +3$ ,  $Z_- = -2$ , and  $T_+^* = 0.03499$  (left panel) or  $T_+^* = 0.03491$  (right panel).

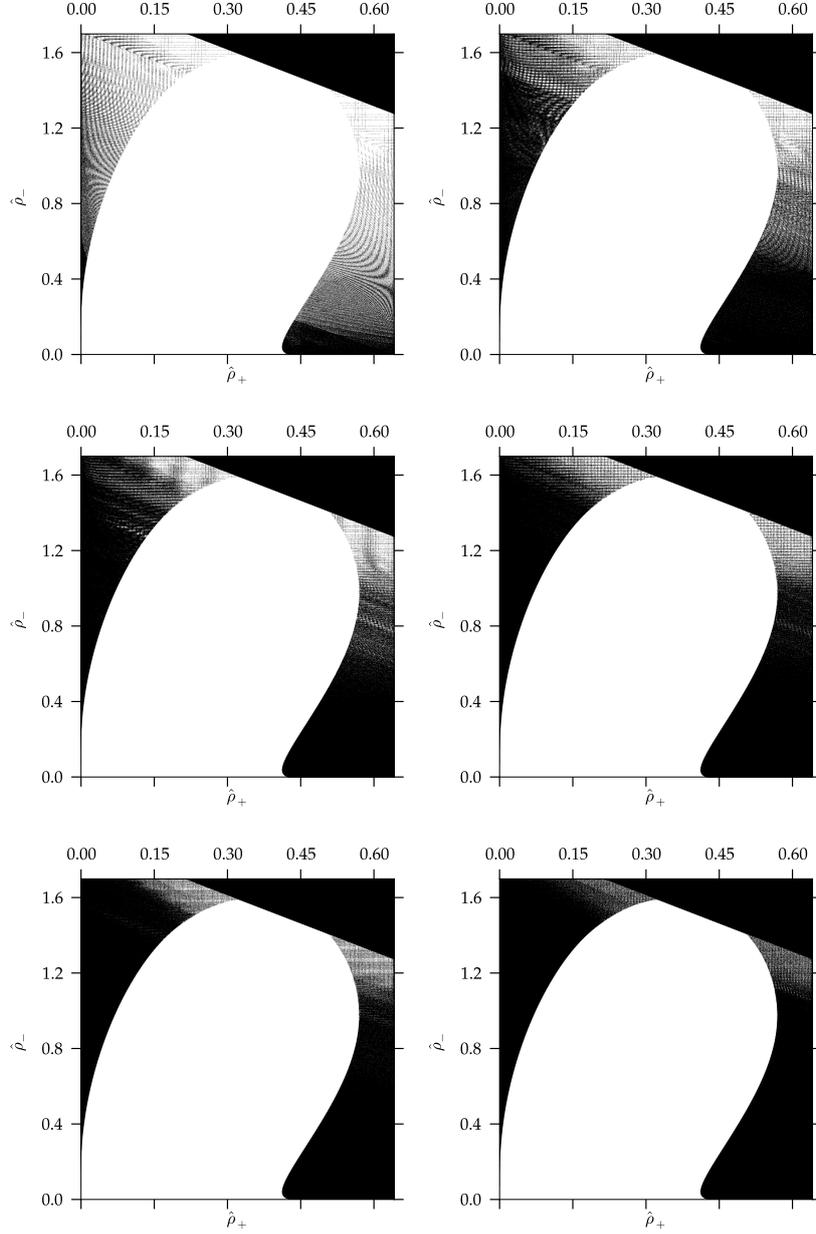


Figure 12: Non-convexity region of the bulk free energy for  $Z_+ = +3$ ,  $Z_- = -1$ , and  $T_+^* = 0.0266$ . The values of  $s$  used are, respectively from left to right and from top to bottom, 2, 5, 10, 20, 30 and 100. The black triangular zone in the upper right corner of each plot lies outside the domain of the bulk free energy.

## 5 Conclusions

In this work, we have studied convex hull algorithms based on the double discrete LFT. In addition to the standard factorization algorithm presented in [8] and [5], we have proposed the alternating factorization variant specifically tailored to bivariate functions, which we have shown to be more efficient than the standard one. Additional improvements are achieved by the max-alternating factorization. Moreover, we have highlighted that the quality of the resulting approximations given by both the standard and alternating variants of the 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 issue, we have presented an efficient method to handle dual grids of arbitrary length and we have proposed a computationally-effective approach for generating non-uniform dual grids. Finally, we have applied the present algorithms to the study of phase separation in ionic solutions consisting of cations and anions dissolved in a solvent, confirming the validity of the approach.

We conclude with the observation 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 hull data in the non-convexity region, whereas for the values outside the region, the analytical expression of the function  $f$  can be used. The non-convexity regions resulting from the application of a threshold to the difference  $(f - \text{conv } f)$  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  $f \neq \text{conv } f$  (as it has been the case up to now), it is possible to use the closing operation from mathematical morphology (see [11]): 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.

## References

- [1] Lorenzo Contento. The Discrete Legendre–Fenchel Transform and its application to phase separation in electrolytes. Master’s thesis, University of Udine, December 2012. URL <http://hal.archives-ouvertes.fr/hal-00806085>.
- [2] Lucilla Corrias. 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. URL <http://epubs.siam.org/doi/abs/10.1137/S0036142993260208>.
- [3] Alexandre Ern, Rémi Joubaud, and Tony Lelièvre. Mathematical study of non-ideal electrostatic correlations in equilibrium electrolytes. *Non-linearity*, 25(6):1635–1652, June 2012. doi: 10.1088/0951-7715/25/6/1635. URL <http://stacks.iop.org/0951-7715/25/i=6/a=1635?key=crossref.f0aec98e542e689159d558f2a1108e76>.
- [4] B. Groh, R. Evans, and S. Dietrich. Liquid-vapor interface of an ionic fluid. *Phys. Rev. E.*, 57:6944–6954, 1998.

- [5] Philippe Helluy and Hélène Mathis. Pressure laws and fast Legendre transform. *Mathematical Models and Methods in Applied Sciences*, 21(4):745–775, April 2011. doi: 10.1142/S0218202511005209. URL <http://www.worldscinet.com/m3as/21/2104/S0218202511005209.html>.
- [6] Jean-Baptiste Hiriart-Urruty. Lipschitz  $r$ -continuity of the approximate subdifferential of a convex function. *Mathematica Scandinavica*, 47:123–134, 1980. URL <http://www.msand.dk/issue.php?year=1980&volume=47&issue=>.
- [7] R. Joubaud, O. Bernard, L. Contento, A. Ern, B. Rotenberg, and P. Turq. Phase separation in highly charged confined ionic solutions. Technical report, University Paris-Est, CERMICS, 2013. URL <http://hal.archives-ouvertes.fr/hal-00806218>.
- [8] Yves Lucet. Faster than the Fast Legendre Transform, the Linear-time Legendre Transform. *Numerical Algorithms*, 16(2):171–185, March 1997. doi: 10.1023/A:1019191114493. URL <http://www.springerlink.com/index/M41T352758814Q50.pdf>.
- [9] James Clerk Maxwell. On the Dynamical Evidence of the Molecular Constitution of Bodies. *Nature*, 11:357–359, March 1875. doi: 10.1038/011357a0. URL <http://www.nature.com/nature/journal/v11/n279/pdf/011357a0.pdf>.
- [10] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1 edition, 1970. ISBN 0-691-08069-0.
- [11] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, January 1983. ISBN 978-0126372427.